AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT. 1956

Master of Computer Applications (MCA)

REGULATION, CURRICULUM AND SYLLABUS
2026

MCA (Master of Computer Applications)-2026

AMRITA VISHWA VIDYAPEETHAM

REGULATIONS FOR THE MCA PROGRAMME

UNDER COMPUTING

(effective from 2026 admissions)

GENERAL

GENERAL

1. The following are the eligibility criteria for the MCA programme.

Eligibility Norm 1	Candidates who have passed B. E or B. Tech are eligible for MCA admission.
Eligibility Norm 2	Candidates who have passed B.Sc. in Computer Science/BCA is also eligible for admission.
Eligibility Norm 3	Candidates who have passed B.Sc. with Mathematics as one of the subjects are eligible for admission
Eligibility Norm 4	Candidates who have passed B. A/ B.Sc. / B. Com/ any other degree with Mathematics as one of the subjects in Class 12th are eligible for admission in MCA course.

2. Procedure for admission will be decided from time to time by the University in accordance with the guidelines from competent authorities.

3. The duration of the Programme will normally be four semesters, spread over two academic years.

4. The award of the respective PG degrees will be recommended by the Academic Council and approved by the Board of Management in accordance with the regulations of the University.

5. Notwithstanding anything stated above, the Amrita Vishwa Vidyapeetham reserves the right to modify any of the ordinances, as deemed fit, from time to time.

R.1 Admissions

R.1.1 The admission to the programme will be as per the ordinances and regulations of the University.

R.1.2 The intake to each school will be decided by the University from time to time.

R.1.3 Transfer of students from one campus to another is generally not permitted. However, based on the availability of vacancy in the discipline and the academic merit of the student, special cases may be allowed in the beginning of the third semester, on the mutual consent of the Heads of both the Departments and Schools and with the approval of the University. The decision of the University will be final in this matter.

R.2 Language of Instruction

The language of instruction will ordinarily be English, for all courses. For *Cultural Education*, instruction may be given partly in Indian languages. In case of languages, the instruction may be in English or in the language concerned.

R.3 Structure of the Programme

R.3.1 The Programme will be structured on a credit-based system and continuous evaluation, following semester pattern.

R.3 2 The programme consists of the following:

(a) Core courses in the primary area of the programme, including seminars, projects, etc.

(b) Humanities and General Studies (like *Cultural Education. A*dditionally, soft skill training and some social interaction/social work programmes, like Live-in Labs, may also be offered.

(c) Electives (General stream, AI & Data Science stream and Cyber Security Stream)

(d) Open labs

R.3.3 The curriculum of the PG degree programme will have credits, apportioned as below in the following knowledge segments:

Core courses /electives/ Professional electives/ Open labs

Electives

Humanities and General Studies

R.3.4 Credits are assigned to the courses based on the following general pattern -

One credit for each lecture period per week

One credit for each tutorial period per week

One credit for each laboratory course/practical of two/three periods per week

R.3.5 Each PG degree programme shall have a prescribed curriculum and syllabi, which will be periodically updated according to the requirements and approved by the Academic Council.

R.3.6 All the Schools will be governed by the same curricula and syllabi for the respective programmes.

R.3.7 Certain courses are identified as Core courses and a few others as electives.

There is mandatory registration and credit earnings requirement for core courses. While it is mandatory to register for elective courses, failure to earn credit in them does not necessarily require repeating the courses. Often another elective course may be permitted as a replacement course, through Regular registration, with the concurrence of the Class Advisor and the Head of the Department.

R.3.8 Students are allowed to take online courses through Government portals such as NPTEL and SWAYAM

1. Amrita's grade, equivalent to the score secured in online courses to be decided by the class committee, and the same is to be awarded to the students.
2. The Class committee should authorize the courses before registration.
3. Students who have a Cumulative Grade Point Average (CGPA) of 6.5 or higher are eligible to enroll in NPTEL/SWAYAM courses for credits. However, the number of credits should not exceed eight considering the entire program.
4. The Course registration requisite of a maximum of 28 credits per semester is to be maintained while registering for online courses.

R.3.9 Programme Educational Objectives (PEOs)

PEO 1: Graduates will be technically strong with comprehensive knowledge and skills to design and develop innovative software for emerging requirements.

PEO 2: Graduates will be continuous learners with aptitude for research with societal focus.

PEO 3: Graduates will be proficient to be employed as Consultant / Entrepreneurs in the IT and ITES industries.

R.3.10 **Programme Outcomes (POs)**

PO 1. **Foundation Knowledge:** Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.

PO 2. **Problem Analysis:** Identify, review, formulate and analyse problems for primarily focussing on customer requirements using critical thinking frameworks.

PO 3. **Development of Solutions:** Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.

PO 4. **Modern Tool Usage:** Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.

PO 5. **Individual and Team Work:** Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.

PO 6. **Project management and finance:** Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.

PO 7. **Ethics:** Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware.

PO 8. **Life-long learning:** Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.

R.4 Tuition Fees

At the beginning of each academic semester, students shall pay all the fees prescribed. A student who drops out of the programme or whose registration is cancelled due to any reason cannot claim refund of any fees paid.

R.5. Mentors and Class Advisor

R.5.1 In order to (i) guide the students in planning their courses of study, (ii) advise them on academic programmes and (iii) monitor their progress, the departments will assign a batch (class) of a certain number of students to a faculty member, who will be designated as their mentors.

R.5.2 One among the mentors shall be designated as the Class Advisor, who shall coordinate the functions of the Class mentors.

R.6. Course Mentor

R.6.1 For each course offered in a School, one of the teachers teaching that subject shall be nominated by the Head of the Institution of the School as the course mentor of the course.

R.7. Course Committees

R.7.1. Course committees are constituted for running courses which are common for more than one discipline. Course committees will be set up in each School for each group of similar courses as decided by the Head of the School.

R.7.2. Each course committee will consist of the following members:

(i) The chairperson of the course committee, nominated by the Head of the School.

(ii) All teachers involved in teaching the courses, shall be included in the course committee

(iii) Two student members from each discipline are nominated into the committee by the chairpersons of departments with equal representation to boys and girls, to the extent possible.

(iv) Chairpersons of the departments and the School Head may attend meetings of the course committees.

R.7.3. The Course Committees shall meet at least thrice in a semester. The Course committees shall meet at the beginning of the semester to finalize the course plans for the academic programme. They shall meet at the end of the semester, without the student representatives, to finalize the results of the respective courses. The results shall be submitted to the Head of the School, who shall approve and forward the same, to the Examination section.

R.7.4. The Course committees shall be reconstituted at the beginning of every semester.

R.8 Class Committees

R.8.1 Every class of a degree programme in each School will have a Class Committee constituted by the Head of the School, based on the recommendation of the Chairperson of the department.

R.8.2. The Constitution of the Class Committee will be as follows:

a) The Chairperson of the class committee, nominated by Head of the School,

b) All the teachers handling courses for the class,

c) Two student representatives nominated by the Chairperson of the department, from each class with equal representation of boys and girls, to the extent possible.

d) The Chairperson of the department and the School Head may attend the committee meetings.

R.8.3. The Class Committee shall meet at least thrice in a semester. The Class committee shall meet at the beginning of the semester to finalize the academic programme. At the end of the semester the committee (without student representatives) will meet to finalize the results. The results shall be submitted to the Head of the School, who shall approve and forward the same, to the Examination section.

R.8.4. The Class committees shall be reconstituted at the beginning of every semester.

R.9 Registration and Enrolment

R.9.1 Every student shall register for the courses which he/she wishes to undergo during a semester.

R.9.2 Except for the first semester, pre-registration for a semester will be done during a specified week before the end-semester examination of the previous semester. The consent of the Class Advisor is mandatory before registering for every course.

R.9.3 From the second semester onwards, all students must enroll on a specified day at the beginning of a semester. A student will be eligible to enroll only if he/she has cleared all the dues to the Institution, hostel, library, etc., at the time of enrolment and if he/she is not debarred from enrolment, as part of any disciplinary action of the Institution.

R.9.4 Late enrolment will be permitted on payment of a prescribed late fee, up to a specified date, to be notified well in advance.

R.9.5 A student can register for a maximum of 28 credits, in a semester, including all the remedial provisions.

R.9.6 **Pre-requisites:** A student is not permitted to register for a course unless he/she has already attended the pre-requisite course, wherever specified. A student will not be deemed to have attended the pre-requisite, if he/she gets an 'FA' grade in such a course.

R.10 Dropping /Substituting Courses

R.10.1 If a student finds his/her load heavy in any semester, or for any other valid reasons, he/she may drop courses, within **first ten working days** of the commencement of the semester, with the written approval of his/her Class Advisor and Chairperson of the Department.

Withdrawal from one or more enrolled courses after the specified date will entail academic penalties in the form of a 'Failed due to insufficient attendance' 'FA' grade appearing in the grade sheet.

R.10.2 A student can substitute a course registered earlier by another for valid reasons, within **the first ten working days** of the commencement of the semester, with the consent of the Class Advisor and Chairperson of the Department.

R.11 Maximum Duration of the Programme

R.11.1 A student is expected to complete the PG programme in four semesters. However, a student may complete the programme at a slow pace within six semesters, with the prior permission of his/her Class Advisor and Chairperson of the Department and Head of the School.

R.11.2 A student may be permitted by the concerned Head of the School to withdraw from the programme for a semester or a longer period for reasons of ill health or on other valid grounds. However, the programme should be completed within a total span of six semesters.

R.11.3 In the event of any student requiring more than six semesters to complete the programme, the extension can be considered on the merits of the case, by the Vice-Chancellor and ratified by the Academic Council.

R.12 Attendance

R.12.1 Attendance of the students will be marked by the concerned teacher during every hour of the course.

A student is required to put in a minimum of 75% of attendance in each of the regular or remedial courses, he/she has registered.

R.12.2 Students who have been selected to be trained for international competitions or have secured distinction in the previous year examinations and are working on directed research under a faculty member, and approved by the Chairperson of the Department, shall be given a waiver of up to 25% attendance.

R.12.3 Leave shall be availed by students only under unavoidable circumstances. It is mandatory that students apply in the prescribed form before proceeding on leave. A leave letter recommended by the Class Advisor shall be submitted to the Chairperson of the department who will consider grant of the leave. Unauthorized absence will be treated as breach of discipline.

Request for leave for more than three consecutive days on medical grounds must be supported by a proper medical certificate. In non-medical cases, requests for leave for more than three consecutive days must be countersigned by the parent/guardian or the Warden, whichever is applicable.

Leave granted will not be counted as physical presence.

R.12.4 Students going on official duty, such as representing the college/University for sports and cultural activities, or presenting papers in seminars, conferences, etc., will be eligible for 'duty leave' on the recommendation of the Class Advisor and approval by the Chairperson of the Department. **Students should get this leave sanctioned before proceeding on 'duty leave'. They will be granted attendance for the periods they missed on account of the duty leave upon production of the relevant participation certificate after attending the duty.**

All kinds of leave, authorized by the Chairperson of the Department, shall not exceed 25% of the total hours in the course.

R.12.5 Finalization of attendance for every course shall be done three working days before the last instruction day of the semester. Any student failing to secure a minimum of 75% attendance in a course, will not be eligible to appear for the end-semester examination in that course.

R.12.6 In case a student who is not permitted to attend the end-semester examination in any course due to shortage of attendance, will be awarded 'FA' grade in that course, indicating "failed due to insufficient attendance" and mentioned in the grade sheet.

Students awarded 'FA' grade in a course, shall re-register for the course, when offered next or as a run-time re-do course.

R.13 Assessment Procedure

R.13.1 The academic performance of each student in each course will be assessed based on Internal Assessment (including Continuous Assessment) and an end-semester examination.

Normally, the teachers offering the course will evaluate the performance of the students at regular intervals and in the end-semester examination.

In theory courses (that are taught primarily in the lecture mode), the weightage for the Internal Assessment and End-semester examination will be 50:50. The Internal assessment in theory courses shall consist of one mid-term examination, weekly quizzes, assignments, tutorials, viva-voce etc. The weightage for these components, for theory-based courses, shall be 25 marks for the Continuous assessment, comprising of Quizzes, assignments, tutorials, viva-voce, etc. and 25 marks for the mid-term examination.

At the end of the semester, there will be an end-semester examination of three hours duration, with a weightage of 50 marks, in each lecture-based course.

R.13.2 In the case of laboratory courses and practical's, the relative weight for Internal assessment and End-semester examination will be 70:30. The weight for the components of Internal assessment will be decided by the course committee/class committee at the beginning of the course.

Evaluation pattern for course having both Theory and Lab. components:

Courses having only one hour per week for lecture/tutorial, be treated as a Lab course, for evaluation purposes; and the evaluation pattern will be 70 marks for continuous assessment and 30 marks for the end-semester Examination.

Courses having two hours or three hours per week for theory and/or tutorials, be given a weightage of 60 marks and 40 marks for the Theory and Lab components respectively; The Lab. component evaluation will be based on continuous evaluation, without any end-semester practical evaluation. 10 marks will be for continuous assessment of the theory portion, 20 marks for the mid-term examination, 30 marks for the theory end-semester examination, and 40 marks for continuous assessment of lab work.

R.13.3 It is mandatory that the students shall appear for the end-semester examinations in all theory and practical courses, for completion of the requirements of the course. Those who do not appear in the end-semester examinations will be awarded 'F' grade, subject to meeting the attendance requirement.

At the end of a semester, examinations shall be held for all the subjects that were taught during that semester and those subjects of the previous semesters for which the students shall apply for supplementary examination, with a prescribed fee.

R.13.4 PROJECT WORK: The continuous assessment of project work will be carried out as decided by the course committee. At the completion of the project work, the student will submit a bound volume of the project report in the prescribed format. The project work will be evaluated by a team of duly appointed examiners. The final evaluation will be based on the content of the report, presentation by student and a viva-voce examination on the project. There will be 60% weightage for continuous assessment and the remaining 40% for final evaluation.

If the project work is not satisfactory, he/she will be asked to continue the project work and appear for assessment later.

Course Category	**L-T-P**	**Internal: External**	**Internal Theory Weightage in Marks**		**External (%)**	**Total Theory Weightage (%)**	**Total Lab Weightage (%)**
			Mid-Term Examination	**CE**			
Theory without a lab component	3-0-0 / 3-1-0	50:50	25	25	50	100	NA
Theory with a lab component	2-0-1/ 3-0-1	70:30	20	10	30	60	40*
Lab Courses	0-1-1 / 1-0-1 / 0-0-1	70:30	NA		30	NA	100

Dissertation Phase I / Dissertation Phase II		60:40	NA	-	-	-
* No End Semester Lab Examination						

The duration for a lab will be either 2 or 3 hours, depending on the course and will be decided by the campus.

R.14 PUBLICATION / INTERNSHIP

R.14.1 All students, to be considered for the award of Degree at the time of graduation, must have published ONE paper in a Scopus-indexed Journal/Conference. The publication shall be as per the guidelines prescribed by the University.

R.15 REMEDIAL PROVISIONS

R.15.1 **Supplementary Examinations:**

Students who failed in a non-semester course (i.e. courses not registered by the student during the current semester), shall apply for by appearance in the respective examination paying a prescribed fee and take the examination.

A student who has secured an ‘F’ grade in a course may take the supplementary examination for a maximum of three additional attempts (excluding the regular end-semester examinations) carrying the previous Internal marks earned by them. Students failing to pass the course after three additional attempts shall henceforth appear for the supplementary examination for the entire 100 marks and the Internal assessment marks earned by them in the regular registration shall not be considered.

If a student wishes to improve his/her internal marks, he/she can do so, by re-registering for the course by choosing any of the appropriate remedial options. In this case, the internal marks obtained by the student will be valid for the end semester of the re-registration and three more additional attempts.

R.15.2 **Other options:**

Certain courses may be offered as run-time-redo or as contact courses, as and when necessary to enable students who have dropped courses or failed in some courses, to register and endeavor to complete them.

a) Re-registration: Students who have failed a course and opt to re-do the course may do so by re-registering for the course, along with a junior batch of students,

R.15.3 Supplementary examinations will be evaluated against the most recent grade rule (whenever the course was offered recently in the regular semester).

R.16 Grading

R.16.1 Based on the performance in each course, a student is awarded at the end of the semester a letter grade in each of the courses registered. Letter grades will be awarded by the Class Committee in its final sitting, without the student representatives.

The letter grades, the corresponding grade points and the ratings are as follows:

Letter Grade	*Grade Points*	*Ratings*
O	10.00	Outstanding
A+	9.50	Excellent
A	9.00	Very Good
B+	8.00	Good
B	7.00	Above Average
C	6.00	Average
P	5.00	Pass
F	0. 00	Fail
FA	0. 00	Failed due to insufficient attendance
I	0. 00	Incomplete (awarded only for Lab. courses/ Project /Seminar)
W		Withheld

R.16.2 'FA' grade once awarded stays in the record of the student and is replaced with the appropriate grade when he/she completes the course successfully later.

Students who have secured an 'FA' in a course must re-register for the course or register for the course, if offered, under run-time re-do mode.

R.16.3 A student who has been awarded 'I' Grade in a Lab course, due to reasons of not completing the Lab, shall take up additional Lab whenever offered next and earn a pass grade, which will be reflected in the next semester's grade sheet.

The 'I' grade, awarded in a Project/Seminar course, will be subsequently changed into an appropriate grade, when the student completes the requirement during the subsequent semester. If he/she does not complete it in the next semester, it will be converted to 'F' grade.

R.16.4 A student is considered to have successfully completed the course and earned the credit if he/she scores a letter grade 'P' or better in that course.

R.17 Declaration of Result

After finalization of the grades by the Class Committee and subsequent approval of the Head of the School, the result will be announced by the Controller of Examinations.

R.18 Revaluation of answer Papers

On publication of the results, an aggrieved student can request for revaluation of answers scripts of the end-semester examination, within five working days of publication of the results, along with the prescribed revaluation fees. The request has to be made to the Examination Section, through the Head of the School.

If the revaluation leads to a better grade, the revised grade will be awarded to the student and in such cases, the revaluation fee will be refunded in full.

Revaluation is permitted only for lecture-based courses.

R.19 Course completion:

A student is said to have successfully completed a course and earned the corresponding credits, if he/she has:

- registered for the course:
- put in 75% or more attendance in the course,
- appeared for the end-semester examinations,
- obtained a pass grade ‘P’ or better in the course,
- no pending disciplinary proceedings against him/her.

R.20 Grade Sheet

The Grade Sheet issued to the student at the end of a semester will contain the following information:

- Name, Roll No. Grade Sheet No., Semester, Branch, Month and year of the Examination
- Course Code, Course Title, Credits, Grade obtained, and Grade points earned for the courses registered
- Credits registered and earned during the semester
- Cumulative credits earned and Grade Points
- SGPA and
- CGPA

R.21 Semester Grade Point Average (SGPA)

On completion of a semester, each student is assigned Semester Grade Point Average (SGPA) which is computed as below for all courses registered by the student during that semester.

Semester Grade Point Average = $\sum (C_i \text{ x } Gp_i) / \sum C_i$

where C_i is the credit for i^{th}course in that semester and Gp_i is the grade point for that course.

The summation is over all the courses registered by the student during the semester, including the failed courses. The SGPA is rounded off to two decimals.

R.22 Cumulative Grade Point Average (CGPA)

The overall performance of a student at any stage of the Degree programme is evaluated by the Cumulative

Grade Point Average (CGPA) up to that point of time.

Cumulative Grade Point Average = $\sum (C_i \text{ x } Gp_i) / \sum C_i$

where C_i is the credit for i^{th} course in any semester and Gp_i is the grade point for that course.

The summation is over all the courses registered by the student during all the semesters up to that point of time, including the failed courses. The CGPA is also rounded off to two decimals.

R.23 Ranking

The ranking of the students in a batch at any intermediate or final stage is based on CGPA. Only those students who have passed all courses up to that stage in the first attempt are considered for ranking.

Students are eligible for final ranking, only if they complete the programme within the normal duration, i.e., within two years of joining the programme.

R.24 Classification of successful candidates:

R.24.1 A student shall be considered to have successfully completed the programme, if he/she has:

i) registered and successfully completed all the core courses, electives, and projects as mentioned in the curriculum.

ii) earned the required minimum number of credits as specified in the curriculum corresponding to the programme, within the stipulated time.

R.24.2 Candidates who have successfully completed the programme, within a period of four semesters from entering the programme, shall be classified as follows:

Candidates securing a CGPA of 8.00 and above – FIRST CLASS WITH DISTINCTION

Candidates securing a CGPA between 6.50 and 7.99 – FIRST CLASS

and the same be mentioned in the Degree certificate.

If the programme is completed after four semesters of study, the candidates securing even a CGPA of 8.00 and above, shall be classified to have completed the programme, only with FIRST CLASS.

R.25 Transcript

The Controller of Examinations will also issue, on request and payment of a prescribed fee, a detailed transcript with his signature or facsimile to every student after completion of the programme. It shall contain all the information that is contained in the grade sheets. Additionally, it shall also include the month and year of passing each course. The transcript card shall contain only the final grades secured, but will not indicate the earlier failures, if any. The detailed transcript will contain the CGPA and the class, if any obtained.

R.26 Discipline

Every student is required and expected to observe strict discipline and decorous behavior both inside and outside the campus. He/she should not indulge in any activity that may tarnish the fair name and prestige of Amrita Vishwa Vidyapeetham. Any act of indiscipline or misbehavior including unfair practice in the examinations will be dealt with by the Disciplinary Action Committee of the Institution, constituted by the Head of the School concerned. The committee will enquire into the charges and make recommendations to the Head of the School concerned. Based on the findings of the committee, the Head of the School will take appropriate disciplinary action. Serious acts of indiscipline on the part of the students may even attract penalties up to the extent of expulsion from the University.

R.27 Redressal of grievances

Students have the right to seek redress of grievances. For this, they must appeal in writing to the Head of the School concerned, who will take the necessary steps in the matter.

R.28 Award of the Degree

A student will be declared eligible for the award of the respective Degree, if he/she has:

a) completed the programme successfully as described in R.24.1 and

b) no outstanding dues against him/her.

c) Specialization – any student who completes the program with minimum requirement (5 and above in the respective stream out of 9 electives) in the stream concerned will be awarded the degree with specialization (Artificial Intelligence and Data Science/ Cyber Security).

The PG Degree, indicating discipline and specialization (if applicable), will be awarded by the Board of Management of Amrita Vishwa Vidyapeetham on the recommendation of the Academic Council.

R.29 Interpretation Clause

Related to any of the academic matters, whenever there arises any doubt or dispute on the interpretation of regulations or rules, the decision of the Academic Council will be final as well as binding on all concerned.

R.30 Amendment to Regulations

Notwithstanding anything stated above, the Amrita Vishwa Vidyapeetham reserves the right to modify any of the regulations, as deemed fit, from time to time.

CURRICULUM

SEMESTER I

Course Code	Course Title	L T P	Credits
26CSA502	**Object-Oriented Programming Using Java**	3 0 0	3
26MAT501	**Mathematical Foundations for Computer Applications**	3 1 0	4
26CSA503	**Data Structures**	3 0 0	3
26CSA504	**Advanced DBMS**	3 0 0	3
26CSA581	**Object-Oriented Programming Using Java Lab**	0 0 2	1
26CSA582	**Data Structures Lab**	0 0 2	1
26CSA580	**Advanced DBMS Lab**	0 0 2	1
25AVP501	**Mastery Over Mind**	1 0 2	2
22ADM501	**Glimpses of Indian Culture**	2 0 1	P/F
24LSK502	**Life Skills - 1**	0 0 2	1
	Elective I		3
	Elective II		4
	Total		26

SEMESTER II

Code	Title	L T P	Credit
26CSA511	**Design and Analysis of Algorithms**	3 1 0	4
26CSA512	**Principles and Implementation of Design Patterns**	3 0 2	4
26CSA583	**Problem Formulation & Research Tools**	0 0 2	1
26CSA584	**Design and Analysis of Algorithms Lab**	0 0 2	1
24LSK512	**Life Skills – II**	0 0 2	1
	Elective III		3
	Elective IV		3
	Elective V		4
	Elective VI		4
	Total		25

SEMESTER III

Code	Title	L T P	Credit
	Elective VII		3
	Elective VIII		4
	Elective IX		4
	Open Lab I	0 0 1	1
	Open Lab II	0 0 1	1
26CSA698	**Dissertation Phase I**		6
	Total		19

SEMESTER IV

Code	Title	L T P	Credit
26CSA699	**Dissertation Phase II**		12
	Total		12

	TOTAL CREDITS		82

PRE-REQUISITE COURSES

Sl. No	Title
1	Python Programming
2	C Programming
3	SQL Basics

* These courses can be for a duration of 10-15 days (about 2 weeks).

Electives - 3 Credits

Code	Title	L T P	Credit
AI & DS STREAM			
26CSA530	Deep Learning	3 0 0	3
26CSA531	Linear Algebra and Applications	3 0 0	3
26CSA532	Artificial Intelligence	3 0 0	3
26CSA533	Database Administration	3 0 0	3
26CSA529	Time Series Analysis	3 0 0	3
26CSA534	Information Retrieval	3 0 0	3
26CSA535	Information Science and Ethics	3 0 0	3
26CSA536	Pattern Recognition	3 0 0	3
26CSA537	Recommendation Systems	3 0 0	3
26CSA538	Web Mining	3 0 0	3
26CSA539	Business Analytics and Visualization	3 0 0	3
26CSA540	Computational Intelligence	3 0 0	3
26CSA527	Generative AI and Intelligent System	3 0 0	3
26CSA528	Agentic AI	3 0 0	3
CYBER SECURITY STREAM			
26CSA550	Essentials of cyber security	3 0 0	3
26CSA551	Malware Analysis	3 0 0	3
26CSA552	Blockchain and decentralized applications	3 0 0	3
26CSA553	Fundamentals of cybersecurity operations	3 0 0	3
26CSA554	Cloud and Infrastructure security	3 0 0	3
26CSA555	Cybersecurity Governance, Risk and Compliance	3 0 0	3
26CSA556	Cyber Security Law	3 0 0	3
26CSA557	Machine learning and artificial Intelligence in Cyber security	3 0 0	3
26CSA558	Mobile Security and Defense	3 0 0	3
26CSA559	Cyber Forensics	3 0 0	3
Regular			
26CSA560	Compiler Design	3 0 0	3
26CSA561	Advanced Operating Systems	3 0 0	3
26CSA562	Software Testing	3 0 0	3
26CSA563	Theory of Computation	3 0 0	3
26CSA564	Enterprise Resource Planning Management	3 0 0	3
26CSA565	Automation and Robotics	3 0 0	3
26CSA566	Software Defined Networks	3 0 0	3
26CSA567	Robotic Operating System	3 0 0	3
26CSA568	Software Quality Assurance	3 0 0	3
26CSA569	Web Services	3 0 0	3

26CSA570	Software Project Management	3 0 0	3

Electives - 4 Credits

Code	Title	L T P	Credit
AI & DS STREAM			
26CSA631	Data Modelling and Visualization	3 0 2	4
26CSA632	Exploratory Data Analysis	3 0 2	4
26CSA633	Data Mining and Applications	3 0 2	4
26CSA634	Machine Learning	3 0 2	4
26CSA635	Explainable AI	3 0 2	4
26CSA636	Big Data Analytics	3 0 2	4
26CSA637	Natural Language Processing	3 0 2	4
26CSA638	Large Language Models	3 0 2	4
26CSA639	Computer Vision	3 0 2	4
CYBER SECURITY STREAM			
26CSA641	Ethical Hacking and Information Security	3 1 0	4
26CSA642	System Security	3 0 2	4
26CSA643	Web Application Security	3 0 2	4
26CSA644	Advanced Network Security	3 0 2	4
26CSA645	VAPT (Vulnerability and Penetration Testing)	3 0 2	4
26CSA646	Computer Networks for Cyber Security	3 0 2	4
Regular			
26CSA651	Complex Network Analysis	3 0 2	4
26CSA652	Connected Internet of Things Devices	3 0 2	4
26CSA653	IoT and Cloud Computing	3 0 2	4
26CSA654	Computer Graphics and Visualization	3 0 2	4
26CSA655	DevOps	3 0 2	4
26CSA656	Digital Image Processing	3 0 2	4
26CSA657	Advanced Computer Networks	3 0 2	4
26CSA658	Advanced Web Technologies and Mean Stack	3 0 2	4
26CSA659	Mobile Application Development	3 0 2	4
26CSA660	Multivariate Statistics	3 1 0	4
26CSA661	Graph Theory and Combinatorics	3 10	4
26CSA662	Operations Research and Optimization	3 1 0	4
26CSA663	Programming for Problem Solving	3 0 2	4
26CSA664	Advanced Programming Using Python	3 0 2	4

Open Labs

Code	Title	L T P	Credit
26CSA585	Python Scripting for Security	0 0 2	1

26CSA586	**Ethical Hacking Lab**	**0 0 2**	**1**
26CSA587	**Python Programming**	**0 0 2**	**1**
26CSA588	**C#.Net**	**0 0 2**	**1**
26CSA589	**Android Programming**	**0 0 2**	**1**
26CSA680	**UI/ UX design**	**0 0 2**	**1**
26CSA681	**Linux Programming**	**0 0 2**	**1**
26CSA682	**Competitive programming**	**0 0 2**	**1**
26CSA683	**Edge computing**	**0 0 2**	**1**
26CSA684	**R programming**	**0 0 2**	**1**
26CSA685	**MATLAB Programming**	**0 0 2**	**1**
26CSA686	**Deep Learning Lab**	**0 0 2**	**1**
26CSA687	**SQLite**	**0 0 2**	**1**

Semester I

26CSA502	OBJECT-ORIENTED PROGRAMMING USING JAVA	L-T-P-C: 3-0-0-3

Course Description

This course introduces object-oriented programming concepts and their application using Java. Students will learn to design, implement, and test robust software systems by mastering core OOP principles, and advanced features like exception handling, input/output, multithreading, and concurrency.

Course Objectives

- To equip students with a strong foundation in object-oriented programming (OOP) concepts, including abstraction, encapsulation, inheritance, and polymorphism, and their practical application in Java programming.
- To develop students' ability to design, implement, and test robust Java programs using fundamental programming constructs, data types, control flow, and object-oriented principles.
- To introduce students to advanced Java features such as exception handling, input/output operations, multithreading, and concurrency, enabling them to build efficient and responsive applications.

Course Outcomes

COs	Description
CO1	Demonstrate a solid understanding of object-oriented programming concepts.
CO2	Design, implement, and test Java applications.
CO3	Understand and employ concurrency concepts.
CO4	Apply Java programming skills to solve real-world problems.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	3	1	-	-	-	-	1
CO2	3	3	2	-	-	-	-	2
CO3	3	3	2	-	-	-	-	2
CO4	3	3	2	-	1	-	-	2
CO5	3	3	2	-	1	-	-	2

Prerequisites
Basic Programming languages like C, C++, etc.

Syllabus

Unit I - Introduction to Object-Oriented Programming and Java Fundamentals

Object-Oriented Programming (OOP) Concepts - Abstraction, Encapsulation, Inheritance, Polymorphism
Introduction to Java (Characteristics of Java, Java Environment, Java Source File Structure, Compilation Process)
Fundamental Programming Constructs - Data Types, Variables, Operators - Control Flow (Conditional and Looping Statements) - Arrays, Strings.

Unit II - Classes, Objects, and Inheritance

Classes and Objects - Defining Classes, Constructors, Methods, Access Specifiers, Static Members, Comments Inheritance - Superclasses and Subclasses, Protected Members, Constructors in Subclasses, The Object Class, Abstract Classes and Methods, Final Methods and Classes.

Unit III - Interfaces, Exceptions, and Input/Output

Interfaces - Defining and Implementing Interfaces, Differences between Classes and Interfaces.

Exception Handling - Exception Hierarchy, Throwing and Catching Exceptions, Built-in Exceptions, Creating Custom Exceptions.

Input/Output - Streams (Byte and Character), Console Input/Output, File Input/Output.

Unit IV - Multithreading and Concurrency

Multithreading - Thread-based vs. Process-based Multitasking, Java Thread Model, Creating and Managing Threads, Thread Priorities, Inter-thread Communication.

Concurrency - Concurrency Issues (Safety, Liveness, Fairness), Locks and Synchronization, Thread Pools, Futures and Callable, Fork-Join Parallel Framework

Textbooks / References:

1. Herbert Schildt, *Java: The Complete Reference*, 13th Edition, McGraw-Hill Education, 2024
2. Goetz, Brian. *Java concurrency in practice*. Pearson Education, 2006.
3. Kathy Sierra, Bert Bates & Trisha Gee, *Head First Java: A Brain-Friendly Guide*, Third Edition, O'Reilly. 2022
4. Barbara Liskov & John Guttag, *Program Development in Java: Abstraction, Specification and Object-Oriented Design*, Addison-Wesley, 2000.

26MAT501 MATHEMATICAL FOUNDATIONS FOR COMPUTER APPLICATIONS L-T-P-C: 3-1-0-4

Course Description

This course introduces the mathematical concepts essential for computer applications. It covers logical reasoning using propositional and predicate logic, recursive problem modeling, basic combinatorics, number theory, and graph theory. The course also emphasizes the application of statistical and optimization techniques for analyzing data and solving real-world computational problems.

Course Objectives

- To apply logical reasoning and recursion in computational problem solving.
- To use combinatorics and number theory concepts in discrete and numerical systems.
- To analyze graphs and apply basic statistical and optimization techniques to real-world data.

Course Outcomes

COs	Description
CO1	Analyze and construct logical arguments using principles of propositional and predicate logic, and develop recursive functions to model mathematical and computational problems.
CO2	Apply and interpret concepts of elementary combinatorics to solve counting problems arising in discrete mathematical structures and real-world applications.
CO3	Formulate and solve problems in number theory, including modular arithmetic, to model and analyze numerical systems relevant to theoretical and applied contexts.
CO4	Analyze graph-theoretic structures and compute centrality measures, and apply graph centrality concepts to real-world datasets for interpreting relational and network-based systems.
CO5	Apply statistical methods and basic optimization techniques to analyze real-time data and determine optimal solutions under given constraints.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	3	1	-	-	-	-	1
CO2	3	3	2	-	-	-	-	2
CO3	3	3	2	-	-	-	-	2
CO4	3	3	2	-	-	-	-	2
CO5	3	3	2	-	-	-	-	2

Syllabus

Unit I

Logic, Mathematical Reasoning and Counting: Logic, Prepositional Equivalence, Predicate and Quantifiers, Theorem Proving. Recursive Definitions, Recursive Algorithms, Basics of Counting, Pigeonhole Principle, Permutation and Combinations.

Unit II

Number theory: Divisibility- Primality Testing. GCD- Properties of the Greatest Common Divisor- Euler's Theorem. - Euclid's Algorithm-Extended Euclid's Algorithm. The Fundamental Theorem of Arithmetic. The Prime Number Theorem. Modular Arithmetic- Congruence - Arithmetic with a Prime Modulus- Multiplicative Inverses- Fermat's Little Theorem- Chinese Remainder Theorem.

Unit III

Graph Theory: Introduction to Graphs, Graph Operations, Graph and Matrices, Graph Isomorphism, Connectivity. Graph centralities: Degree and distance-based centralities. Clustering and Eigenvalue centralities. Case studies on data networks.

Unit IV

Review of basic probability and distributions.

Statistics – Bayesian statistical inference, point estimators, parameter estimators, test of hypotheses, tests of significance.

Unit V

Introduction to optimization: classical optimization, Optimality criteria – Necessary and sufficient conditions for existence of extreme point.

Direct search methods: unidirectional search, evolutionary search method, simplex search method, Introduction, Conditions for local minimization. One dimensional Search methods: Golden search method, Fibonacci method, Newton's Method, Secant Method.

Textbooks / References:

1. Kenneth H Rosen, 'Discrete Mathematics and its Applications, Seventh Edition, McGraw-Hill publications, 2007.
2. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying Ye, Probability and Statistics for Engineers and Scientists, 8th Edition, Pearson Education Asia, 2007.
3. Edwin K.P. Chong, Stanislaw H. Zak, "An Introduction to Optimization", 2nd edition, Wiley, 2013.

26CSA503	DATA STRUCTURES	L-T-P-C: 3-0-0-3

Course Description

This course is a foundation course for data structure. This course introduces linear and nonlinear data structures such as arrays, stacks, queues, linked lists, graphs, and trees and their applications.

Course Objectives

- To provide knowledge of basic data structures and their implementation
- To get familiarized with linear data structures and their applications
- To understand the concepts of nonlinear data structures and their applications
- To understand the importance of data structures for writing efficient programs and to apply appropriate data structures in problem-solving

Course Outcomes

COs	Description
CO1	Analyze and justify the selection of appropriate data structures and Abstract Data Types (ADT) for solving structured computational problems.
CO2	Develop and apply searching, sorting, and list-based data structures to organize and process data effectively in software applications.
CO3	Design and implement stack- and queue-based solutions, including recursion-oriented problems, using static and dynamic representations.
CO4	Construct and utilize hashing techniques and graph traversal methods to solve problems involving data access, relationships, and connectivity.
CO5	Design and evaluate tree-based data structures such as binary trees, binary search trees, AVL trees, B-trees, Red-Black trees, and heaps for hierarchical data organization and application development.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	3	1	-	-	-	-	1
CO2	3	3	2	-	-	-	-	2
CO3	3	3	2	-	-	-	-	2
CO4	3	3	2	-	-	-	-	2
CO5	3	3	2	-	-	-	-	2

Syllabus

Unit I

Introduction to the data structure, types of data structures, abstract data type. Array: Definition, memory representation; Applications: Searching: Linear search and Binary search; Soring: Bubble sort, Insertion sort, Selection sort, Shell sort; List: Introduction, representation; Types: Singly linked list, circular linked list, doubly linked list and their applications.

Unit II

Stacks: Introduction, Stack as an ADT, static and dynamic representation, implementation of the stack; Application of stack: polish notation representation and conversion; Recursion: Implementation of recursion, tower of Hanoi, merge sort and quick sort.

Unit III

Queues: Definition, Queue as an ADT, static and dynamic representation; Types of queues: Circular queue, double-ended queue, priority queue, applications of the queue. Hashing: Representation, search and collision handling techniques.

Unit IV

Graphs: Definition, basic terminologies, types of graphs, representation; Traversal: BFS and DFS. Trees: Definition, basic terminology, binary tree and its traversal, binary search tree, expression tree, AVL tree, B tree, Red-Black tree, heap tree and its applications.

Textbooks / References:

1. E. Horowitz & Sahni, Fundamental Data Structure, Galgotia Book Source, 1983.
2. Y. Langsam, M J. Augenstein and A M. Tenenbaum, Data Structures using C and C++. PHI Learning Pvt. Ltd.
3. Classic Data Structures by D. Samanta, Second Edition.

26CSA504	ADVANCED DBMS	L-T-P-C: 3-0-0-3

Course Description

The course gives theoretical knowledge and practical skills in the various aspects of databases including architecture, advanced queries, query processing, optimization, and the diverse types of databases including object-based databases which will help students to pursue a career in this field. The students will become more proficient in writing queries as well as in doing database design for a distributed database-oriented application.

Course Objectives

- To improve the database design skills of the students
- To develop strong foundations of DBMS for industry-level competence
- To use complex databases for scalable performance

Course Outcomes

Cos	Description
CO1	Gain confidence in writing SQL and PL/SQL
CO2	Understand the relevance of transactions and recovery, parallel and distributed databases including for large real-time applications
CO3	Learn the Object-relational Database concepts, and advanced Queries based on these concepts.
CO4	Understand JSON usage and compare it with XML
CO5	Gain knowledge in different other types of databases for handling diverse data and purposes.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	3	2	-	-	-	-	1
CO2	3	3	2	-	-	-	-	2
CO3	3	3	2	-	1	-	-	2
CO4	3	3	2	-	1	-	-	2
CO5	3	3	2	-	1	-	-	2

Prerequisites

- Basics of SQL using PostgreSQL/ MySQL/ ORACLE
- DBMS fundamentals

Syllabus

Unit I

Advanced SQL – Sub queries, Joins, Set Operations, EXISTS, views, With clause – Recursion in SQL
PL-SQL, Transaction serializability, Locking.
DB Architecture- Transaction model and properties, Transaction structure, Transaction serialization-Concurrency Control and Recovery.

Unit II

Introduction to Parallel database and I/O Parallelism, Interquery Parallelism, Intraquery Parallelism. Intraoperation Parallelism, Interoperation Parallelism – Introduction to Distributed Databases.

Unit III

Introduction to object-relational database - Complex Data Types, Structured Types and Inheritance in SQL, Table Inheritance, Array. Query planning, Evaluation and Optimization Techniques

Unit IV

JSON – syntax, datatypes, stringify, objects, schema

XML Databases: XML Data Model – DTD – XML Schema – XML Querying

Comparison of JSON with XML

Unit V

Intelligent Databases-Active Databases- Taxonomy- Applications- Design Principles for Active Rules- Temporal Databases: Overview of Temporal Databases - Spatial Databases - Deductive Databases.

Textbooks / References:

1. Database Systems Concepts, Silberschatz, Abraham, Henry F. Korth, and S.Sudarshan. McGraw-Hill, ISBN 9780078022159, 2019, 7th Edition
2. Fundamentals of Database Systems, Ramez Elmasri and Shamkant Navathe, 7th Edition, Addison Wesley, 2015 (Unit V)
3. Database Systems: The Complete Book, Hector Garcia-Molina, Jeffrey Ullman and Jennifer Widom, Second Edition, Prentice Hall, 2008
4. PostgreSQL Documentation.

26CSA581 OBJECT-ORIENTED PROGRAMMING USING JAVA LAB L-T-P-C: 0-0-2-1

Course Description

This course introduces object-oriented programming concepts and their application using Java. Students will learn to design, implement, and test robust software systems by mastering core OOP principles, and advanced features like exception handling, input/output, multithreading, and concurrency.

Course Objectives

- To equip students with a strong foundation in object-oriented programming (OOP) concepts, including abstraction, encapsulation, inheritance, and polymorphism, and their practical application in Java programming.
- To develop students' ability to design, implement, and test robust Java programs using fundamental programming constructs, data types, control flow, and object-oriented principles.
- To introduce students to advanced Java features such as exception handling, input/output operations, multithreading, and concurrency, enabling them to build efficient and responsive applications.

Course Outcomes

COs	Description
CO1	Design and implement object-oriented Java solutions by effectively applying abstraction, encapsulation, inheritance, and polymorphism.
CO2	Develop modular Java applications using classes, interfaces, inheritance hierarchies, and access control mechanisms.
CO3	Implement reliable Java programs using structured exception handling and input/output streams for real-world data processing tasks.
CO4	Design and analyze multithreaded Java applications using synchronization and inter-thread communication mechanisms.
CO5	Apply Java concurrency utilities and frameworks to develop scalable and efficient concurrent applications.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	3	3	-	-	-	-	2
CO2	3	3	2	-	-	-	-	2
CO3	3	3	2	-	-	-	-	2
CO4	3	3	2	-	-	-	-	2
CO5	3	3	2	-	-	-	-	2

Syllabus

1. Write Java programs to understand the Java program structure, compilation, execution process, and use of basic input/output statements.
2. Develop Java programs using data types, operators, conditional statements, and looping constructs to solve basic computational problems.
3. Write Java programs using arrays and strings for data manipulation and processing.
4. Implement Java programs demonstrating Object Oriented Programming concepts—abstraction, encapsulation, inheritance, and polymorphism.
5. Develop Java programs using classes and objects, including constructors, method definitions, and method invocation.
6. Write Java programs illustrating the use of access specifiers, static members, method overloading, and method

overriding.

7. Implement Java programs using inheritance hierarchies, including constructors in subclasses and the use of the super keyword.

8. Develop Java programs demonstrating abstract classes, abstract methods, final methods, final classes, and core methods of the Object class.

9. Write Java programs using interfaces, demonstrating multiple inheritance through interfaces and comparing interfaces with abstract classes.

10. Implement Java programs for exception handling, including built in exceptions, throwing and catching exceptions, and user defined (custom) exceptions.

11. Develop Java programs for console input/output and file handling, using byte streams and character streams.

12. Create Java programs using multithreading, including thread creation using the Thread class and the Runnable interface, and demonstrate the thread life cycle.

13. Write Java programs demonstrating thread synchronization, thread priorities, and inter thread communication using wait(), notify(), and notifyAll().

14. Develop concurrent Java applications using the Executor framework, thread pools, and Callable and Future interfaces.

15. Implement Java programs using the Fork Join framework and develop an integrated application combining OOP concepts, exception handling, file I/O, and concurrency.

Textbooks / References:

1. Herbert Schildt, *Java: The Complete Reference*, 13th Edition, McGraw-Hill Education, 2024

2. Goetz, Brian. *Java concurrency in practice*. Pearson Education, 2006.

3. Kathy Sierra, Bert Bates & Trisha Gee, *Head First Java: A Brain-Friendly Guide*, Third Edition, O'Reilly. 2022

4. Barbara Liskov & John Guttag, *Program Development in Java: Abstraction, Specification and Object-Oriented Design*, Addison-Wesley, 2000.

26CSA582 DATA STRUCTURES LAB L-T-P-C: 0-0-2-1

Course Description

This course is a foundation course for data structure. This course introduces linear and nonlinear data structures such as arrays, stacks, queues, linked lists, graphs, and trees and their applications.

Course Objectives

- To provide hands-on experience in implementing data structures using modular programming.
- To enable students to design efficient data handling solutions using appropriate data structures.
- To reinforce higher-order problem-solving skills through implementation-based learning.

Course Outcomes

COs	Description
CO1	Implement and test array-based and linked data structures to construct reliable data handling modules.
CO2	Design and develop stack- and recursion-based solutions for expression processing and classical algorithmic problems.
CO3	Construct and apply queue variants and hashing techniques to solve problems involving orderly data access and retrieval.
CO4	Implement and integrate tree-based data structures to model hierarchical relationships and structured data representations.
CO5	Develop and analyze graph-based solutions using traversal techniques to represent and solve connectivity-oriented problems.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	3	3	-	-	-	-	2
CO2	3	3	2	-	-	-	-	2
CO3	3	3	2	-	-	-	-	2
CO4	3	3	2	-	-	-	-	2
CO5	3	3	2	-	-	-	-	2

Syllabus

1. Write programs to perform linear search and binary search on one-dimensional arrays.
2. Implement sorting algorithms:
    - Bubble sort
    - Insertion sort
    - Selection sort
    - Shell sort
3. Implement array operations: insertion, deletion, merging, and traversal.
4. Implement singly linked list with operations: insertion, deletion, and traversal.
5. Implement doubly linked list and perform forward and backward traversal.

6. Implement circular linked list and demonstrate its applications.
7. Implement stack using array representation.
8. Implement stack using linked list representation.
9. Develop applications of stack:
   - Infix to postfix conversion
   - Postfix expression evaluation
10. Implement recursive algorithms:
    - Tower of Hanoi
11. Implement merge sort using recursion.
12. Implement quicksort using recursion.
13. Implement queue using array representation.
14. Implement circular queue and demonstrate enqueue and dequeue operations.
19. Implement binary tree and perform:
    - Inorder traversal
    - Preorder traversal
    - Postorder traversal
20. Implement binary search tree with insertion and deletion.
21. Construct and evaluate an expression tree.
22. Implement graph representation and perform:
    - Breadth First Search (BFS)
    - Depth First Search (DFS)

Textbooks / References:

1. E. Horowitz & Sahni, Fundamental Data Structure, Galgotia Book Source, 1983.
2. Y. Langsam, M J. Augenstein and A M. Tenenbaum, Data Structures using C and C++. PHI Learning Pvt. Ltd.
3. Classic Data Structures by D. Samanta, Second Edition.

26CSA580	ADVANCED DBMS LAB	L-T-P-C: 0-0-2-1

Course Description

The course gives theoretical knowledge and practical skills in the various aspects of databases including architecture, advanced queries, query processing, optimization, and the diverse types of databases including object-based databases which will help students to pursue a career in this field. The students will become more proficient in writing queries as well as in doing database design for a distributed database-oriented application.

Course Objectives

- To improve the database design skills of the students
- To develop strong foundations of DBMS for industry-level competence
- To use complex databases for scalable performance

Course Outcomes

COs	Description
CO1	Design and implement advanced SQL solutions using subqueries, joins, set operations, recursive queries, views, and PL/SQL constructs to solve complex data retrieval and manipulation problems.
CO2	Analyze and demonstrate transaction behavior by simulating concurrency issues, locking mechanisms, and serializability concepts in multi-transaction database environments.
CO3	Implement and evaluate parallel and distributed database concepts through query-level experimentation and execution-plan analysis.
CO4	Design and manipulate object-relational database features, including complex data types, structured types, table inheritance, and array processing, using advanced SQL capabilities.
CO5	Develop data-centric applications using semi-structured data models by storing, querying, and comparing JSON and XML representations and integrating them using Python or PHP.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	3	3	-	-	-	-	2
CO2	3	3	2	-	-	-	-	2
CO3	3	3	2	-	1	-	-	2
CO4	3	3	2	-	1	-	-	2
CO5	3	3	2	-	1	-	-	2

Syllabus

1. Write advanced SQL queries using subqueries, nested subqueries, and correlated subqueries.
2. Implement and apply joins (inner, outer, natural, self joins) and set operations (UNION, INTERSECT, EXCEPT) on multi-table databases.
3. Develop SQL queries using EXISTS, WITH clause (CTE), and recursive queries for hierarchical data representation.
4. Create and manipulate views, including updatable views, and analyze their role in logical data abstraction.
5. Develop PL/SQL programs using variables, control structures, procedures, functions, cursors, and exception handling.

6. Simulate transaction processing, demonstrating ACID properties, transaction states, serializability, and locking mechanisms.

7. Implement case studies illustrating concurrency control issues such as lost updates, dirty reads, non-repeatable reads, and phantom reads.

8. Design SQL queries to demonstrate parallel execution plans and observe query behavior using database explain/analysis tools.

10. Implement object-relational database features:
    - Creation of complex data types
    - Structured user-defined types
    - Methods and inheritance in SQL

11. Create and manipulate databases using table inheritance and array data types, and write queries on inherited tables.

12. Create and manipulate JSON data in databases:
    - JSON syntax and datatypes
    - Storing JSON objects
    - Querying JSON attributes

13. Design and query XML databases using:
    - XML schema / DTD
    - XPath / XQuery
    - Storage and retrieval of XML data

14. Comparative case study:
    - Compare JSON vs XML for data representation
    - Use JSON with PHP or Python to interact with the database
    - Demonstrate a small integrated application

Textbooks / References:

1. Database Systems Concepts, Silberschatz, Abraham, Henry F. Korth, and S.Sudarshan. McGraw-Hill, ISBN 9780078022159, 2019, 7th Edition
2. Fundamentals of Database Systems, Ramez Elmasri and Shamkant Navathe, 7th Edition, Addison Wesley, 2015 (Unit V)
3. Database Systems: The Complete Book, Hector Garcia-Molina, Jeffrey Ullman and Jennifer Widom, Second Edition, Prentice Hall, 2008
4. PostgreSQL Documentation.

25AVP501	MASTERY OVER MIND	L-T-P-C: 1-0-2-2

COURSE OBECTIVES

Master Over the Mind (MAOM) is an Amrita initiative to implement schemes and organize university-wide programs to enhance health and wellbeing of all faculty, staff, and students (UN SDG -3). This program as part of our efforts for sustainable stress reduction introduces immediate and long-term benefits and equips every attendee to manage stressful emotions and anxiety facilitating inner peace and harmony. With a meditation technique offered by Amrita Chancellor and world-renowned humanitarian and spiritual leader, Sri Mata Amritanandamayi Devi (Amma), this course has been planned to be offered to all students of all campuses of AMRITA, starting off with all first years, wherein one hour per week is completely dedicated for guided practical meditation session and one hour on the theory aspects of MAOM. The theory section comprises lecture hours within a structured syllabus and will include invited guest lecture series from eminent personalities from diverse fields of excellence. This course will enhance the understanding of experiential learning based on university's mission: "Education for Life along with Education for Living" and is aimed to allow learners to realize and rediscover the infinite potential of one's true Being and the fulfilment of life's goals.

COURSE OUTCOME

After successful completion of the course, students will be able to:	
S.No.	**Course Outcomes**
1.	Understand the scientific benefits of meditation. (CO1)
2.	Explain the science behind meditation and its effects on physical and mental well-being (CO2).
3.	Understand the meditation techniques to cultivate emotional intelligence and improve relationships (CO3).
4.	Learn and practice MAOM meditation in daily life (CO4).
5.	To apply the effect of meditation to compassion-driven action (CO5)

Syllabus:

Scientific benefits of Meditation (CO1)

Scientific benefits of meditation, exploring its effects on physical and mental wellbeing.
Learn about the different types of meditation practices, the essential elements of meditation, and the empirical evidence supporting its benefits.
Video resource-Swami Atmanandamrita Puri

Science Behind Meditation (CO2)

A: A preliminary understanding of the Science of meditation. What can modern science tell us about this
tradition-based method?
B: How meditation helps humanity according to what we know from scientific research
Reading 1: Does Meditation Aid Brain and Mental Health (Dr Shyam Diwakar)
Reading 2: 'Science and Spirituality.' Chapter 85 in Amritam Gamaya (2022). Mata Amritanandamayi Mission Trust.

Role of Meditation in Emotional intelligence (CO3)

Learn how meditation practices can enhance self-awareness, self-regulation, motivation, empathy, and social skills, leading to improved relationships and decision-making. Improve communication, emotional intelligence, and interpersonal skills. Logical and analytical reasoning

Practicing MA OM Meditation in Daily Life (CO4)

Guided Meditation Sessions following scripts provided (Level One to Level Five)
Reading 1: MA OM and White Flower Meditation: A Brief Note (Swami Atmananda Puri)
Reading 2: 'Live in the Present Moment.' Chapter 71 in Amritam Gamaya (2022). Mata Amritanandamayi Mission Trust.

Meditation and Compassion-driven Action (CO5)

Understand how meditation can help to motivate compassion-driven action.

Reading 1: Schindler, S., & Friese, M. (2022). The relation of mindfulness and prosocial behavior:What do we (not) know? Current Opinion in Psychology, 44, 151-156.
Reading 2: 'Sympathy and Compassion.' Chapter 100 in Amritam Gamaya (2022). Mata Amritanandamyi Mission Trust.

Textbooks / References:

1. Mata Amritanandamayi Devi, "Cultivating Strength and vitality," published by Mata Amritanandamayi Math, Dec 2019
2. Swami Amritaswarupananda Puri," The Color of Rainbow "published by MAM, Amritapuri. 3. Craig Groeschel, "Winning the War in Your Mind: Change Your Thinking, Change Your Life" Zondervan Publishers, February 2019
4. R Nagarathna et al, "New Perspectives in Stress Management "Swami Vivekananda Yoga Prakashana publications, Jan 1986
5. Swami Amritaswarupananda Puri "Awaken Children Vol 1, 5 and 7 - Dialogues with Amma on Meditation", August 2019
6. Swami Amritaswarupananda Puri "From Amma's Heart - Amma's answer to questions raised during world tours" March 2018
7. Secret of Inner Peace- Swami Ramakrishnananda Puri, Amrita Books, Jan 2018.
8. Mata Amritanandamayi Devi "Compassion: The only way to Peace:Paris Speech", MA Center, April 2016.
9. Mata Amritanandamayi Devi "Understanding and collaboration between Religions", MA Center, April 2016.
10. Mata Amritanandamayi Devi "Awakening of Universal Motherhood: Geneva Speech" M A center, April 2016.

24LSK502	LIFE SKILLS - I	L-T-P-C: 0-0-2-1

Course Description

This course is designed to prepare students for a smooth transition from campus to corporate life by developing essential soft skills, communication abilities, aptitude, and professional competencies. The course emphasizes effective communication, self-awareness, goal setting, presentation skills, verbal ability, reasoning, quantitative aptitude, and professional etiquette required to succeed in recruitment processes and workplace environments.

Course Objectives

- Develop effective communication and listening skills for professional environments
- Build assertiveness, self-confidence, and positive self-perception
- Apply goal-setting and time-management strategies for personal and professional growth
- Enhance presentation and public speaking skills
- Strengthen verbal ability, grammar, and written communication
- Improve logical reasoning and quantitative aptitude for competitive assessments

Course Outcomes (COs)

CO Code	Course Outcome
CO1	Communicate effectively using verbal, non-verbal, listening, and presentation skills in professional environments
CO2	Demonstrate self-confidence, assertiveness, ethical behavior, and effective interpersonal skills
CO3	Apply logical reasoning and quantitative aptitude skills to solve analytical problems

CO Code	Course Outcome
CO4	Use professional writing, data interpretation, and time-management skills for workplace effectiveness

CO–PO Mapping

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	3	2	3	1	2	3
CO2	2	2	2	3	3	1	3	3
CO3	2	3	1	1	1	3	2	2
CO4	2	2	3	2	2	2	3	3

Soft Skills

Introduction to 'Campus to Corporate Transition':

Communication and listening skills:

Communication process, barriers to communication, verbal and non-verbal communications, elements of effective communication, listening skills, empathetic listening, role of perception in communication.

Assertiveness Skills:

The concept, assertiveness and self-esteem, advantages of being assertive, assertiveness and organizational effectiveness.

Self-perception and Self-confidence:

The locus of control (internal v/s external), person perception, social perception, attribution theories-self presentation and impression management, the concept of self and self-confidence, how to develop self-confidence.

Goal Setting:

The concept, personal values and personal goals, goal setting theory, six areas of goal setting, process of goal setting: SMART goals, how to set personal goals.

Time Management:

The value of time, setting goals/ planning and prioritizing, check the time killing habits, procrastination, tools for time management, rules for time management, strategies for effective time management.

Presentation Skills:

The process of presentation, adult learning principles, preparation and planning, practice, delivery, effective use of voice and body language, effective use of audio-visual aids, dos and don'ts of effective presentation.

Public speaking-an art, language fluency, the domain expertise (Business GK, Current affairs), self-confidence, the audience, learning principles, body language, energy level and conviction, student presentations in teams of five with debriefing

Verbal Skills

Vocabulary:

Familiarize students with the etymology of words, help them realize the relevance of word analysis and enable them to answer synonym and antonym questions. Create an awareness about the frequently misspelt words, commonly confused words and wrong form of words in English.

Grammar:

Train students to understand the nuances of English Grammar and thereby enable them to spot grammatical errors and punctuation errors in sentences.

Reasoning: Stress the importance of understanding the relationship between words through analogy questions and learn logical reasoning through syllogism questions. Emphasize the importance of avoiding the gap (assumption) in arguments/ statements/ communication.

Oral Communication Skills:

Aid students in using the gift of the gab to improve their debating skills.

Writing Skills:

Introduce formal written communication and keep the students informed about the etiquettes of email writing. Make students practice writing emails specially composing job application emails.

<u>Aptitude Skills</u>

Numbers:

Types, Power Cycles, Divisibility, Prime, Factors & Multiples, HCF & LCM, Surds, Indices, Square roots, Cube Roots and Simplification.

Percentage:

Basics, Profit, Loss & Discount, and Simple & Compound Interest.

Ratio, Proportion & Variation:

Basics, Alligations, Mixtures, and Partnership, Averages and Weighted Average.

Time and Work:

Basics, Pipes & Cistern, and Work Equivalence.

Time, Speed and Distance:

Basics, Average Speed, Relative Speed, Boats & Streams, Races and Circular tracks.

Statistics:

Mean, Median, Mode, Range, Variance, Quartile Deviation and Standard Deviation.

Data Interpretation:

Tables, Bar Diagrams, Line Graphs, Pie Charts, Caselets, Mixed Varieties, and other forms of data representation.

Equations:

Basics, Linear, Quadratic, Equations of Higher Degree and Problems on ages.

22ADM501 GLIMPSES OF INDIAN CULTURE L-T-P-C: 2-0-1 P/F

Course Objectives

To identify and analyze the cultural practices, values, and beliefs of modern India, recognizing the ways in which cultural identities intersect with systems of power and oppression. Moreover, the students will be able to reflect on their own cultural biases and assumptions to work towards becoming more culturally competent and inclusive in their personal and professional lives.

Course Outcomes

COs	Description
CO1	Understand the relevance of legendary people who are involved with restoring balance and harmony or guiding humanity toward spiritual liberation.
CO2	Demonstrate an understanding of the historical and cultural contexts that have shaped the role of women in society.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								

CO1	-	-	-	-	2	-	-	3
CO2	-	-	-	-	2	-	-	3

Syllabus

Unit 1
What is Sanatana Dharma; The Heritage of Scriptures ; The idea of Īśvara; Guru Tattva and Avatara Tattva

Unit 2
Theory of Karma; Purusharthas; Sanyasa; Yajna; Symbolism

Unit 3
Understanding Nataraja; Temples: The Cradle of Culture;

Unit 4
Other Heterodox Systems in India; Sadhana

References:

1. Glimpses of Indian Culture
2. Sanatana Dharma- The Eternal Truth (A compilation of Amma's teachings on Indian Culture)

SEMESTER II

26CSA511 DESIGN AND ANALYSIS OF ALGORITHMS L-T-P-C: 3-1-0-4

Course Description

The primary objective of this course is to introduce the design and analysis of algorithms. Students will learn how to design an efficient algorithm with a computable problem. This includes modelling the problem, selecting appropriate algorithm design techniques, analyzing the efficiency of algorithms and proving the correctness of algorithms.

Course Objectives

- To learn different algorithm design techniques and design algorithms using the same.
- To analyze an algorithm and determine its time complexity.
- To learn methods to deal with intractable problems.

Course Outcomes

COs	Description
CO1	Analyze and estimate the asymptotic complexity of algorithms and compare alternative algorithmic solutions to justify appropriate choices for given problem scenarios.
CO2	Derive, solve, and interpret recurrence relations that describe the performance of divide-and-conquer algorithms.
CO3	Apply the greedy algorithmic strategy to design correct and efficient solutions for optimization and scheduling problems.
CO4	Formulate and solve problems using dynamic programming techniques to obtain optimal solutions for multistage decision problems.

CO5	Analyze intractable and NP-complete problems and devise suitable approximation, heuristic, or reduction-based approaches to address computational limitations.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	3	2	-	1	-	-	1
CO2	3	3	3	-	1	-	-	1
CO3	3	3	3	-	1	-	-	1
CO4	3	3	3	-	1	-	-	1
CO5	3	3	3	-	1	-	-	2

Prerequisites

Data Structures

Syllabus

Unit I

Notion of an Algorithm – Fundamentals of Algorithmic Problem Solving – Important Problem Types – Fundamentals of the Analysis of Algorithmic Efficiency –Asymptotic Notations and growth rate. Sorting Algorithms and analysis: Bubble sort, Selection sort, Insertion sort, Shell sort. Recurrence Relations- substitution & master's method.

Unit II

Divide and conquer strategy – finding minimum and maximum, integer multiplication, binary search, binary search tree. Merge sort, Quick sort, heap sort, and its analysis. Strassen's algorithm for matrix multiplication, maximum subarray.

Unit III

Greedy Method – Knapsack problem, Job sequencing with deadlines, optimal merge pattern. Minimum spanning tree – Prim's and Kruskal's algorithm. Single source shortest path – Dijkstra's algorithm, Bellman-Ford algorithm. Huffman Coding, TSP.

Unit IV

Dynamic programming – Principle of optimality, knapsack problem, matrix chain multiplication, longest common subsequence problem, optimal binary search tree, traveling salesman problem. All pair shortest path – Floyd Warshal algorithm.

Unit V

Backtracking and Branch and Bound – Queen's problem, sum of subset, graph coloring, Hamiltonian cycle. The class P and NP, polynomial reduction, NP-completeness problem, and NP-Hard problem. Reduction – 3CNF SAT to Clique, Clique to Vertex cover.

Textbooks / References:

1. Analysis of Algorithms, Jeffrey J McConnel, Jones and Bartlett Publishers, Inc, 2nd Revised edition, 2 November 2007
2. Introduction to the Design and Analysis of Algorithms, Anany Levitin, Third Edition, Pearson Education, 2012
3. Introduction to Algorithms, Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Third Edition, Prentice-Hall of India Private Limited; 2009.

26CSA512 PRINCIPLES AND IMPLEMENTATION OF DESIGN PATTERNS L-T-P-C: 3-0-2-4

Course Description

This course introduces the intent, structure, and application of design patterns, followed by in-depth study of creational (Abstract Factory, Factory Method, Prototype, Singleton), structural (Adapter, Bridge, Composite, Decorator, Façade, Proxy), and behavioral (Command, Iterator, Mediator, Observer, State, Template Method) patterns with comparative analysis and case studies to support reusable, robust software design. It covers core software engineering principles and object-oriented design patterns for building scalable and maintainable systems. It also deals with software classification, layered architecture, SDLC, generic and agile process models (Scrum, XP), process assessment, design engineering, and reengineering.

Course Objectives

- To understand the idea of software engineering process models by analyzing SDLC, layered architecture, agile methods (Scrum, XP), and reengineering techniques for systematic software development.
- To identify, compare, and implement object-oriented design patterns including creational, structural, and behavioral patterns to produce scalable, reusable, and maintainable software architectures.
- To develop robust software designs using pattern-based solutions through case studies and practical analysis to improve flexibility, performance, and quality in real-world applications.

Course Outcomes

COs	Description
CO1	Understand and apply software engineering methods, layered technology, process frameworks, and requirements analysis with design engineering using agile methodology along with reengineering.

CO2	Understand the concept of Design patterns and its importance in gaining behavioral knowledge of the problem and its solutions using Creational, Structural design patterns.
CO3	Understand and apply common creational design patterns to incremental and iterative development, and identify appropriate object creation mechanisms to design and propose effective solutions for given problems.
CO4	Understand and apply common design patterns to incremental/iterative development. To identify appropriate behavioral patterns for the design to propose solutions to the given problem.
CO5	Understand the need for programming by using basic design principles in solving real-life problems or case studies.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	-	2	3	1	1	1	3	3
CO2	-	2	3	2	2	1	3	3
CO3	3	2	3	2	3		3	
CO4	3	3	3	3	3		3	
CO5	3	2	2	2	3		3	

Prerequisites
Object oriented programming languages

Syllabus

Unit I
Software Engineering – Introduction - Software Classification - Layered Technology - SDLC - Generic Process Model, Perspective Models - Agile Process Models – Scrum and Extreme Programming (XP) - Design Engineering - Reengineering

Unit II
Introduction: What Is a Design Pattern? Describing Design Patterns, The Catalogue of Design Patterns, Organizing the Catalogue, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

Unit III
Creational Patterns: Abstract Factory, Factory Method, Prototype, Singleton – Comparisons of various creational patterns – Discussion of Case studies – ex: Cross-Platform UI Framework – Banking solution - Cloud Notification Service etc.,

Unit IV
Structural Patterns: Adapter, Bridge, Composite, Decorator, Façade, and Proxy – Comparisons of various structural patterns - Discussion of Case studies – ex: Payment gateways, ERP migrations etc.,

Unit V
Behavioral Patterns: Command, Iterator, Mediator, Observer, State and Template Method. Comparisons of various behavioral patterns - Discussion of Case studies – ex: smart homes, Smart Factory Automation etc.,

Textbooks / References:

1. Roger S. Pressman, "Software Engineering-A Practitioner's Approach", Seventh Edition, Tata McGraw-Hill, 2010.
2. Richard Fairley, "Software Engineering concepts", Tata McGraw-Hill Publishing Company Pvt. Ltd., Ninth Edition

3. Patterns in JAVA Vol-I By Mark Grand, Wiley Dream Tech.
4. Head First Design Patterns by Eric Freeman-O'Reilly-spd

26CSA583	PROBLEM FORMULATION AND RESEARCH TOOLS	L-T-P-C: 0-0-2-1

Course Description

This course provides an introduction to research fundamentals, including ethical principles, challenges, and core components of the research process. It also offers practical guidance on research skills such as academic writing, paper preparation, and effective research presentation.

Course Objectives

- To introduce the basic elements of research
- Hands-on experience with tools used for research organization
- Understand the various elements of research paper writing

Course Outcomes

COs	Description
CO1	Understand the basic framework of the research process.
CO2	Identify various sources of information for literature review.
CO3	Gain a practical understanding of the various tools and techniques used for conducting research.
CO4	Conduct a research study from its inception to its report and study Ethical issues in research.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								

CO1	-	-	-	1	3	-	-	3
CO2	-	-	-	2	3	-	1	3
CO3	-	-	-	3	3	-	2	3
CO4	-	-	-	3	3	1	3	3

Syllabus

Unit I

Research: Meaning, Purpose, Types of Research, Steps in Research, Identification, Selection and Formulation of Research Problem, Research Questions, Research Design, Research Methods in Computer Science:- Formal Methods: Formal Specification, Algorithm, and Complexity; Building Artefacts: Proof of Performance, Proof of Concept, and Proof of Existence; Process Methodology: Methods for Software Engineering and Human-Computer Interaction, Cognitive Processes, Interactive Games, Social Networks, and Web Analytics.

Unit II

Review of Literature: - source selection, citation and indexes, impact factor, tools for literature organization.

Unit III

Brief overview of Measurement, Data, and Analytics, Elements of Theoretical Research, Elements of Preparing a Paper and a Thesis: Title, Abstract, Keywords, Acknowledgements, Symbols and Abbreviations, Introduction, Literature Review, Materials and Methods, Mathematical Materials, Graphical and Tabular Presentation, Results and Discussion, Conclusion, Interpretation, Generalization, Scope for Future Work, Citations and List of References, and Appendixes

Unit IV

Research Ethics, Plagiarism, and Their Prevention.

Textbooks / References:

1. Huma Anwar Ehtiram Raza Khan: “Research Methods of Computer Science”, Laxmi Publications, 2019
2. Keshav, Srinivasan. "How to read a paper." ACM SIGCOMM Computer Communication Review 37, no. 3 (2007): 83-84.
3. https://en.wikibooks.org/wiki/LaTeX

26CSA584	DESIGN AND ANALYSIS OF ALGORITHMS LAB	L-T-P-C: 0-0-2-1

Course Description

The primary objective of this course is to introduce the design and analysis of algorithms. Students will learn how to design an efficient algorithm with a computable problem. This includes modelling the problem, selecting appropriate algorithm design techniques, analyzing the efficiency of algorithms and proving the correctness of algorithms.

Course Objectives

- To learn different algorithm design techniques and design algorithms using the same.
- To analyze an algorithm and determine its time complexity.
- To learn methods to deal with intractable problems.

Course Outcomes

COs	Description
CO1	Analyze and compare searching and sorting algorithms to determine their suitability for different problem contexts and data characteristics.
CO2	Design and implement algorithmic solutions using greedy and dynamic programming paradigms to solve structured optimization and decision-based problems.
CO3	Apply and analyze graph- and tree-based algorithms to model, solve, and interpret problems involving hierarchical and network-oriented data structures.

CO-PO Mapping

PO/PSO							PO7	PO8

CO	PO1	PO2	PO3	PO4	PO5	PO6		
CO1	3	3	2	-	1	-	-	1
CO2	3	3	3	-	1	-	-	1
CO3	3	3	3	-	1	-	-	1
CO4	3	3	3	-	1	-	-	1
CO5	3	3	3	-	1	-	-	2

Syllabus

Searching algorithms: Linear, Binary, and Hashing and its analysis. Sorting Algorithms: Bubble sort, insertion sort, selection sort, merge sort, quick sort, and its analysis. Greedy Methodology: Fractional knapsack, Job sequencing, Optimal Merge pattern with time complexity. Dynamic Programming: 0 or 1 knapsack problem, TSP and its analysis. Graph and tree applications: BFS, DFS, BST, Minimum Spanning Tree, Dijkstra's, Floyd's algorithm with time complexity.

Textbooks / References:

1. Analysis of Algorithms, Jeffrey J McConnel, Jones and Bartlett Publishers, Inc, 2nd Revised edition, 2 November 2007
2. Introduction to the Design and Analysis of Algorithms, Anany Levitin, Third Edition, Pearson Education, 2012
3. Introduction to Algorithms, Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Third Edition, Prentice-Hall of India Private Limited; 2009.

24LSK512 LIFE SKILLS – II L-T-P-C: 0-0-2-1

Course Description

This course is designed to prepare students for a smooth transition from campus to corporate life by developing essential soft skills, communication abilities, aptitude, and professional competencies. The course emphasizes effective communication, self-awareness, goal setting, presentation skills, verbal ability, reasoning, quantitative aptitude, and professional etiquette required to succeed in recruitment processes and workplace environments.

Course Objectives

- Develop effective communication and listening skills for professional environments
- Build assertiveness, self-confidence, and positive self-perception
- Apply goal-setting and time-management strategies for personal and professional growth
- Enhance presentation and public speaking skills
- Strengthen verbal ability, grammar, and written communication
- Improve logical reasoning and quantitative aptitude for competitive assessments

Course Outcomes (COs)

CO Code	Course Outcome
CO1	Communicate effectively using verbal, non-verbal, listening, and presentation skills in professional environments
CO2	Demonstrate self-confidence, assertiveness, ethical behavior, and effective interpersonal skills
CO3	Apply logical reasoning and quantitative aptitude skills to solve analytical problems

CO Code	Course Outcome
CO4	Use professional writing, data interpretation, and time-management skills for workplace effectiveness

CO–PO Mapping

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	3	2	3	1	2	3
CO2	2	2	2	3	3	1	3	3
CO3	2	3	1	1	1	3	2	2
CO4	2	2	3	2	2	2	3	3

Soft Skills

Interpersonal Skills:

Ability to manage conflict, flexibility, empathetic listening, assertiveness, stress management, problem solving, understanding one’s own interpersonal needs, role of effective teamwork in organizations.

Group Problem Solving:

The process, the challenges, the skills and knowledge required for the same.

Conflict Management:

The concept, its impact and importance in personal and professional lives, (activity to identify personal style of conflict management, developing insights that helps in future conflict management situations.)

Team Building and Working Effectively in Teams:

The concept of groups (teams), different stages of group formation, process of team building, group dynamics, characteristics of effective team, role of leadership in team effectiveness. (Exercise to demonstrate the process of emergence of leadership in a group, debrief and reflection), group discussions.

Interview Skills:

What is the purpose of a job interview, types of job interviews, how to prepare for an interview, dos and don’ts of interview, One on one mock interview sessions with each student

Verbal Skills

Vocabulary:

Help students understand the usage of words in different contexts. Stress the importance of using refined language through idioms and phrasal verbs.

Grammar:

Enable students to identify poorly constructed sentences or incorrect sentences and improvise or correct them.

Reasoning:

Facilitate the student to tap her/his reasoning skills through critical reasoning questions and logical ordering of sentences.

Reading Comprehension:

Enlighten students on the different strategies involved in tackling reading comprehension questions.

Public Speaking Skills:

Empower students to overcome glossophobia and speak effectively and confidently before an audience.

Writing Skills:

Practice closet tests that assess basic knowledge and skills in usage and mechanics of writing such as punctuation, basic grammar and usage, sentence structure and rhetorical skills such as writing strategy, organization, and style.

Aptitude Skills

Sequence and Series:

Basics, AP, GP, HP, and Special Series.

Geometry:

2D, 3D, Coordinate Geometry, and Heights & Distance.

Permutations & Combinations:

Basics, Fundamental Counting Principle, Circular Arrangements, and Derangements.

Probability:

Basics, Addition & Multiplication Theorems, Conditional Probability and Bayes' Theorem.

Logical Reasoning I:

Arrangements, Sequencing, Scheduling, Venn Diagram, Network Diagrams, Binary Logic, and Logical Connectives, Clocks, Calendars, Cubes, Non-Verbal reasoning and Symbol based reasoning.

Logical Reasoning II:

Blood Relations, Direction Test, Syllogisms, Series, Odd man out, Coding & Decoding, Cryptarithmetic Problems and Input - Output Reasoning.

Data Sufficiency:

Introduction, 5 Options Data Sufficiency and 4 Options Data Sufficiency. Campus recruitment papers: Discussion of previous year question papers of all major recruiters of Amrita Vishwa Vidyapeetham.

Miscellaneous:

Interview Puzzles, Calculation Techniques and Time Management Strategies.

Semester III

26CSA698 DISSERTATION PHASE I 6 Credits

Course Description

Dissertation Phase I introduces students to research-oriented project work with emphasis on problem identification, literature survey, research methodology, system analysis, feasibility study, and ethical research practices. The course prepares students to conceptualize, plan, and design a substantial research-based computing project, laying a strong foundation for implementation and evaluation in Phase II.

Course Objectives

- Identify relevant research problems in emerging computing domains
- Conduct a systematic and critical review of research literature
- Apply appropriate research methodologies and ethical practices
- Analyze system requirements and design suitable solution frameworks
- Prepare feasibility studies and structured research documentation

Course Outcomes (COs)

CO Code	Description
CO1	Identify and define research problems and formulate objectives

CO Code	Description
CO2	Review, analyze, and synthesize research literature
CO3	Apply ethical research practices and appropriate methodologies
CO4	Analyze system requirements and design solution models
CO5	Prepare structured technical and research documentation

CO-PO Mapping

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	1	1	–	–
CO2	3	3	2	3	1	1	–	–
CO3	2	2	1	3	1	3	–	–
CO4	3	3	3	2	2	–	–	–
CO5	2	2	2	1	1	2	–	2

Syllabus

1 Orientation – Introduction to research areas, dissertation guidelines, awareness of research tools

2 Topic Finalization – Selection of project domain and finalization of dissertation title

3 Research Platforms – Usage of Google Scholar, ResearchGate, citation management tools

4 Literature Survey – Techniques and strategies for conducting an effective literature review

5 Review Paper – Review paper analysis, understanding structure, and presentation

6 Gap Identification – Identification of research gaps and novelty in the chosen domain

7 Review Preparation – Preparation and presentation for Review I

8 Problem Definition – Formulation of problem statement, objectives, and research questions

9 Research Methodology – Quantitative, qualitative, and experimental research methods

10 Ethics – Research ethics, plagiarism issues, data privacy, and responsible research practices

11 System Analysis – Requirement analysis, functional and non-functional modeling

12 System Design – UML diagrams, OOAD / SSAD design approaches

13 Feasibility Study – Technical, economic, and operational feasibility analysis

14 Documentation – Dissertation report structure, formatting, and standard guidelines

15 Final Review – Preparation for Review II and submission of documentation

Semester IV

26CSA699	DISSERTATION PHASE II	12 Credits

Course Description

Dissertation Phase II is a continuation of Phase I and focuses on the implementation, experimentation, evaluation, and validation of the research work proposed earlier. The course emphasizes system development, result analysis, documentation, and dissemination of research outcomes.

Course Objectives

- Implement the proposed research solution identified in Phase I
- Integrate system modules and refine the developed model or application
- Conduct experiments and evaluate results using appropriate metrics
- Analyze, interpret, and validate experimental outcomes
- Prepare a research paper suitable for conference or journal submission
- Document the complete dissertation work and present findings effectively

Course Outcomes

COs	Description
CO1	Implement the proposed system or model using appropriate computing tools and technologies
CO2	Integrate system modules and perform functional and performance testing
CO3	Conduct experiments and analyze results using suitable evaluation metrics
CO4	Compare proposed solutions with existing approaches and validate outcomes
CO5	Prepare a complete dissertation report and communicate research findings effectively

CO-PO Mapping

CO \ PO	**PO1**		**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**
CO1	3		3	3	2	2	–	–	–
CO2	3		3	2	3	2	–	–	–
CO3	2		3	2	3	2	1	–	–
CO4	2		3	2	3	1	1	–	–
CO5	2		2	2	1	1	3	–	2

Syllabus

1. Environment setup and configuration of tools, libraries, and frameworks required for implementation
2. Algorithm / model implementation based on the proposed methodology
3. Module-wise development of the system and integration of individual components
4. Code optimization, version control practices, and technical documentation of implementation
5. Functional testing and debugging of the developed system
6. Performance testing and refinement of system parameters
7. Review I – Evaluation of implementation progress, testing, and experimental results
8. Experimental design and preparation of datasets for evaluation
9. Selection and application of appropriate evaluation metrics
10. Result analysis and interpretation of experimental outcomes
11. Comparative analysis with existing or baseline approaches, validation of results, and discussion of advantages and limitations
12. Research paper preparation: structure, abstract, methodology, results, citation style, and plagiarism checking
13. Dissertation report writing: formatting standards, results discussion, conclusion, and future scope
14. Review II – Final review including research paper, dissertation documentation, system demonstration, and viva-voce

ELECTIVES – 3 Credits

AI & DS STREAM

26CSA530	DEEP LEARNING	L-T-P-C: 3-0-0-3

Course Description

This course builds from a one node neural network to a multiple feature, multiple output neural networks. After an understanding of how neural networks work and the parameters that control deep learning systems, building of deep learning neural networks and various applications.

Course Objectives

- Understand the context of neural networks and deep learning
- Know how to use a neural network
- Understand the data needs of deep learning
- Have a working knowledge of neural networks and deep learning.
- Explore the parameters for neural networks

Course Outcomes

Cos	Description

CO1	Identify the roles of neural networks in deep learning
CO2	Design of different Convolutional Neural Network architectures to specific problem solving
CO3	Implement various unsupervised deep learning techniques for feature learning
CO4	Design convolution networks for computer vision problems

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	3	-	-	-	-	-
CO2	3	3	1	-	1	-	-	-
CO3	2	3	2	-	1	-	-	-
CO4	3	1	3	1	1	-	-	-

Syllabus

Unit I
Introduction to Deep Neural Networks: Feed forward Neural networks. Gradient descent and the back propagation algorithm, Intuition of Neural Networks Loss functions, Optimization, Unit saturation, aka the vanishing gradient problem, and ways to mitigate it.

Unit II
Convolutional Neural Networks, Training Neural Networks, Understanding Neural Networks Through Deep Visualization and Recurrent Neural Networks: Architectures, convolution / pooling layers, LSTM, Encoder Decoder architectures.

Unit III
Deep Unsupervised Learning: Auto encoders (standard, sparse, denoising, contractive, etc), variational Auto encoders, denoising encoders, Adversarial Generative Networks.

Unit IV
Deep Belief Networks: Energy Based Models, Restricted Boltzmann Machines, Sampling in an RBM. Applications of deep neural networks in handwritten character recognition, face recognition, semantic web, social networks.

Textbooks / References:

1. Domingos, Pedro. "A few useful things to know about machine learning." Communications of the ACM 55.10 (2012): 78-87.
2. Li Fei-Fei (Stanford), Rob Fergus (NYU), Antonio Torralba (MIT), “Recognizing and Learning Object Categories” (Awarded the Best Short Course Prize at ICCV 2005).
3. Baydin, AtilimGunes, Barak A. Pearlmutter, and Alexey AndreyevichRadul. "Automatic differentiation in machine learning: a survey." arXiv preprint arXiv:1502.05767 (2015).
4. Bengio, Yoshua. "Practical recommendations for gradient-based training of deep architectures." Neural Networks: Tricks of the Trade. Springer Berlin Heidelberg, 2012. 437-478.
5. LeCun, Yann A., et al. "Efficient backprop." Neural networks: Tricks of the trade. Springer Berlin Heidelberg, 2012. 9-48.
6. Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps." arXiv preprint arXiv:1312.6034 (2013).
7. Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." Computer vision–ECCV 2014. Springer International Publishing, 2014. 818- 833.
8. Springenberg, Jost Tobias, et al. "Striving for simplicity: The all convolutional net." arXiv preprint arXiv:1412.6806 (2014).

9. Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International Journal of Computer Vision 115.3 (2015): 211-252.

26CSA531	LINEAR ALGEBRA AND APPLICATIONS	L-T-P-C: 3-0-0-3

Course Description

This course provides students with an in-depth understanding of modern operating system technologies, implementation techniques and research issues. The course enables students to specialize in operating Systems by exposing the recent developments and research trends in the field. This course covers broad range of topics which includes Unix architecture, design of modern operating systems, resource sharing and scheduling, software and hardware interaction, memory management, distributed and real time system behaviors.

Course Objectives

- To understand how the principles of quantum physics give rise to computational models and algorithms.
- To explore potentials of quantum computation in enabling artificial intelligence systems.
- To learn computational algorithms as described by quantum computation

Course Outcomes

Cos	Description
CO1	Apply the fundamental concepts of vector space, Basis and Dimension
CO2	Understand linear transformation and its applications
CO3	Understand the concepts of inner products, orthogonality and projections
CO4	Apply the concepts of Eigen Values, Eigen Vectors & Diagonalization.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	1	2	1	-	-	-	-	2
CO2	2	3	2	-	-	-	-	2
CO3	1	3	2	1	-	-	-	-
CO4	2	3	3	1	-	-	-	-

Syllabus

Unit I
Vector space:
Vectors, Vector spaces - Sub spaces, Four fundamental subspaces, Linear independence, Basis and Dimensions

Unit II
Linear Transformations:
Linear Transformations, Matrix representation, Kernel, Range, Characteristic Roots, Characteristic Vector, Matrix of a Linear Transformation. Rank Nullity Theorem, Relation between matrices and linear transformations - Kernel and range of a linear transformation

Unit III
Norms, Inner product and Orthogonality, Vector Norms, Matrix Norms, Inner product, Orthogonal vectors, Gram-Schmidt procedure, Orthogonal projection.

Unit IV
Eigen values and Eigen vectors: Elementary properties of Eigen Systems, Diagonalization, Orthogonal Diagonalization, Functions of diagonalizable matrices, Normal Matrices

Textbooks/References:

1. Carl. D. Meyer, ‘Matrix Analysis and Applied Linear Algebra’, SIAM publications
2. David C. Lay, Linear Algebra and its Applications, Pearson.
3. Gilbert Strang, “Linear Algebra and Its Applications”, Fourth Edition, Cengage, 2006.

26CSA532	ARTIFICIAL INTELLIGENCE	L-T-P-C: 3-0-0-3

Course Description

This course introduces the historical perspective and core principles of AI emphasizing problem solving, inference, perception, knowledge representation, and learning. It provides students an understanding of how intelligent systems reason, learn, and make decisions in real-world applications.

Course Objectives

- Illustrate the reasoning techniques for handling Uncertain Knowledge
- Explore the explanation-based learning to solve AI problems
- Demonstrate the applications of soft computing and Evolutionary Computing algorithms
- To apply knowledge representation and reasoning techniques to real-world problems.
- To understand AI ethical issues and responsibilities.

Course Outcomes

Cos	Description
CO1	Understanding the basics of AI and its significance and challenges in designing search problems.
CO2	Understand and apply various search algorithms in real world problems.
CO3	Analyse the fundamentals of knowledge representation, inference and theorem proving.
CO4	To analyse reasoning processes under incomplete and/or uncertain information.
CO5	Understand AI Ethical principles and social implications

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	2	-	1	-	-	-
CO2	2	1	1	-	1	2	1	-
CO3	3	2	1	-	-	-	-	-
CO4	1	2	1	-	1	2	1	1
CO5	2	-	-	-	-	-	1	-

Prerequisites

- Machine Learning
- Programming languages
- Probability

Syllabus

Unit I

Artificial Intelligence – Basics, The AI Problems – The Underlying Assumption – What is an AI technique – Criteria for Success. Problems, Problem Spaces and Search – Defining Problem as a State Space Search – Production Systems – Problem Characteristics – Production System Characteristics – Issues in the design of Search Programs.

Unit II

Heuristic Search Techniques - Generate – and – Test – Hill Climbing – Best-First Search – Problem Reduction – Constraint Satisfaction - Game Playing - The Minimax Search Procedure – Adding Alpha-Beta Cut-offs. Means - Ends Analysis. Knowledge Representation issues – Representations and Mapping - Approaches to knowledge Representation – Issues in knowledge Representation – The Frame Problem.

Case study based on search algorithms (to be considered as part of continuous assessment).

Unit III

Using Predicate Logic – Representing simple facts in Logic – Representing Instance and Isa Relationship – Computable Functions and Predicates – Resolution – Natural Deduction. Representing Knowledge Using Rules – Procedural versus Declarative knowledge – Logic Programming – Forward versus Backward Reasoning – Matching – Control Knowledge.

Case study based on reasoning (to be considered as part of continuous assessment).

Unit IV

Reasoning under Uncertainty – Introduction to Non-monotonic Reasoning – Augmenting a Problem Solver – Implementation: Depth - First Search, Fuzzy Logic.

Applications of artificial intelligence, DNA sequencing using AI techniques.

AI Ethics- Algorithmic bias and fairness, Explainability and transparency, Privacy and robustness

Textbooks / References:

1. Artificial Intelligence (Second Edition) – Elaine Rich, Kevin knight (Tata McGraw-Hill)
2. Artificial Intelligence: A Modern Approach (3rd Edition) - Stuart Russell and Peter Norvig(Pearson)
3. A Guide to Expert Systems – Donald A. Waterman (Addison-Wesley)
4. Principles of Artificial Intelligence – Nils J. Nilsson (Narosa Publishing House)
5. Introduction to Artificial Intelligence – Eugene Charnaik, Drew McDermott (Pearson Education Asia).

26CSA533	DATABASE ADMINISTRATION	L-T-P-C: 3-0-0-3

Course Description

This course provides a comprehensive introduction to Database Administration, focusing on the effective organization, management, and maintenance of database systems. It covers the principles and practices involved in administering a DBMS, including database creation, configuration, user and security management, backup and recovery strategies, performance monitoring and tuning, and routine maintenance operations. The course emphasizes hands-on experience with administrative tools and SQL to ensure data integrity, availability, security, and efficient retrieval of information in real-world database environments.

Course Objectives

- The course's objective is to present an introduction to database management systems, emphasizing how to organize, maintain and retrieve - efficiently, and effectively - information from a DBMS.
- This course focuses on the administration of a DBMS including creation, management, maintenance, and operation of a database management system.

Course Outcomes

Cos	Description
CO1	Establish an in-depth understanding of database administration using the oracle dbms interfaces.
CO2	Analyze and model requirements and constraints for installing, configuring, and tuning a dbms.
CO3	Develop methods for implementing security, back-up and recovery measures.
CO4	Apply database storage structures and understand network responsibilities for dba.
CO5	Apply the knowledge and skills required to monitor the performance of the database computing.

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	2	-	-	1	-	-
CO2	2	2	2	-	-	-	-	-
CO3	1	3	2	-	-	-	-	-
CO4	3	2	2	-	-	-	-	-
CO5	2	3	3	-	-	-	-	-

Prerequisites

- Database Management Systems
- Structured Query Languages

Syllabus

Unit I

Introduction: DBMS architecture-data independence-Users- DBA roles-SQL *PLUS Overview: Producing more readable outputs, accepting values at runtime, using iSQL *Plus-Introduction to DML Statements, truncating a table, Transaction control language, Managing Views: Creating and modifying views, using views, Inserting, Updating and deleting data through views.

Unit II

User Access and Security: Creating and modifying use accounts, creating and using roles, granting and revoking privileges, managing user groups with profiles-Oracle Overview and Architecture: storage structures, Oracle memory structures, Oracle background processes, connecting to oracle instance.

Unit III

Managing Oracle: starting up the oracle instance, managing sessions, shutting down the oracle instance, instances messages -instance alerts. Control and Redo Log Files: Managing the control files, Maintaining and monitoring log files. Managing tables, indexes and constraints: Storing data (create, alter, analyzing, querying table information).

Unit IV

Introduction to Network Administration: Network design considerations, network responsibilities for the DBA, network configuration, Overview of oracle Net features, Oracle Net Stack Architecture.

Unit V

Backup and Recovery Overview: Database backup, restoration and recovery, Types of failure in oracle environment, defining a backup and recovery strategy, Testing the backup and recovery plan. Introduction to performance tuning: brief overview of Tuning methodology, General tuning concepts.

Textbooks / References:

1. Craig S. Mullins, ―Database Administration: The Complete Guide to DBA Practices and Procedures, Second Edition, Addison Wesley, 2012.
2. C.J. Date, ―Introduction to Database Systems, Eighth Edition, Addison Wesley, 2003.
3. Chip Dawes, Biju Thomas, ―Introduction to Oracle 9i SQL‖, BPB, 2002.
4. Bob Bryla, Biju Thomas, ―Oracle 9i DBA Fundamental I‖, BPB, 2002.
5. Kevin Loney,"Oracle Database 10g: The Complete Reference", McGraw-Hill

26CSA529	TIME SERIES ANALYSIS	L-T-P-C: 3-0-0-3

Course Description

This course introduces the basic time series analysis and forecasting methods that include stationary processes, ARMA models, spectral analysis, model and forecasting using ARMA models, nonstationary and seasonal time series models, multivariate time series, state-space models, and forecasting techniques.

Course Objectives

- This course aims to give students a clear understanding of basic concepts of time series analysis applicable to real-world analytical problems across various domains
- Students will learn several important tools and techniques for trend analytics and forecasting based on past data and time series.
- Students will be able to analyse and solve complex problems by applying time series models to derive effective solutions.

Course Outcomes

Cos	Description
CO1	Comprehend the important time series models and formulate real-world problems using time series models.
CO2	Explore various ARIMA models using statistical software to estimate parameters, interpret results, and develop data-driven solutions
CO3	Evaluate forecasting techniques and non-stationary time series using real-world case studies

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								

CO1	3	2	-	-	-	1	-	-
CO2	1	2	2	-	-	1	-	-
CO3	3	1	1	-	3	1	-	-

Syllabus

Unit I
Introduction - Examples of time series, Stationary models and autocorrelation function, Estimation and elimination of trend and seasonal components, Stationary Process and ARMA Models- Basic properties and linear processes, Introduction to ARMA models, properties of sample mean and autocorrelation function, Forecasting stationary time series, ARMA (p, q) processes, ACF and PACF, Forecasting of ARMA processes, Linear Filtering- Definitions and the Theorem of Filtering Convolutions and compositions, causal processes.

Unit II
Modeling and Forecasting with ARMA Processes - Preliminary estimation, Maximum likelihood estimation, Diagnostics, Forecasting, Order selection, Nonstationary and Seasonal Time Series Models- ARIMA models, Identification techniques, Unit roots in time series, Forecasting ARIMA models, Seasonal ARIMA models, Regression with ARMA errors, Multivariate Time Series- Second-order properties of multivariate time series, Estimation of the mean and covariance, Multivariate ARMA processes, best linear predictors of second-order random vectors, Modeling and forecasting.

Unit III
Forecasting Techniques - The ARAR algorithm, the Holt-Winter algorithm, the Holt-Winter seasonal algorithm Estimation of time series models, Non-stationary Processes- ARIMA and SARIMA processes, simulations and examples, Model selection and case studies, Exponential Smoothing Based Methods.

Textbooks /References:

1. Chatfield, C. (2022). The Analysis of Time Series, 7th edition, Chapman and Hall, New York
2. James D. Hamilton (1994). Time Series Analysis, 1st Edition, Princeton University Press
3. Douglas C. Montgomery , Cheryl L. Jennings , Murat Kulahci, Introduction to Time Series Analysis and Forecasting (Wiley Series in Probability and Statistics) 2nd Edition.

26CSA534	INFORMATION RETRIEVAL	L-T-P-C: 3-0-0-3

Course Description

This course provides a scientific foundation for information search and retrieval focusing on the design and development of text-based information systems. It covers fundamental and advanced information retrieval techniques for the web, including crawling, link-based algorithms, and metadata usage, document clustering and classification.

Course Objectives

- To understand and comprehend the challenges in IR-related issues
- To gain the skills required to build and deploy real-world applications leveraging information retrieval methods.
- To learn advanced multimodal information system approaches

Course Outcomes

Cos	Description
CO1	Familiarize with the basic methods for information extraction and retrieval of textual data.
CO2	Implement text processing techniques to prepare documents for statistical and machine learning models
CO3	Evaluate the performance of machine learning models for textual data.
CO4	Analyse the concept of machine learning models for analyzing textual data and correctly Interpreting the results.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								

CO1	3	2	2	1	-	-	-	-
CO2	2	2	2	2	-	-	-	-
CO3	3	3	1	3	-	-	2	3
CO4	1	1	2	1	-	-	3	3

Syllabus

Unit I

Boolean Expression Based Retrieval: Vocabulary and Postings –Lists –Dictionaries and Tolerant Retrieval – Index Construction and Compression -Scoring and Vector Space Model–Score Computation.

Unit II

Evaluating Information Retrieval Systems. Relevance Feedback and Query Expansion –XML Based Retrieval– Probabilistic Models –Language Models-Text Classification –Vector Space Classification –SVM Based Document. Data Fusion-Metasearch Data fusion, early and late fusion, Metasearch engines of retrieval.

Unit III

Latent Semantic Indexing –Web Search –Web Crawlers –Link Analysis –Unstructured Data Retrieval Semantic Web.

Unit IV

Ontology -Implementations using Natural Language Toolkit. Distributed Information Retrieval: A theoretical Model of Distributed retrieval, web search.

Textbooks / References:

1. C. Manning, P. Raghavan and H. Schütze, “Introduction to Information Retrieval”, Cambridge University Press, 2008.
2. R. Baeza-Yates and B. Ribeiro Neto, “Modern Information Retrieval: The Concepts and Technology Behind Search”, Second Edition, Addison Wesley, 2011.
3. David A. Grossman and Ophir Frieder “Information Retrieval: Algorithms and Heuristics”, Second Edition, Springer 2004.

26CSA535 INFORMATION SCIENCE AND ETHICS L-T-P-C: 3-0-0-3

Course Description

This course explores the ethical foundations of information science in the context of modern information and communication technologies. It examines the philosophical principles underlying information ethics, including ethical issues related to computing, data, and information management. The course focuses on the responsible use of information, challenges arising from misuse and abuse of information technologies, human moral agency, and social responsibility in the global information society.

Course Objectives

- To create awareness of engineering ethics and human values in professional practice.
- To understand moral, social values, and the importance of loyalty in personal and professional life.
- To appreciate the rights, responsibilities, and dignity of others in society.
- To develop awareness of safety, risk assessment, and ethical decision-making in engineering contexts.

Course Outcomes

Cos	Description
CO1	Identify and analyze current ethical issues facing the information professions.
CO2	Use major and alternative traditions of ethics to engage contemporary dilemmas.
CO3	Evaluate and articulate arguments regarding professional decision making.
CO4	Interpret the professional and scholarly literature of information ethics.
CO5	Recognize the ethical challenges of contemporary information trends and extrapolate the future direction of the information ethics field.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								

CO1	1	1	-	-	1	1	2	1
CO2	2	1	-	-	1	-	2	1
CO3	3	3	3	-	1	1	2	1
CO4	-	4	3	-	1	1	3	1
CO5	2	3	2	-	1	-	3	1

Syllabus

Unit I

Introduction to Information Ethics

Introduction: a unified model of information ethics, First stage: IE as an ethics of informational resources, Second stage: IE as an ethics of informational products, Third stage: IE as an ethics of the informational environment, Fourth stage: IE as a macro ethics. Information ethics as environmental ethics: The foundationalist problem, Classic macro ethics and ICTs ethical problems, An informational model of macro ethics, From computer ethics to information ethics, Information ethics as a patient-oriented and onto centric theory, The normative aspect of information ethics: four ethical principles, Information ethics as a macro ethics.

Unit II

Information ethics and the foundationalist debate: looking for the foundations of computer ethics, The 'no resolution approach': CE as not a real discipline. Distributed morality: Introduction: the basic idea of distributed morality, The old ethical scenario without distributed morality, The new ethical scenario with distributed morality, Some examples of distributed morality.

Unit III

Information business ethics: Introduction: from information ethics to business ethics, The informational analysis of business, The WHI ethical questions: what, how, and impact, The ethical business. Global information ethics: Introduction: from globalization to information ethics, Globalizing ethics, Global-communication ethics vs. global-information ethics, Global-information ethics and the problem of the lion, Global information-ethics and its advantages, the cost of a global-information ethics: postulating the ontic trust.

Unit IV

Ethics for IT workers and IT users, IT security incidents: a major concern, implementing trustworthy computing, privacy: privacy protection and the law, intellectual property: what is intellectual property, Copyrights, Patents, Trade Secrets, Key Intellectuals property issues. In defense of information ethics: Introduction: addressing the sceptic, IE is an ethics of news, IE is too reductivist, IE fails to indicate what information constitutes an individual, IE's de-anthropocentrization of the ethical discourse is mistaken, IE's measure of intrinsic moral value is insufficiently clear and specific.

Textbooks / References:

1. Handbook of research on Machine and Deep Learning Applications for Cyber Security, Padmavathi Ganapathi and D. Shanmugapriya, IGI Global.
2. The Ethics of Information, Luciano Floridi, Oxford
3. Ethics in information technology, George W. Reynolds, 5th Editions.

26CSA536	PATTERN RECOGNITION	L-T-P-C: 3-0-0-3

Course Description

This course focuses on the principles and methods used in designing automated systems that improve performance through experience. It explores statistical approaches to pattern recognition, covering fundamental techniques, tools, and algorithms. Students will study methods such as Bayesian decision theory, estimation theory, linear discriminant functions, nonparametric techniques, support vector machines, neural networks, decision trees, and clustering algorithms, with an emphasis on their application to real-world data analysis and intelligent systems.

Course Objectives

- To understand the fundamentals of pattern recognition.
- To explore the most cutting-edge algorithms and techniques used in pattern recognition research.
- To learn pattern recognition theories such as Bayes classifier and linear discriminant analysis and to use the techniques to solve real-world challenges.

Course Outcomes

Cos	Description
CO1	Implement the concepts of Tree classifiers and decision trees on patterns
CO2	Design both supervised and unsupervised classification methods to develop classifiers for real-world data.
CO3	Apply advanced techniques like Dimensionality Reduction for different features
CO4	Compare various techniques used by different clustering algorithms

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	2	2	-	-	-	-	-
CO2	3	2	2	2	-	-	-	-
CO3	2	2	3	2	1	-	-	-

CO4	2	2	3	2	2	-	2	2

Syllabus

Unit I

Introduction to Pattern Recognition-Tree Classifiers Getting our feet wet with real classifiers, Decision Trees: CART, C4.5, ID3-Random Forests-Bayesian Decision Theory Grounding our inquiry-Linear Discriminants Discriminative Classifiers: the Decision Boundary, Separability, Perceptron.

Unit II

Support Vector Machines, Parametric Techniques Generative Methods grounded in Bayesian Decision Theory, Maximum Likelihood Estimation-Bayesian Parameter Estimation. Non-Parametric Techniques-Kernel Density Estimators-Nearest Neighbor Methods, Unsupervised Methods.

Unit III

Exploring the Data for Latent Structure -Component Analysis and Dimension Reduction-The Curse of Dimensionality, Principal Component Analysis, Fisher Linear Discriminant, Locally Linear Embedding.

Unit IV

Clustering, K-Means. Expectation Maximization, Mean Shift, Classifier Ensembles, Bagging, Boosting / AdaBoost.

Textbooks / References:

1. Duda, Hart and Stork, Pattern Classification, Second Edition, Wiley, 2001.
2. T.M. Mitchell, Machine learning, McGraw-Hill, New York, 1997.
3. S. Theodoridis, K. Koutroumbas, Pattern recognition, Academic Press, 1999.

26CSA537	RECOMMENDATION SYSTEMS	L-T-P-C: 3-0-0-3

Course Objectives

This course focuses on the theory and practice of recommender systems for automating personalized decision-making processes. It covers the fundamental concepts, models, and algorithms used to design scalable, accurate, and cost-effective recommendation systems. The course emphasizes both theoretical foundations and practical applications, enabling students to build and evaluate high-quality recommender systems for real-world domains.

Course Objectives

To develop state-of-the-art recommendation systems that automate a variety of choice-making strategies with the goal of providing affordable, personal, and high-quality Recommendations and focusing on theory as well as on the practical use and applications of Recommender systems.

Course Outcomes

Cos	**Description**
CO1	Understand basic idea of Recommendation system and its applications in various fields.
CO2	Explore the various recommendation system, analyse the content based and collaborative filtering methods.
CO3	Understand Hybrid approaches and its applications in recommender systems
CO4	Apply different methods to build and evaluate recommender systems on historical datasets, conceive the various recommendation metrics.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	2	2	2	-	-	-	-
CO2	3	3	3	2	-	-	-	-
CO3	2	3	3	2	-	-	-	-
CO4	3	2	3	2	2	-	-	-

Prerequisites

- Data mining
- Machine learning

Syllabus

Unit I
Introduction and basic taxonomy of recommender systems (RSs). Traditional and non-personalized RSs. Overview of data mining methods for recommender systems, Applications of recommendation systems, Issues with recommender system.

Unit II
Content-based recommendation: High level architecture of content-based systems, Advantages and drawbacks of content-based filtering, Basic components of content-based RSs. Feature selection. Item representation Methods for learning user profiles.

Knowledge based recommendation: Knowledge representation and reasoning, Constraint based recommenders, Case based recommenders.

Collaborative Filtering: User-based nearest neighbour recommendation, Item-based nearest neighbour recommendation, Model based and pre-processing based approaches, Attacks on collaborative recommender systems

Unit III
Hybrid approaches: Opportunities for hybridization, Monolithic hybridization design: Feature combination, Feature augmentation, Parallelized hybridization design: Weighted, Switching, Mixed, Pipelined hybridization design: Cascade Meta-level, Limitations of hybridization strategies.

Unit IV
Evaluating Recommender System: Introduction, General properties of evaluation research, Evaluation designs, Evaluation on historical datasets, Error metrics, Decision-Support metrics, User-Centered metrics.

Textbooks / References:

1. C.C. Aggarwal, Recommender Systems: The Textbook, Springer, 2016.
2. F. Ricci, L Rokach, B. Shapira and P.B. Kantor, Recommender systems handbook, Springer 2010.
3. J. Leskovec, A. Rajaraman and J. Ullman, Mining of massive datasets, 2nd Ed., Cambridge, 2012. (Chapter 9).
4. Jannach D., Zanker M. and FelFering A., Recommender Systems: An Introduction, Cambridge University Press (2011), 1st ed.
5. Ricci F., Rokach L., Shapira D., Kantor B.P., Recommender Systems Handbook, Springer (2011), 1st ed.
6. Manouselis N., Drachsler H., Verbert K., Duval E., Recommender Systems for Learning, Springer (2013), 1st ed.

26CSA538	WEB MINING	L-T-P-C: 3-0-0-3

Course Description

This course introduces the concepts and techniques of web mining for discovering useful information and knowledge from web data, including hyperlink structures, web page content, and user usage logs. It explores the applications of web mining in areas such as e-commerce, web analytics, information retrieval and filtering, personalization, and recommender systems. The course also emphasizes the practical relevance of web mining techniques in understanding user behavior and extracting patterns from large-scale web data for developing intelligent web-based applications**.**

Course Objectives

- Introduce students to the basic concepts and techniques of Information Retrieval, Web Search, Data Mining, and Machine Learning for extracting knowledge from the web.
- Develop skills of using recent data mining software for solving practical problems of Web Mining.
- Gain experience of doing independent study and research

Course Outcomes

Cos	**Description**
CO1	Perceive Information Retrieval Models, explore web preprocessing and community discovery
CO2	Design and implement a crawler application to collect and index documents from the web and analyze structured data extraction.
CO3	Analyze text to determine the reliability of the information including potential bias. Explore web usage mining and opinion mining with case studies.

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	1	2	-	-	2
CO2	2	2	3	3	2	-	-	1
CO3	2	2	3	3	2	-	-	1

Prerequisites

Data Mining concepts

Syllabus

Unit I
Information Retrieval and Web Search: Information Retrieval Models, Text and Web Page Pre-Processing - Stopword Removal, Stemming, Web Page Pre-Processing, Inverted Index and Its Compression - Inverted Index, Search Using an Inverted Index. Latent Semantic Indexing, Web Search, Web Spamming. Link Analysis: Social Network Analysis, Co-Citation and Bibliographic Coupling, PageRank, HITS, Community Discovery- Bipartite Core Communities, Maximum Flow Communities, Email Communities Based on Betweenness.

Unit II
Web Crawling: A Basic Crawler Algorithm - Breadth-First Crawlers, Preferential Crawlers. Implementation Issues, Universal Crawlers, Focused Crawlers, Topical Crawlers, Evaluation, Crawler Ethics and Conflicts. Structured Data Extraction: Wrapper Induction, Automatic Wrapper Generation, String Matching and Tree Matching, Multiple Alignment, Extraction Based on a Single List Page and Multiple pages.

Unit III
Information Integration: Introduction to Schema Matching, Pre-Processing for Schema Matching, Schema-Level Match, Domain and Instance-Level Matching, Combining Similarities, Integration of Web Query Interfaces. Opinion Mining: Sentiment Classification, Feature-Based Opinion Mining and Summarization, Comparative Sentence and Relation Mining, Opinion Search, Opinion Spam. Web Usage Mining: Data Modeling for Web Usage Mining, Discovery and Analysis of Web Usage Patterns - Session and Visitor Analysis, Cluster Analysis and Visitor Segmentation, Analysis of Sequential and Navigational Patterns.

Textbooks / References:

1. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications) by Bing Liu, Springer Publisher.
2. Mining The Web: Discovering Knowledge from Hypertext Data by Chakrabarti Soumen, Elsevier Science
3. Web Mining: Applications and Techniques, Anthony Scime (State University of New York at Brockport, USA) Release Date: August, 2004|Copyright: © 2005 |Pages: 442 ISBN13: 9781591404149|ISBN10: 1591404142|EISBN13: 9781591404163, IGI Global Publisher.

26CSA539 BUSINESS ANALYTICS AND VISUALIZATION L-T-P-C: 3-0-0-3

Course Description:

This course focuses on applying statistical and analytical techniques to support data-driven business decision-making. It introduces the use of probability, statistical theories, and analytical models for solving real-world business problems. The course also emphasizes best practices in data visualization and explores time-series modeling for analyzing historical data and identifying trends to support strategic and operational decisions.

Course Objectives:

- To promote the ability to critically analyze and solve data-oriented real-world decision problems.
- To utilize the theories of statistics and probabilities in business analytics.
- To gain familiarity with an array of modeling techniques used to solve data-oriented decision problems.
- To learn best practices in visualization.
- To understand time series modeling of historical data.

Course Outcomes:

Cos	Description
CO1	Design visualizations like charts and advanced dashboard for various types of data
CO2	Evaluate statistical inferences of the classification and clustering techniques used using different case studies
CO3	Implement the smoothing techniques for real time problems for better forecasting

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	3	3	2	-	1	-	1
CO2	2	3	2	2	1	1	-	1
CO3	2	3	2	3	2	1	1	1

Syllabus

Unit I

Introduction to Business Analytics, Descriptive Statistics – Types of Data and its Measures, Data cleansing. Data Visualization-Design Techniques, Tables, Charts, Advanced data Visualization, Dashboards, Case Studies.

Unit II

Inferential Analysis – Statistical Inference, Descriptive Data mining - Clustering and Association Rules. Performance Evaluation, Overview of key Classification and prediction techniques, Case studies.

Unit III

Introduction to Forecasting, Time Series – Level, Trend, and Seasonality, Smoothing Techniques – Moving Average and Exponential Smoothing, Determining the best forecasting model to use. Case Study.

Textbooks / References:

1. Jeffrey D. Camm, James J. Cochran, Michael J. Fry, Jeffrey W. Ohlmann, David R. Anderson, Dennis J. Sweeney, Thomas A. Williams 'Business Analytics',3/e, Cengage Learning,2019.
2. GalitShmueli, Kenneth C. Lichtendahl Jr., 'Practical Time Series Forecasting with R: A Hands-On Guide', 2/e, Axelrod Schnall Publishers, 2016.
3. Joel Grus, 'Data Science from Scratch: First Principles with Python', 2/e, O'Reilly Media, 2019.
4. Cole Nussbaumer Knaflic, 'Storytelling with Data: A Data Visualization Guide for Business Professionals', John Wiley& Sons, 2015.
5. Claus O. Wilke, "Fundamentals of Data Visualization: A primer for making informative and compelling figures", O'Reilly, 2019.

26CSA540 COMPUTATIONAL INTELLIGENCE L-T-P-C: 3-0-0-3

Course Description

This course introduces the fundamental principles and methods of computational intelligence used for intelligent data processing and problem solving. It provides an overview of key intelligent system technologies, including knowledge-based systems, neural networks, fuzzy systems, and evolutionary computation, with an emphasis on their concepts, models, and applications in real-world computational environments.

Course Objectives

The Objective of the course is to make students familiar with basic principles of various computational methods of data processing that can commonly be called computational intelligence. This course introduces the fundamentals of key intelligent systems technologies including knowledge-based systems, neural networks, fuzzy systems, and evolutionary computation.

Course Outcomes

COs	Description
CO1	Understand the need for and importance of Computational intelligence.
CO2	Understand the concepts of neural networks and backpropagation learning.
CO3	Implement associative memory using neural networks.
CO4	Understand the idea of fuzzy logic in real-world problems.
CO5	Understand hybrid approaches to solve real-world problems.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	-	-	-	-	-
CO2	3	2	2	-	-	-	-	-
CO3	3	2	3	-	-	-	-	-
CO4	2	3	3	-	-	-	-	
CO5	2	3	3	-	-	-	-	-

Prerequisites

- Machine Learning
- Programming languages

- Probability

Syllabus

Unit I

Artificial Intelligence – a Brief Review – Pitfalls of Traditional AI – Need for Computational Intelligence – Importance of Tolerance of Imprecision and Uncertainty – Overview of Artificial Neural Networks - Fuzzy Logic – Evolutionary Computation.

Unit II

Neural Network: Biological and Artificial Neuron, Neural Networks, Supervised and Unsupervised Learning. Single Layer Perceptron - Multilayer Perceptron – Backpropagation Learning.

Unit III

Neural Networks as Associative Memories - Hopfield Networks, Bidirectional Associative Memory. Topologically Organized Neural Networks – Competitive Learning, Kohonen Maps.

Unit IV

Fuzzy Logic: Fuzzy Sets – Properties – Membership Functions - Fuzzy Operations. Fuzzy Logic and Fuzzy Inference - Applications. Evolutionary Computation – Constituent Algorithms.

Unit V

Swarm Intelligence Algorithms - Overview of other Bio-inspired Algorithms - Hybrid Approaches (Neural Networks, Fuzzy Logic, Genetic Algorithms etc.). Case Studies: Prediction Models, Optimization Models.

Textbooks / References:

1. Laurene Fausett, Fundamentals of Neural Networks, 2nd edition, Pearson, 1993
2. Ross T J, ―Fuzzy Logic with Engineering Applications‖, McGraw Hill, 1997.
3. Eiben A E and Smith J E, ―Introduction to Evolutionary Computing‖, Second Edition, Springer, Natural Computing Series, 2007.
4. Kumar S, ―Neural Networks - A Classroom Approach‖, Tata McGraw Hill, 2004.
5. Engelbrecht, A.P, ―Fundamentals of Computational Swarm Intelligence‖, John Wiley & Sons, 2006.
6. Konar. A, ―Computational Intelligence: Principles, Techniques and Applications‖, Springer Verlag, 2005.

26CSA527	GENERATIVE AI AND INTELLIGENT SYSTEM	L-T-P-C: 3-0-0-3

Course Description

This course provides a comprehensive introduction to the mathematical and computational foundations of Generative Artificial Intelligence, focusing on modern generative models and their applications. It covers techniques such as autoencoders, generative adversarial networks, Transformers, and structured probabilistic models, along with ethical considerations and real-world use cases.

Course Objectives

- To develop a strong understanding of the mathematical and computational foundations of generative modeling and its real-world applications.
- To introduce and analyze key generative AI techniques including variational autoencoders, generative adversarial networks, Transformers, normalizing flow models, and information lattice learning.
- To equip students with skills in neural text decoding, prompt programming, and detection of AI-generated content for responsible and effective use of generative AI systems.

Course Outcomes

COs	Description
CO1	Explain the fundamental concepts of Generative Artificial Intelligence and analyze the representational power of autoencoders and variational autoencoders for dimensionality reduction and feature learning.
CO2	Design and evaluate Generative Adversarial Networks and autoregressive models, and apply them to real-world problems such as anomaly detection and intrusion detection systems
CO3	Understand and implement Transformer-based architectures, analyze scaling laws, and apply large language models for sequence modeling and text generation tasks.
CO4	Analyze structured probabilistic models using graphical representations, perform inference and sampling from graphical models, and apply Restricted Boltzmann Machines to generative learning problems.
CO5	Evaluate advanced generative models, address ethical and societal challenges, and apply responsible Generative AI techniques to practical applications across multiple domains.

CO-PO Mapping

PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO7	PO8
CO								
CO1	3	2	1	1	-	-	-	-

CO2	2	3	3	-	-	-	-	-
CO3	2	2	3	1	-	-	-	-
CO4	3	3	1	-	-	-	-	-
CO5	1	2	2	2	-	-	-	-

Unit I – Introduction to Generative Models and Autoencoders
Introduction to Generative Artificial Intelligence and its evolution, generative vs discriminative models, applications of generative models. Autoencoders: representational power, effect of layer size and depth, undercomplete autoencoders, regularized autoencoders, denoising autoencoders, contractive autoencoders. Variational Autoencoders (VAEs), Case study: Applications of autoencoders in dimension reduction.

Unit II - Generative Adversarial Networks and Autoregressive Models
Generative Adversarial networks (GAN) – structure and training algorithm, Deep Convolutional GAN, Autoregressive models – Finite memory, long range memory through RNN and CNN, Transformers – Encoder, decoders, scaling laws, Case study: Generative Adversarial Networks-aided Intrusion Detection System.

Unit III - Transformers and Large Language Models
Limitations of sequential models, attention mechanisms, self-attention, multi-head attention, positional encoding. Transformer architecture: encoder, decoder, encoder–decoder models. Scaling laws in large models, pre-training and fine-tuning strategies. Overview of large language models (BERT, GPT, T5). Applications in text generation, summarization, and sequence modeling.
Case Study: Transformer-based text generation systems.

Unit IV – Structured Probabilistic Models and Graphical Models
Issues with unstructured generative models. Structured probabilistic models: directed and undirected graphical models. Bayesian Networks and Markov Random Fields. Partition function, marginalization and inference. Separation and D-separation, conversion between directed and undirected graphs.

Unit V – Advanced Generative Models, Ethics, and Applications
Energy-based models, diffusion models and denoising diffusion probabilistic models. Multimodal generative models combining text, image, and audio. Evaluation metrics for generative models. Ethical considerations: bias, fairness, deepfakes, data privacy, intellectual property issues. Social impact of generative AI and responsible deployment.
Case Study: Generative AI applications in healthcare, cybersecurity, and creative industries.

Textbook(s)

1. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning", MIT Press, 2016.

Reference(s)

1. Raut, R., Pathak, P. D., Sakhare, S. R., & Patil, S. (Eds.), "Generative Adversarial Networks and Deep Learning: Theory and Applications". CRC Press, 2023.
2. M. Tomcsak, "Deep Generative Modeling", Springer, 2022.
3. Langr J, Bok V. "GANs in action: deep learning with generative adversarial networks". Manning. 2019.
4. Papoulis and S. U. Pillai, "Probability – Random Variables, and Stochastic Processes", Fourth Edition, McGraw-Hill, 2017.

26CSA528 AGENTIC AI L-T-P-C: 3-0-0-3

Course Description

This course will enable students to understand the principles and architectures of Agentic Artificial Intelligence systems and their role in building autonomous and intelligent applications. It covers the foundations, design methodologies, implementation strategies, evaluation techniques, and operational aspects of single-agent and multi-agent AI systems.

Course Objective(s)

- To understand the fundamental concepts and paradigms of Agentic Artificial Intelligence and autonomous AI systems.
- To familiarize students with architectural patterns, design trade-offs, and orchestration strategies for agentic AI systems.
- To develop the ability to engineer, evaluate, deploy, and manage production-ready agentic AI solutions using contemporary frameworks and tools.

Course Outcomes

COs	**Description**
CO1	Explain the fundamental concepts, architectures, and paradigms of Agentic Artificial Intelligence systems.
CO2	Analyze real-world problems and identify suitable agentic AI design approaches to meet user and system requirements.
CO3	Design and implement agentic AI solutions using modern frameworks, tools, and orchestration strategies.
CO4	Evaluate, deploy, and manage agentic AI systems considering performance, safety, ethics, and operational constraints.
CO5	Work effectively in teams to design agentic AI systems, manage projects, and adapt to emerging technologies through continuous learning.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	2	1	2	–	–	-	2
CO2	2	3	2	2	–	–	-	2

CO3	2	2	3	3	2	-	-	2
CO4	1	2	2	3	1	-	-	2
CO5	1	1	2	2	3	-	-	3

Syllabus

Unit 1 – Foundations of Agentic AI – Overview of AI-enabled systems, architecting AI-enabled systems, essentials of Large Language Models (LLMs), prompt engineering techniques, Retrieval Augmented Generation (RAG), understanding the agentic paradigm with emphasis on autonomy, reasoning, planning, and goal-directed behavior.

Unit 2 – Designing Agentic AI Systems – Architectural patterns for Agentic AI systems including single-agent and multi-agent architectures, quality attributes and design trade-offs in agentic systems, orchestration and coordination strategies for multi-agent workflows, inter-agent communication mechanisms, collaboration and conflict resolution strategies.

Unit 3 – Agent Engineering and Implementation – Agentic reasoning–planning–action loop, function calling and structured tool invocation, tool integration and memory management in agents, Model Context Protocol (MCP), Agent-to-Agent (A2A) interaction, frameworks and platforms for implementing Agentic AI systems, practical implementation strategies using real-world application scenarios.

Unit 4 – Evaluation, Deployment and Operations – Testing, benchmarking, and evaluation of Agentic AI systems, safety, robustness, and failure handling in autonomous agents, deployment architectures and monitoring strategies, observability and lifecycle management, production-ready agents with focus on AgentOps and maintainability, industrial case study on the end-to-end design, deployment, and evaluation of an Agentic AI system, emerging trends, best practices, and future research opportunities.

References

1. Building Agentic AI Systems: Create intelligent, autonomous AI agents that can reason, plan, and adapt, Anjanava Biswas, Wrick Talukdar Packt Publishing, 2025
2. Agentic Design Patterns: A Hands-on Guide to Building Intelligent Systems, Antonio Gulli, Springer 2025
3. Agentic Artificial Intelligence: Harnessing AI Agents to Reinvent Business, Work, and Life, Bornet , Jochen Wirtz , Thomas H Davenport , 2025
4. Agentic AI Engineering: The Definitive Field Guide to Building Production-Grade Cognitive Systems, Yi Zhou, Argolong Publishing, 2025

CYBER SECURITY STREAM

26CSA550	ESSENTIALS OF CYBER SECURITY	L-T-P-C: 3-0-0-3

Course Description

This course introduces the fundamental concepts of cyber security, including digital security principles, access control mechanisms, browser and network security, and common security protocols. It also builds awareness of cyber-attacks, data privacy issues, and safe practices required to protect digital systems and information in modern computing environments.

Course Objectives

Gain an understanding of digital security, access control mechanisms, browser security keywords and jargon, network basics and security protocols, and awareness of cyber-attacks and data privacy.

Course Outcomes

COs	Description
CO1	Develop a solid foundation in digital security and implement measures to protect devices from threats.
CO2	Learn access control mechanisms and understand how to secure servers.
CO3	Comprehend the fundamentals of networking and gain a brief introduction to the security of network protocols.
CO4	Understand cyber-attacks, learn about data privacy issues, and explore preventive measures.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	-	-	-	1	-
CO2	3	-	-	-	-	-	-	-
CO3	3	-	-	-	-	-	-	-
CO4	3	-	-	-	-	-	1	-

Syllabus

Unit I

Basics of digital security, protecting personal computers and devices, Types of malwares, protecting devices from Virus and Malware, Identity, Authentication and Authorization, need for strong credentials, keeping credentials secure

Unit II

protecting servers using physical and logical security, World Wide Web (www), the Internet and the HTTP protocol, security of browser to web server interaction

Unit III

Networking basics, Networking concepts (CIDR, subnets), and protocols (DNS, DHCP, IP). Security of protocols, sample application hosted on-premises

Unit IV

Introduction to cyber-attacks, application security (design, development and testing), operations security, monitoring, identifying threats and remediating them, Principles of data security - Confidentiality, Integrity and Availability.

Textbooks / References:

1. Sammons, John, and Michael Cross. The basics of cyber safety: computer and mobile device safety made easy. Elsevier, 2016.
2. Charles P. Pfleeger, Shari Lawrence, Pfleeger Jonathan Margulies; Security in Computing, Pearson Education Inc. 5th Edition, 2015
3. Brooks, Charles J., Christopher Grow, Philip Craig, and Donald Short. Cybersecurity essentials. John Wiley & Sons, 2018

26CSA551	**MALWARE ANALYSIS**	**L-T-P-C: 3-0-0-3**

Course Description

This course introduces the fundamental concepts and techniques used in malware analysis. It focuses on identifying different types of malwares through static and behavioral analysis, understanding malware capabilities and persistence mechanisms, and assessing the potential threats posed by malicious activity on computer systems and networks.

Course Objectives

This course aims to identify malware types based on static and behavioral analysis, determine malware capabilities and persistence vectors and evaluate potential threat from malware activity on the network.

Course Outcomes

COs	**Description**
CO1	Possess the skills necessary to carry out independent analysis of modern malware samples using both static and dynamic analysis techniques.
CO2	Understand executable formats, Windows internals and API, and analysis techniques.
CO3	Extract investigative leads from host and network-based indicators associated with a malicious program.
CO4	Apply techniques and concepts to unpack, extract, decrypt, or bypass new anti-analysis techniques in future malware samples.
CO5	Achieve proficiency with industry standard tools including IDA Pro, OllyDbg, WinDBG, PE Explorer, ProcMon etc.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	3	-	-	-	2
CO2	3	1	-	-	-	-	-	-
CO3	3	1	-	1	-	-	-	-
CO4	3	2	-	1	-	-	-	-
CO5	3	1	-	3	1	-	-	-

Syllabus

Unit I

Introduction to malware, Basic Static and Dynamic Analysis,

Unit II

Overview of Windows file format, PEView.exe, Patching Binaries, Disassembly (objdump, IDA Pro),

Unit III
Introduction to IDA, Introduction to Reverse Engineering, Extended Reverse Engineering using GDB and IDA, Advanced Dynamic Analysis - debugging tools and concepts, Malware Behavior - malicious activities and techniques, Analyzing Windows programs – Win API, Handles, Networking, COM, Data Encoding, Malware Countermeasures, Covert Launching and Execution,

Unit IV
Anti Analysis - Anti Disassembly, VM, Debugging -, Packers – packing and unpacking, Intro to Kernel – Kernel basics, Windows Kernel API, Windows Drivers, Kernel Debugging,

Unit V
Rootkit Techniques- Hooking, Patching, Kernel Object Manipulation, Rootkit Anti-forensics, Covert analysis.

Textbooks / References:

1. Michael Sikorski and Andrew Honig, “Practical Malware Analysis”, No Starch Press,2012
2. Jamie Butler and Greg Hoglund, “Rootkits: Subverting the Windows Kernel”, Addison-Wesley, 2005
3. Dang, Gazet and Bachaalany, “Practical Reverse Engineering”, Wiley,2014
4. Reverend Bill Blunden, “The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System” Second Edition, Jones& Bartlett, 2012.

26CSA552 BLOCKCHAIN AND DECENTRALIZED APPLICATIONS L-T-P-C: 3-0-0-3

Course Description

This course introduces the concepts of blockchain technology and decentralized applications. It explores the fundamental algorithms and architectures that enable secure, transparent, and decentralized systems, and examines the role of blockchain in addressing security challenges in the modern technological landscape.

Course Objectives

Students will be exposed to blockchains and decentralized applications. They will understand the fundamental algorithms supporting this modern technology and its place in the security setting of the modern technology era.

Course Outcomes

COs	Description
CO1	Understand the fundamental characteristics of Blockchain and cryptocurrency
CO2	Understand the basics concepts of Bitcoin and Ethereum Blockchain
CO3	Develop smart contracts using Solidity and Remix IDE
CO4	Understand the architecture of distributed applications
CO5	Develop DApps for real-life use cases

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	-	-	-	-	-
CO2	3	-	-	-	-	-	-	-
CO3	2	-	1	3	-	-	-	-
CO4	3	-	-	-	-	-	-	-
CO5	2	-	2	3	-	-	-	-

Syllabus

Unit I

Need for Distributed Record-Keeping, distributed ledger technology, Modeling faults and adversaries, Byzantine Generals problem, Nakamoto’s concept with Blockchain-based cryptocurrency, Transaction: - syntax, structure and validation, Blocks- Structure, Genesis block, and Merkle tree. Mining: -target, hash rates, Consensus mechanisms, forking. Byzantine fault-tolerant distributed computing,

Unit II

coins, wallets, Bitcoin scripting language, Ethereum smart contract architecture, contract transactions, comparing Bitcoin scripting vs. Ethereum Smart Contracts,

Unit III

Remix IDE, Solidity: - variables, data types, addresses and balances, strings in Solidity, global Msg-Object, mapping, structure, array, require, assert revert, constructor, fallback functions, View/Pure Getter functions. modifier, inheritance, importing of Files, events and return variables, ABI array, debugging libraries

Unit IV

DApps architecture, blockchain server, Truffle suite: setup and test cases, Web3 SDK, Web3 provider, Ganache, MetaMask integration with web3, channel concept and micropayment channel, web interface for DApps, Deployment to public testnet and mainnet, Network ID, Infura API, private Blockchain, Go-Ethereum, Type of DApps, Oracles, Ethereum improvement proposal(EIP) framework, standard ERC 20 for token Dapps, ERC 721 for non-fungible tokens, RES4

Textbooks / References:

1. Ramamurthy, Bina. *Blockchain in action*. Manning Publications, 2020.
2. https://web3js.readthedocs.io/en/v1.7.3/
3. Merunas Grincalaitis, “Mastering Ethereum: Implement Advanced Blockchain Applications Using Ethereum-supported Tools, Services, and Protocols”, Packt Publishing.

26CSA553	FUNDAMENTALS OF CYBERSECURITY OPERATIONS	L-T-P-C: 3-0-0-3

Course Description

This course introduces students to the foundational concepts of cybersecurity operations with a focus on real-world security monitoring, incident detection, and response. It covers common network and application activities, attack vectors, and security vulnerabilities across IT infrastructures. Students gain exposure to security alerts, breach analysis, and incident investigation techniques used in modern Cybersecurity Operations Centers (SOCs). The course also emphasizes understanding logs, threat indicators, and operational workflows required to support effective security operations and contribute as an entry-level SOC analyst.

Course Objectives

The course teaches students security concepts, common network and application operations and attacks, and the types of data needed to investigate security incidents. Students will learn how to monitor alerts and breaches and become a contributing member of a Cybersecurity Operations Center (SOC) including understanding the IT infrastructure, operations, and vulnerabilities.

Course Outcomes

COs	**Description**
CO1	Understand the functionalities of various SOC generations.
CO2	Understand different data collection, data analysis, and security analysis techniques as part of SOC technologies.
CO3	Understand vulnerability management techniques and threat intelligence methodologies.
CO4	Assess the SOC capabilities using different SOC tools and techniques.
CO5	Learn how SOC helps in business continuity and disaster recovery plan.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	-	-	-	-	-
CO2	3	1	-	-	-	-	-	-
CO3	3	1	1	-	-	-	-	-
CO4	3	-	1	-	-	-	-	-
CO5	2	-	-	-	-	-	-	-

Syllabus

Unit I

Information security incident management (Incident detection, triage and incident categories, Incident severity, resolution, Closure, Post-incident), Security Operations Center (SOC) Generations (First-generation, second, third and fourth generation SOC), SOC Maturity models (Introduction to maturity models, and applying maturity models in SOC),

Unit II

SOC Technologies-1 (Data collection and analysis, syslog protocol), SOC Technologies-2 (Telemetry Data, Security analysis, Data enrichment),

Unit III

Vulnerability Management (Broad introduction), Threat intelligence (Broad introduction)

Unit IV

Assessment of SOC capabilities (Business and IT Goals, Assessing capabilities & IT processes), SOC - Business Continuity, Disaster recovery (Importance of BCP and DR processes, and its interface to SOC), Security event generation and collection (Cloud Security, IDPS, Breach Detection), SOC and SIEM – Introduction (Role of SIEM in SOC), SOC and Splunk (Splunk architecture & SOC, Splunk Rules, Splunk log management, Splunk correlation), SOC and Health Care - A Case study (SOC Considerations for a HealthCare situation), SOC and Application security (OWASP, Application security and SOC).

Textbooks / References:

1. Security Operations Center: Building, Operating, and Maintaining Your SOC, Book by Gary McIntyre, Joseph Muniz, and Nadhem AlFardan
2. Designing and Building Security Operations Center, 2015, Book by David Nathans
3. Security Operations Center - SIEM Use Cases and Cyber Threat Intelligence, 2018, Book by Arun E Thomas.
4. The Modern Security Operations Center, 2021, Book by Joseph Muniz
5. Principles for Cyber Security Operations, 2020, Book by Hinne Hettema.

26CSA554	CLOUD AND INFRASTRUCTURE SECURITY	L-T-P-C: 3-0-0-3

Course Description

This course provides a comprehensive understanding of cloud and infrastructure security concepts with hands-on exposure to modern cloud platforms. It covers virtualization technologies, virtual machine provisioning, container and Kubernetes management, storage and database security, and secure network configuration in cloud environments. Students learn access control mechanisms, identity and key management, virtual private cloud (VPC) architecture, and secure networking practices. The course also introduces the design and deployment of secure microservice-based applications, load balancing strategies, homomorphic encryption concepts, and cloud security best practices, enabling learners to design, deploy, and manage secure cloud infrastructures.

Course Objectives

Familiarization of popular cloud platforms, VM creation, Container management, and Kubernetes, Storage management, Database creation, Network management, Access control mechanism in a computing environment, Virtual private cloud, Design and deployment of secure microservice applications, load balancing, Identity management, Homomorphic encryption, VPC Networking, and security.

Course Outcomes

COs	Description
CO1	Understand the architecture and infrastructure of cloud computing along with hands-on experience in various cloud computing platforms.
CO2	Identify the known threats, risks, vulnerabilities, and privacy issues in the various layers of cloud computing.
CO3	Compare modern security concepts as they are applied to cloud computing.
CO4	Understand the concepts and various methods of secure data management in the cloud.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	-	-	-	-	-
CO2	3	2	-	-	-	-	-	-
CO3	3	1	1	-	-	-	-	-
CO4	3	-	2	1	-	-	-	-

Syllabus

Unit I

Cloud computing essentials: - Characteristics, service models, deployment models, NIST reference architecture, virtualization, containers, Kubernetes, design of microservices, high availability, Load Balancing in the cloud, cloud storage, and databases, cloud networking and vpc, popular cloud platforms, open-source architectures.

Unit II

Threats classification and countermeasures: - Infrastructure and host threats, service provider threats, generic threats, threats assessment, CSA Top threats, Virtualization system vulnerabilities, Authentication and authorization techniques for cloud solutions, Protection of application infrastructure, Protecting Data in the Cloud:- Tokenization, Cryptographic key management for data protection, Encryption techniques and applications for cloud computing, homomorphic encryption, Intrusion Detection and Prevention for cloud workloads

Unit III

Security breaches management for cloud computing, Cloud-centric regulatory compliance issues, and mechanisms.

Textbooks / References:

1. John R. Vacca (Editor), "Cloud Computing Security - Foundations and Challenges" CRC Press, 2017
2. Ronald L. Krutz and Russell Dean Vines, "Cloud Security- A Comprehensive Guide to Secure Cloud Computing", Wiley, 2010
3. Chris Dotson "Practical Cloud Security ", O'Reilly,2019
4. Tim Mather, S. Kumaraswamy, and S. Latif, "Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", O'Reilly Media, 2009.

26CSA555 CYBERSECURITY GOVERNANCE, RISK AND COMPLIANCE L-T-P-C: 3-0-0-3

Course Description

This course introduces students to the principles and practices of cybersecurity governance, risk management, and compliance within organizations. It focuses on identifying, assessing, and managing cyber risks using established frameworks and standards. The course covers vulnerability management, threat detection methods, security metrics, and evaluation techniques used to assess an organization's security posture. Students gain an understanding of the NIST Cybersecurity Framework, regulatory and compliance requirements, and the roles and responsibilities involved in managing cybersecurity at organizational levels. The course equips learners with the knowledge required to support effective security governance and risk-based decision-making in enterprise environments.

Course Objectives

The students will learn the principles of cybersecurity governance, risk, and compliance. They will understand the tools methods, including vulnerability management, threat detection, metrics, and evaluations of organizations. Students will study the NIST framework and learn organizational roles within a company.

Course Outcomes

COs	Description
CO1	Understand the different methods to assess cybersecurity maturity.
CO2	Understand the vulnerability management techniques and threat management methodologies.
CO3	Understand the governance metrics (Application security, vulnerability, and network security).
CO4	Know the relation between security analytics and security governance.
CO5	Understand the NIST compliance for security mandate.

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	-	-	-	-	-	-	-
CO2	3	1	-	-	-	-	-	-
CO3	3	-	-	-	-	-	-	-
CO4	3	1	-	1	-	-	1	-
CO5	2	-	-	-	-	-	-	-

Syllabus

Unit I

Basics of Cyber security governance, Principles of cyber-security governance, Assessment of cyber security maturity, Theories of governance – introduction, Governance – definitions and typologies.

Unit II

Governance of security operations, Tools, methods, and processes, Vulnerability management, Threat management, Endpoint management, Intrusion detection and prevention (IDPS), Security incident management.

Unit III

Security metrics and governance, Measurement of governance: Metrics – concepts, Application security metrics, Network security metrics, Security incident metrics, Vulnerability metrics, Service level objectives/agreement (SLO / SLA), NIST metrics.

Unit IV

Security analytics and governance, Basics of security analytics, Threat intelligence and governance, Data-driven security governance, Impact of cognitive security on security governance.

Unit V

Compliance and governance, Industry-specific security compliance, Cyber security governance – Republic of India, NIST mandates for compliance, Security reporting basics, CISO – role and organization structure, HIPAA, COBITZ compliance.

Unit VI

Cyber Security Risk: Information security risk management framework and methodologies, Risk Management Process, Framework, and Life Cycle, Identifying and modeling information security risks, Qualitative and quantitative risk assessment methods, Articulating information security risks as business consequences

Textbooks / References:

1. Information Security Governance: A Practical Development and Implementation Approach, Wiley publications 2009.
2. Information Security Governance, S.H. Solms, Rossouw Solms, Springer Science & Business Media.
3. Internet governance in an age of cyber insecurity,2010, Council on Foreign Relations Press.
4. Cyber justice: human rights and good governance for the internet, 2017, Springer.
5. Cyber Risk Management: Prioritize Threats, Identify Vulnerabilities and Apply Controls 1st Edition, Kogan Publishers, 2019.

26CSA556	CYBER SECURITY LAW	L-T-P-C: 3-0-0-3

Course Description

This course provides an overview of cyber security laws and legal frameworks governing cyberspace in India and internationally. It introduces students to legal principles related to data protection, privacy, cybercrimes, and handling of security breaches. The course covers national and international regulations, laws for combating cybercrime, and legal responsibilities of individuals and organizations. Students gain awareness of compliance requirements, legal procedures, and the role of law enforcement and regulatory bodies in ensuring cyber security, enabling them to understand and apply legal and ethical considerations in digital environments.

Course Objectives

Students will learn the cybersecurity laws in India and abroad. A clear introduction to the laws behind data security, breaches, cybercriminal combat, and much more.

Course Outcomes

COs	Description
CO1	Understand the history of cybercrime and the laws created
CO2	Understand the different classes of cyber-crime.
CO3	Gain knowledge of the IT act.
CO4	Know the procedures and authorities in India and abroad.
CO5	Gain familiarity with all laws regarding privacy.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	1	-	-	-	-	-	2	-
CO2	1	-	-	-	-	-	2	-
CO3	1	-	-	-	-	-	2	-
CO4	1	-	-	-	-	-	2	-
CO5	1	-	-	-	-	-	2	-

Syllabus

Unit I
Jurisprudence of cyber law, Information Technology Act, 2008, Cybercrimes, history and evolution of cybercrime, unauthorized access crimes, BEC, ATM frauds, online banking frauds, SIM swap frauds, email frauds, lottery frauds, Web defacement, Web Jacking, crimes relating to digital signature

Unit II
Penalties under the IT Act, Relevant Offences under the IT Act
Exemption of liability of intermediaries, Information Technology (Intermediary Guidelines and Digital Media

Ethics Code) Rules, 2021, due diligence, Procedures & Authorities

Unit III

Authorities and their duties; The National Cyber Coordination Centre (NCCC), Cyber and Information Security (C&IS) Division, National Critical Information Infrastructure Protection Centre (NCIIPC), National Technical Research Organization (NTRO)
Law of Privacy.
GDPR and the EU.

Textbooks / References:

1. Satish Chandra, “Cyberlaw in India”.
2. Nilakshi Jain, Ramesh Menon “Cybersecurity and Cyber laws” Wiley media.

26CSA557 **L-T-P-C:3-0-0-3**

MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE IN CYBERSECURITY

Course Description

This course introduces the application of machine learning and artificial intelligence techniques in the field of cybersecurity. It covers fundamental machine learning concepts such as classification, regression, supervised and unsupervised learning, along with model selection, validation, and evaluation methods. The course explores how these techniques are applied to security problems including intrusion detection, malware analysis, anomaly detection, and threat intelligence. It also examines the limitations, risks, and adversarial challenges associated with using machine learning in cybersecurity, enabling students to understand both the capabilities and constraints of intelligent security systems.

Course Objectives

The students will be exposed to the fundamentals of machine learning: classification, regression, supervised and unsupervised learning. They will learn which algorithms to use in which context, including model validation and evaluation. They will be exposed to a thorough survey of the fundamental security applications that machine learning provides the current security limitations of machine learning as well.

Course Outcomes

COs	**Description**
CO1	Understand relevance of machine learning and AI in cyber security
CO2	Gain proficiency in scikit-learn, using supervised and unsupervised learning.
CO3	Learn the fundamentals of regression and classification.
CO4	Make use of classification and anomaly detection systems in security – fraud and spam detection.
CO5	Learn to threat model for machine learning, understanding adversarial attacks.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	3	2					
CO2	1	3	1					
CO3	3	1						
CO4	2	3	3	1				
CO5	2	2	3					

Syllabus

Unit I

Python, Jupyter Notebooks, Pandas, Numpy, Matplotlib, Seaborn

Unit II

Scikit-Learn, Supervised learning: Linear regression, Decision Trees, Support Vector Machines, K-nearest neighbors, random forests, AdaBoost, gradient boosting, multi-layer perceptrons, logistic regression.
Unsupervised learning: k-means clustering, DBSCAN, GMM, PCA, T-SNE.

Bias-variance tradeoff. Learning and validation curves. Cross-validation, shuffle split, k-fold, time-series split. Random seeds. Baseline and benchmarking models. Gradient descent, regularization, feature scaling, one-hot encoding, label encoding. Train-test-split.
Metrics: accuracy, f1-score, precision, recall, confusion matrices. Gini impurity, information gain ratio, feature ranking with multivariate and univariate methods. Hyper-parameter tuning with grid search and random search. Natural language processing, ngrams, bag of words, vectorizers. Data wrangling with feature preprocessing and EDA.

Unit III

Artificial Intelligence (AI), Deep Learning (DL), and Machine learning (ML) in security. Understand the role AI plays in making decisions in large-scale settings. Algorithm bias and fraud.
Machine learning for security - anomaly detection, fraud detection, malware detection, spam detection, phishing detection, IDS, and NIDS.
Security of machine learning: adversarial attacks on machine learning. Data poisoning, model stealing, evasion attacks at inference time. Adversarial hardening.

Textbooks / References:

1. Tom M Mitchell, Machine Learning, McGraw Hill, 1997
2. Jake Vanderplas, Python Data Science Handbook, O’Reilly Media, 2016
3. Clarence Chio, David Freeman, Machine Learning and Security, O’Reilly Media, 2018

26CSA558	MOBILE SECURITY AND DEFENCE	L-T-P-C: 3-0-0-3

Course Description

This course introduces the fundamentals of mobile security and defense with a primary focus on the Android platform. It covers the Android architecture, application framework, and APIs to understand how mobile applications are designed and deployed. The course examines common mobile security threats, vulnerabilities, and exploits, along with hardening and defense techniques to secure Android applications and devices. Students gain hands-on exposure to static and dynamic analysis methods for detecting malicious applications, enabling them to identify security risks and implement effective mobile security measures.

Course Objectives

The students will learn the fundamental of mobile security and defense. Focus on android platform using android framework and APIs. Students will learn the structure of android applications, the exploits involved and common hardening techniques. Students will learn to perform static and dynamic analysis to identify malicious apps.

Course Outcomes

COs	Description
CO1	Understand internals of Android Operating System, security model of Android and iOS.
CO2	Understand how to make use of relevant tools to inspect and understand the working of Android and iOS application.
CO3	Learn how to identify vulnerable codebase and insecure configuration of application components.
CO4	Learn how to reverse engineer and perform advanced static and dynamic analysis.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	-	-	-	-	-
CO2	2	-	-	3	-	-	-	-
CO3	2	1	-	2	-	-	-	-
CO4	3	1	-	2	-	-	-	-

Syllabus

Unit I

History of Smartphones, Smartphone Applications and Development Ecosystem, Android Architecture, Syscalls, IPC mechanism in Android, Android Framework and APIs - APK, App Signing, Java/Dalvik Byte code, Android Run-Time, Reflection, Dynamic Code Loading, Serialization,

Unit II

Android Apps Overview - Java, Kotlin, Flutter and Android Studio, Activities and Intents - Life cycle, State and Architecture (Eg: MVVM), Broadcast Receiver, Content Provider, Services, Room Database and Shared Preference, Android emulator, AVD, ADB, SSL Pinning,

Unit III

Static Analysis - Assets and resources, Android Manifest, Native Code, Reverse engineering – apktool, jadx, Android App Bundles (AAB), Android System Security – Google Services, Android OS and Kernel, Device hardware, Android Malware – Stalkerware, Spyware, Adware. Vulnerabilities and Attack surfaces, Dynamic Analysis – Frida, Proxying Android traffic, Intercepting traffic using burp.

Textbooks / References:

1. Joshua J. Drake, Pau Oliva Fora, Zach Lanier, Collin Mulliner, Stephen A. Ridley,Georg Wicherski - “Android™ Hacker’s Handbook” 2014
2. Keith Makan - “Android Security Cookbook”, ISBN - 978-1782167167, December 2013
3. Dominic Chell, Tyrone Erasmus, Shaun Colley, Ollie Whitehouse - “The Mobile Application Hacker's Handbook”, ISBN: 978-1-118-95850-6, February 2015
4. Nikolay Elenkov - “Android Security Internals: An In-Depth Guide to Android's Security Architecture”, ISBN - 978-1593275815, 2014
5. Jonathan Levin, “Android Internals - A Confectioner's Cookbook - Power User's View - 1st edition”, ISBN - 978-0991055524, January 2015
6. Mobile Systems and Smartphone Security course (MOBISEC), Fall 2020 at EUROCOM

26CSA559	CYBER FORENSICS	L-T-P-C: 3-0-0-3

Course Description

After completion of the course the students will be able to learn investigation tools and techniques, analysis of data to identify evidence, Technical Aspects & Legal Aspects related to cybercrime.

Course Objectives

Students will learn an introduction to cyber forensics with investigation tools. They will be able to perform evidence collection, preservation, and data recovery. All platforms: Windows, Linux, iOS, Android will be covered.

Course Outcomes

COs	Description
CO1	Understand digital evidence collection and preservation techniques.
CO2	Familiarize with hardware forensics including disk, SSD, memory, and mobile device analysis.
CO3	Explore host/OS forensics for MS Windows, Linux, Android, iOS, and related file system forensics.
CO4	Understand forensic analysis of databases, emails, browsers, the dark web, and anti-forensic techniques
CO5	Explore network, wireless, cloud, and IoT forensics.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	-	-	-	-	-
CO2	3	1	-	1	-	-	-	-
CO3	3	1	-	1	-	-	-	-
CO4	3	1	-	1	-	-	-	-
CO5	3	1	-	1	-	-	-	-

Syllabus

Unit I

Introduction to Cyber Forensic Investigation, Investigation Tools, Digital Evidence Collection, Evidence Preservation

Unit II

Data Recovery, Encryption and Decryption methods, Search and Seizure of Computers and devices, Recovering deleted evidence, Password Cracking, Security Standards, Cyber Laws and Legal Frameworks, Cyber laws in India, Case studies and tools.

Hardware/SSD/Device Forensics

Unit III

File System Forensics, OS Forensics (Windows, Linux, Android and iOS), Memory Forensics, Web/Browser Forensics, Dark Web/Tor Forensics, E-Mail Forensics,

Unit IV

Mobile/Wireless Forensics, Network and Communication Forensics, Anti-forensics, Steganography, and Image File Forensics, IOT Forensics, Cloud Forensics, Overwriting/Forging/Wiping/Destruction, Obfuscation, Online Anonymity and Rootkits.

Assessing Threat Levels, Operating System Attacks, Malware Analysis, Financial Frauds, Espionage and

Investigations, Investigating copiers, IVR, Video surveillance, RFID and Sim cards.

Textbooks / References:

1. File System Forensic Analysis by Brain Carrier ISBN: 978-0-32-126817-2
2. Incident Response and Computer Forensics, Third Edition by Jason T Luttgens, Mathew Pepe ISBN: 978-0-07-179869-3
3. Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software by Michael Sikorski, Andrew Honig ISBN: 978-1-59327-290-6
4. Android Forensics: Investigation, Analysis and Mobile Security for Google Android by Andrew Hoog, ISBN: 978-1-59749-651-3
5. iPhone and iOS Forensics: Investigation, Analysis and Mobile Security for Apple iPhone, iPad, and iOS Devices by Andrew Hoog, Katie Strzempka ISBN: 978-1-59749-659-9.

REGULAR STREAM

26CSA560	COMPILER DESIGN	L-T-P-C: 3-0-0-3

Course Description

A compiler is system software that is required to convert computer programmes into a format that can be executed on the intended machine. Creating a compiler necessitates knowledge of several areas of computer science, including logic, formalism, mathematics, data structures, algorithms, and programming. This course is intended to serve as an introduction to the various stages involved in the design of standard compilers, beginning with the front-end stages of compilation and progressing to the back end and some recent advancements in the field.

Course Objectives

The goal of this course is to educate students on the phases of a compiler and the techniques for designing a compiler. This course introduces students to the fundamental concepts of compilation phases such as lexical analysis, syntax analysis, semantic analysis, intermediate code generation, code optimization, and code generation.

Course Outcomes

COs	Description
CO1	Describe stages of compilation, and lexical Analysis
CO2	Compare different types of parsers (Bottom-up and Top-down) and construct a parser for a given grammar.
CO3	Analyze syntax directed translation and representations of intermediate code
CO4	Describe type checking and run time environment
CO5	Illustrate code optimization and code generation techniques in the compilation.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	2	-	-	-	-	-	2
CO2	2	2	-	-	-	-	-	2
CO3	3	3	2	-	-	-	-	2
CO4	3	3	2	-	-	-	-	2
CO5	3	3	-	-	-	-	-	2

Syllabus

Unit I

Introduction To Compilers: Definition of compiler, interpreter and its differences, the phases of a compiler, role of lexical analyzer, regular expressions, finite automata, from regular expressions to finite automata, pass and phases of translation, bootstrapping, LEX-lexical analyzer generator.

Unit II

Parsing: Parsing, the role of the parser, context free grammar, derivations, parse trees, ambiguity, elimination of left recursion, left factoring, eliminating ambiguity from dangling-else grammar, classes of ptop-downtop down

parsing - backtracking, recursive descent parsing, predictive parsers, LL (1) grammars.
Bottom-Up Parsing: Definition of bottom-up parsing, handles, handle pruning, stack implementation of shift-reduce parsing, conflicts during shift-reduce parsing, LR grammars, LR parsers-simple LR, canonical LR(CLR) and Look Ahead LR (LALR) parsers, error recovery in parsing, parsing ambiguous grammars, YACC-automatic parser generator.

Unit III

Syntax Directed Translation: Syntax directed definition, construction of syntax trees, S-attributed and L-attributed definitions, translation schemes.
Intermediate Code Generation: intermediate forms of source programs– abstract syntax tree, polish notation and three address code, types of three address statements and their implementation syntax-directed translation into three-address code, translation of simple statements, Boolean expressions and flow-of-control statements.

Unit IV

Type Checking: Definition of type checking, type expressions, type systems, static and dynamic checking of types, specification of a simple type of checker, equivalence of type expressions, type conversions, overloading of functions and operators.
Run Time Environments: Source language issues, Storage organization, storage-allocation strategies, access to non-local names, parameter passing, symbol tables and language facilities for dynamic storage allocation.

Unit V

Code Optimization: Organization of code optimizer, basic blocks and flow graphs, optimization of basic blocks, the principal sources of optimization, the directed acyclic graph (DAG) representation of basic block, and global data flow analysis.
Code Generation: Machine dependent code generation, object code forms, the target machine, a simple code generator, register allocation and assignment, peephole optimization.

Textbooks / References:

1. Keith Cooper and Linda Torczon, "Engineering a Compiler", Second Edition, Morgan Kauffmann, 2011.
2. Alfred V.Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, "Compilers: Principles, Techniques and Tools", Prentice Hall, Second Edition, 2006.
3. Andrew W. Appel and Jens Palsberg, "Modern Compiler Implementation in Java", Cambridge University Press, Second Edition, 2002.
4. Tremblay and Sorenson, The Theory and Practice of Compiler Writing, Tata McGraw Hill & Company.

26CSA561 ADVANCED OPERATING SYSTEMS L-T-P-C: 3-0-0-3

Course Description

This is a graduate level course which provides a platform for students to dig deeper into modern operating system technology, implementation techniques and research issues. The course enables the students to specialize in Operating Systems by exposing the recent developments and research in the area. This course covers a broad range of topics which includes Unix architecture, design of modern operating systems, resource sharing and scheduling, software and hardware interaction, memory management, distributed and real time system behaviors etc.

Course Objectives

- Provide insights on the design principles of modern operating systems
- Understanding low level OS code and its interaction with hardware
- To gain knowledge on Distributed Operating System concepts
- To gain insights on the distributed resource management
- Create interest in students to explore more on the research aspects in the area

Course Outcomes

After completing this course, students will be able to:

COs	Description
CO1	Describe the architecture and process management system calls
CO2	Discuss memory management and I/O management services of OS
CO3	Illustrate the file and process subsystem of Linux Operating System
CO4	Apply the concepts of file management to implement different file access methods and directory structures in an operating system.
CO5	Interpret the challenges involved in designing distributed and real-time operating systems, emphasizing their practical applications and limitations.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	1	-	-	-	-	-	-
CO2	2	2		1	-	-	-	-
CO3	2	1		-	-	-	-	2
CO4	2	1	1	1	-	-	-	-
CO5	2	2		-	-	-	-	

Prerequisites

- Computer Architecture
- C Programming
- Data Structures

Syllabus

Unit I

Computer hardware review – Instruction execution cycle, Interrupts; Operating system concepts: Process abstraction, System calls for process management, Process execution mechanisms, Scheduling policies, Inter-process communication, Classic synchronization problems and their solutions, Deadlocks.

Unit II

Memory Management: physical memory organization, Address space abstraction, Address binding, Memory allocation strategies, fragmentation, swapping, Paging, Segmentation, Virtual memory, demand paging and its implementation, Page replacement algorithms

Unit III

Unix Internals: Architecture of Unix OS- Kernel Data structures, File subsystem and process subsystem – Process states and transitions – sleep and wakeup – buffer cache. File system – Internal representation of files – system calls for the file system.

Unit IV

File Management File concept, Access methods, Access Matrix, Implementation of Access Matrix, Access Control. File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management.

Unit V

Distributed Operating Systems – Architecture of Distributed systems, Communication mechanisms, Real Time Operating Systems: Introduction to Real Time Operating Systems, Concepts of scheduling, Real Time Memory Management.

Textbooks / References:

1. Silberschatz, Galvin, Gagne, Operating System Concepts, Tenth Edition, John Wiley & Sons, Inc.
2. Distributed Operating Systems Concepts and Design – Pradeep K Sinha - Prentice-Hall India.
3. The Design of the Unix Operating System - Maurice J Bach – Prentice-Hall India.

26CSA562	SOFTWARE TESTING	L-T-P-C: 3-0-0-3

Course Description

Software testing courses equip and help students to understand the various theoretical aspects of the program ranging from manual testing to test automation. Also, the students can understand the skills that are relevant to the industry by getting experience in the latest and advanced technology.

Course Objectives

To study the underlying concepts in software testing and to examine the various software testing issues and find their solutions. Students are also exposed to advanced software testing topics, such as object-oriented software testing activities, methods and tools.

Course Outcomes

COs	Description
CO1	Identify the different software testing techniques, processes and errors handled in software projects.
CO2	Classify black box and white box testing techniques for functional and structural testing and test case designing.
CO3	Describe the different testing activities and levels of testing which aim to uncover the defects in all the project's stages.
CO4	Discuss the non-functional testing and debugging methods.
CO5	Recognize the various issues for object-oriented testing and tools for testing.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	1	1	-	-	-	1	-
CO2	3	2	2	-	-	-	-	1
CO3	2	2	2	-	-	-	1	-
CO4	2	2	-	-	-	-	1	-
CO5	2	2	2	-	-	-	-	-

Syllabus

Unit I

Introduction: Introduction to software testing and analysis - Purpose of Software testing – Some Dichotomies – a model for testing - Error, Fault, Failure, Incident, Test Cases, Testing Process, Limitations of Testing - No absolute proof of correctness.

Software testing Fundamentals - Specification-based testing techniques, code-based testing techniques, Model-based testing.

Unit II

Blackbox box testing- Functional Testing: Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique.

Whitebox testing- Structural Testing: Path testing, DD-Paths, Cyclomatic Complexity, Graph Metrics, Data Flow Testing, Mutation testing, Static Analysis, Dynamic Analysis. Reducing the number of test cases: Prioritization guidelines, Priority category, Scheme, Risk Analysis, Regression Testing, Slice-based testing

Unit III

Testing Activities - Unit Testing, Levels of Testing, Integration Testing, System Testing, Debugging, Domain Testing, Regression Testing, Acceptance testing,

Unit IV

Object Oriented Testing: Issues in Object Oriented Testing, Class Testing, GUI Testing, Object Oriented Integration and System Testing, Methods of test data generation and validation.

Unit V

Program slicing and its application, Reliability analysis, Formal methods; verification methods; oracles. Testing Tools: Static Testing Tools, Dynamic Testing Tools, and Characteristics of Modern Tools

Textbooks / References:

1. William Perry, "Effective Methods for Software Testing", John Wiley & Sons, New York, 2007.
2. Cem Kanerner, Jack Falk, Nguyen Quoc, "Testing Computer Software", Second Edition, Van Nostrand Reinhold, New York, 2000.
3. Boris Beizer, "Software Testing Techniques", Second Volume, Second Edition, Van Nostrand Reinhold, New York, 1990.
4. Louise Tamers, "Software Testing", Pearson Education Asia, 2002
5. "Software Testing: A Craftsman's Approach, Second Edition," by Paul C Jorgensen, CRC Press, June 26, 2002. (required)
6. "The Art of Software Testing," 2nd ed., Glenford J. Myers, John Wiley & Sons, Inc., Hoboken, New Jersey, 2004. (optional)
7. "Lessons Learned in Software Testing: a Context-Driven Approach," Cem Kaner, James Bach, and Bret Pettichord, John Wiley & Sons, Inc., New York, 2002. (optional).

26CSA563	THEORY OF COMPUTATION	L-T-P-C: 3-0-0-3

Course Description
Formal languages and automata theory deals with the concepts of automata, formal languages, grammar, computability and decidability. The reasons to study. Automata Theory possesses a high degree of permanence and stability, in contrast with the ever-changing paradigms of the technology, development, and management of computer systems.

Course Objectives
This course gives an overview of the theoretical foundations of computer science from the perspective of formal languages. Formal Languages and Automata Theory provide a simple, elegant view of the complex machine we call a computer. Further, parts of the Automata theory have a direct bearing on circuit design, compiler design, and search algorithms.

Course Outcomes

COs	Description
CO1	Explain kinds of finite automata and their capabilities.
CO2	Design Finite Automata for different Regular Expressions and Languages.
CO3	Construct context-free grammar for various languages.
CO4	Solve various problems by applying normal form techniques, push down automata and Turing Machines.
CO5	Explain Recursively enumerable languages

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	2	-	-	-	-	-	2
CO2	2	2	-	-	-	-	-	2
CO3	3	2	-	-	1	-	-	2
CO4	3	2	-	-	-	-	-	2
CO5	3	2						2

Syllabus

Unit I
Finite Automata (FA): Introduction, Deterministic Finite Automata (DFA) -Formal definition, simpler notations (state transition diagram, transition table), the language of a DFA. Nondeterministic Finite Automata (NFA)-Definition of the FA, language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machine Interconversion.

Unit II
Regular Expressions (RE): Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA's to Regular Expressions, Converting Regular Expressions to Automata, applications of Regular Expressions. REGULAR GRAMMARS: Definition, regular grammar and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular -Pumping lemma, applications, Closure properties of regular languages.

Unit III

Context Free Grammar (CFG): Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFGs, Minimization of CCFGs CNF, GNF, Pumping Lemma for CFLs Enumeration of Properties of CFL

Unit IV

Pushdown Automata: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA. TURING MACHINES (TM): Formal definition and behavior, Languages of a TM, TM as accepters, and TM as a computer of integer functions, Types of TMs.

Unit V

Recursive And Recursively Enumerable Languages (REL): Properties of recursive and recursively enumerable languages, Universal Turing machine, The Halting problem, Undecidable problems about TMs. Context-sensitive language and linear bounded automata (LBA), Chomsky hierarchy, Decidability, Post's correspondence problem (PCP), undecidability of PCP.

Textbooks / References:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2007), Introduction to Automata Theory Languages and Computation, 3rdedition, Pearson Education, India.
2. Martin, John C., Introduction to Languages and the Theory of Computation, 3rd ed., Tata McGraw Hill Education Private Limited.
3. H.R.Lewis and C.H.Papadimitriou, ―Elements of the theory of Computation‖, Second Edition, PHI, 2003.
4. Micheal Sipser, ―Introduction of the Theory and Computation‖, Thomson Brokecole, 1997.
5. Peter Linz, "An Introduction to Formal Languages and Automata", Third Edition, 2002.

26CSA564 ENTERPRISE RESOURCE PLANNING MANAGEMENT L-T-P-C: 3-0-0-3

Course Description

This course deals with ERP to improve the productivity of your organization's processes and product life cycle management in a company. Adopt the necessary skills to select and implement the most suitable ERP system for your business.

Course Objectives

To facilitate the flow of information between all business functions inside the organization's boundaries and manage connections to outside stakeholders.

Course Outcomes

Cos	Description
CO1	Demonstrate significance and principles of BE.
CO2	Use Business modelling concepts for ERP and its implementation.
CO3	Describe the concept of ERP and the competitive strategy and different ERP domains.
CO4	Examine market dynamics and competitive strategy of ERP using case studies.
CO5	Interpret ERP and client-server architecture, open-source ERP and commercial ERP.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	2	-	-	-	-		-
CO2	2	2	-	-	-	-		-
CO3	2	2	-	-	-	-		-
CO4	2	3	-	-	-	-		-
CO5	2	2	3	-	-	-		-

Syllabus

Unit I

Introduction to ERP

Accommodating Variety – Integrated Management Information – Seamless Integration – Supply Chain Management – Resource Management – Integrated Data Model – Scope – Technology – Benefits of ERP.

Business Engineering and ERP

What is BE? – Significance and Principles of BE – BPR, ERP and IT – BE with IT – ERP and Management Concerns.

Unit II

Business Modelling for ERP

Building the Business Model.

ERP Implementation

Role of Consultants, Vendors and Users – Customization – Precautions – ERP: Post-implementation Options – ERP Implementation Methodology – Guidelines for ERP Implementation.

Unit III

ERP and the Competitive Advantage

ERP and the Competitive Strategy
The ERP Domain
MFG/PRO, IFS/Avalon - Industrial and Financial systems – Baan IV – SAP – SAP R/3 Applications – Example of an Indian ERP Package – The Arrival of ERP III.

Unit IV
Marketing of ERP
Market Dynamics and Competitive Strategy.
Sample Case Studies.

Unit V
Client Server and ERP Architecture
Introduction to Client Server – Advantages and Disadvantages – N tier Architecture – ERP Architecture.
http://ebuild.imtindia.com/erp_software_architecture.html
Open Technology
Background of Open Technology – Introduction – Proprietary v/s Open source – Need for Open-Source Solutions – Open-Source ERP.
http://elearning.nic.in/mdp/2-open-technology/opentechnology-mdp.pdf
Commercial ERP
Commercial ERP – Open-Source ERP v/s Commercial ERP.
http://www.erpwire.com/erp-articles/commercial-and-open-source-erp.htm.

Textbooks / References:

1. Enterprise Resource Planning – Concepts and Practice", Vinod Kumar Garg, N.K. Venkitakrishnan, Second Edition, Eastern Economy Edition, Prentice-Hall of India Pvt., Ltd., 2008.

26CSA565 AUTOMATION AND ROBOTICS L-T-P-C: 3-0-0-3

Course Description

This course provides the detailed idea about the fields of robotics and its control mechanisms.

Course Objectives

- The main objective is to provide information on various parts of robots and ideas on robotics.
- It also focuses on various kinematics and inverse kinematics of robots, trajectory
- planning of robots and to study the control of robots for some specific applications.

Course Outcomes

COs	Description
CO1	Describe the fields of robotics and explain the major components
CO2	Explain about various robot processes and functions
CO3	Discuss the various Programmable Logic Control and Experiment with various control mechanisms of robotics.
CO4	Explain the kinematics of robots and trajectory,
CO5	Implement different applications of robotics

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	-	-	-	-	-	3
CO2	2	1	-	-	-	-	-	1
CO3	3	-	2	-	1		3	
CO4	-	1	-	-	-	-	2	1
CO5	2		-	-	1	1	-	-

Syllabus

Unit I

Introduction - Definition and Origin of Robotics, Types of Robotics, Major Components, Historical development of Robot, Robotic System and Robot anatomy, Degrees of freedom, Coordinate System and its type Asimov's laws of robotics, Dynamic stabilization of robots.

Unit II

Power Sources and Sensors - Hydraulic, pneumatic and electric drives, determination of HP of motor and gearing ratio, variable speed arrangements, path determination, micro machines in robotics, machine vision, ranging, laser, acoustic, magnetic, fibre optic and tactile sensors.

Unit III

Manipulators, Actuators, and Grippers - Manipulators, Classification, Construction of manipulators, manipulator dynamics and force control, electronic and pneumatic manipulator control, End effectors, Loads and Forces, Grippers, design considerations, Robot motion Control, Position Sensing.

Unit IV

Kinematics and Path Planning - Solution of Inverse Kinematics Problem, Multiple Solution Jacobian Work Envelop, Hill Climbing Techniques, Robot Programming Languages. Process Control and Types, On-Off Control Systems, Proportional Control Systems, Proportional Plus Integral (PI) Control Systems, Three Mode Control

(PID) Control Systems, Process Control Tuning.

Case Studies:
Multiple robots, Machine Interface, Robots in Manufacturing and not-Manufacturing Application, Robot Cell Design, Selection of a Robot.

Laboratory Works:
The laboratory work should be focused on the implementation of sensors, design of control systems. It should also deal with developing programs related to Robot design and control using python.

Textbooks / References:

1. Mikell P. Weiss G.M., Nagel R.N., Odraj N.G., Industrial Robotics, McGraw Hill.
2. Ghosh, Control in Robotics and Automation: Sensor Based Integration, Allied Publishers.
3. Jain K.C. and Aggarwal B.E., Robotics – Principles and Practice, Khanna Publishers
4. Schuler, C.A. and McNamee, W.L. Modern Industrial Electronics, Macmillan/McGraw-Hill
5. Klafter R.D., Chimielewski T.A., Negin M., Robotic Engineering – An Integrated Approach, Prentice Hall of India.
6. Deb.S.R., Robotics Technology and Flexible Automation, John Wiley, USA 1992.
7. Asfahl C.R., Robots and Manufacturing Automation, John Wiley, USA 1992
8. Mc Kerrow P.J. Introduction to Robotics, Addison Wesley, USA, 1991.
9. Issac Asimov I. Robot, Ballantine Books, New York, 1986.

26CSA566	SOFTWARE DEFINED NETWORKS	L-T-P-C: 3-0-0-3

Course Description

This course introduces the fundamental concepts of computer networking and provides a conceptual understanding of Software Defined Networks (SDN). It focuses on the separation of control and data planes, SDN architectures, controllers, southbound and northbound interfaces, and network programmability. The course also examines real-world industrial deployment scenarios and use cases of SDN, highlighting its role in modern data centers, cloud computing, and large-scale enterprise networks. Students gain insight into how SDN improves network flexibility, scalability, and manageability compared to traditional networking approaches.

Course Objectives

This course aims to develop knowledge in networking fundamentals, gain a conceptual understanding of Software Defined Networks (SDN) and study industrial deployment use-cases of SDN.

Course Outcomes

Cos	Description
CO1	Differentiate between traditional networks and software defined networks and learn the fundamentals of software defined networks.
CO2	Describe characteristics of SDN
CO3	Explain Open SDN Implementations
CO4	Use SDN in data centers
CO5	Apply SDN concepts to solve real time world problems.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	-	-	-	-	-
CO2	3	1	1	-	-	-	-	1
CO3	3	2	-	-	-	-	-	2
CO4	3	2	2	-	-	-	-	2
CO5	3	2	2	-	-	-	-	2

Syllabus

Unit I
Basic Packet Switching Terminology, Historical Background, The Modern Data Center, Traditional Switch Architecture, Autonomous and Dynamic Forwarding Tables, Open Source and Technological Shifts. Why SDN? Genesis of SDN.

Unit II
Working of SDN- Fundamental Characteristics of SDN, SDN Operation, SDN Devices, SDN Controller, SDN Applications, Alternate SDN Methods. Introduction to OpenFlow Specification, Improving OpenFlow Interoperability, OpenFlow Limitations, Optical Transport Protocol Extensions.

Unit III
Introduction to Open SDN and its limitations, SDN via APIs, SDN via Hypervisor Based Overlays, SDN via Opening up the Device, Introduction of SDN Controllers and its general concepts, Layer 3 Centric, Plexxi, Cisco OnePK. Introduction of Network Programmability, Management Interface, Application-Network Divide, Modern Programmatic Interfaces, I2RS, Modern Orchestration

Unit IV
SDN in the Data Center- Introduction of Data Center and its demands, Tunneling Technologies for the Data Center, Path Technologies in the Data Center, Ethernet Fabrics in the Data Center, SDN Use Cases in the Data Center, Comparison of Open SDN, Overlays and APIs, Real-World Data Center Implementations.

Unit V
Introduction SDN application and its usages, SDN in the Data Center - SDN in Other Environments - SDN Applications - SDN Use Cases – The Open Network Operating System.

Textbooks / References:

1. Paul Goransson and Chuck Black, Software Defined Networks: A Comprehensive Approach, Second Edition, Morgan Kaufmann, 2014.
2. SDN: Software Defined Networks, An Authoritative Review of Network Programmability Technologies, By Thomas Nadeau, Ken Gray, Publisher: O'Reilly Media.
3. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud 1st Edition, Kindle Edition, by William Stallings.
4. SDN and NFV Simplified: A Visual Guide to Understanding Software Defined Networks and Network Function Virtualization 1st Edition, Kindle Edition, by Jim Doherty.

26CSA567	ROBOTIC OPERATING SYSTEM	L-T-P-C: 3-0-0-3

Course Description

This course introduces the Robot Operating System (ROS) including many of the available tools that are commonly used in robotics. With the help of different examples, the course should provide a good starting point for students to work with robots. They learn how to create software including simulation, to interface sensors and actuators, and to integrate control algorithms.

Course Objectives

- Introduce the basics of Robot Operating Systems and its architecture.
- Provide knowledge on the hardware interfacing aspects.
- Analyze the working of ROS in real world complex applications.

Course Outcomes

Cos	Description
CO1	Explain the Role of ROS in real time scenario and its significance.
CO2	Apply the Linux commands in ROS used in robotics.
CO3	Discuss the concepts behind navigation through file system.
CO4	Analyze and debug the node created using hardware for application.
CO5	Analyze the issues in hardware interfacing and implement the working of specific application hardware using Hardware with ROS.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	1	1	-	1	-	-	-
CO2	-	3	2	2	-	-	-	-
CO3	1	2	1	-	-	-	-	-
CO4	1	-	1	1	-	-	-	-
CO5	1	-	-	1	1	-	-	-

Pre-requisites:

- Introduction to Linux and terminal commands
- Basics of Python programming.

Syllabus

Unit I

Introduction –The ROS Equation - History - distributions -difference from other meta-operating systems– services - ROS framework – operating system – releases. ROS Best Practices: ROS Local Setup guidelines, Using open-source packages with ROS, ROS Unit tests and ROS Bags.

Unit II

UNIX commands - file system – redirection of input and output - File system security - Changing access rights – process commands – compiling, building and running commands – handling variables.

Unit III

File system - packages – stacks – messages – services – catkin workspace – working with catkin workspace – working with ROS navigation and listing commands.

Unit IV

Navigation through file system -Understanding of Nodes – topics – services – messages – bags – master – parameter server. Introduction to the ROS Navigation Stack, Navigation stack-creating transforms.

Unit V

Debugging of Nodes – topics – services – messages – bags – master – parameter – visualization using Gazebo – Rviz – URDF modeling – Xacro – launch files. Hardware Interface: Sensor Interfacing – Sensor Drivers for ROS – Actuator Interfacing – Motor Drivers for ROS. Case Studies: Using ROS In Real World Applications.

Textbooks / References:

1. Lentin Joseph, "Robot Operating Systems (ROS) for Absolute Beginners, Apress, 2018
2. Aaron Martinez, Enrique Fernández, "Learning ROS for Robotics Programming", Packt Publishing Ltd, 2013.
3. Reference Books: 1. Jason M O'Kane, "A Gentle Introduction to ROS", CreateSpace, 2013.
4. AnisKoubaa, "Robot Operating System (ROS) – The Complete Reference (Vol.3), Springer, 2018.
5. Kumar Bipin, "Robot Operating System Cookbook", Packt Publishing, 2018.
6. Wyatt Newman, "A Systematic Approach to learning Robot Programming with ROS", CRC Press, 2017.
7. Patrick Gabriel, "ROS by Example: A do it yourself guide to Robot Operating System", Lulu, 2012.

26CSA568 SOFTWARE QUALITY ASSURANCE L-T-P-C: 3-0-0-3

Course Description

This course helps and throws an insight into students by making then understand that all software engineering processes, methods, activities, and work items are monitored and comply with the defined standards. The course also incorporates all software development processes starting from defining requirements to coding until release. Its prime goal is to make students aware of different quality standards and its management methods.

Course Objectives

To convey quality management processes, various activities of quality assurance, quality planning and quality control. Students understand the importance of standards in the quality management process and their impact on the final product.

Course Outcomes

Cos	Description
CO1	Recognize the quality challenges, factors and activities in the project life cycle.
CO2	Describe the idea on the testing strategies and building a testing process.
CO3	Identify the software quality in management and business context. Also, regarding Process and Product Quality.
CO4	Explain the ISO origins, different audit methods and quality assessment procedures.
CO5	Execute a clear-cut idea on CMM and Process improvement models, Configuration Management and processes along with experience gaining through various case studies.

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	1	-	-	-	1	2
CO2	3	2	2	-	-	-	-	-
CO3	2	2	2	-	-	-	1	-
CO4	2	2	2	-	-	-	-	2
CO5	2	2	-	-	1	-	1	-

Syllabus

Unit I

Introduction: The Software Quality Challenge - Software Quality Factors-The Components of Software Quality System-Integrating Quality Activities in the Project Life Cycle.

Unit II

Software Testing: Strategies and Implementation-Building the Software Testing Process-Software Quality- Five Views of Software Quality, McCall’s Quality Factors and Criteria, Quality Factors Quality Criteria, Relationship

Unit III

Quality Factors and Criteria- Management Components: Metrics and Costs-Software Quality in the Business Context- Product Quality and Process Quality

Unit IV

ISO 9001: The Origins of ISO 9001- need for ISO 9001-Assessment and Audit Preparation-The Assessment Process.

Unit V

Software CMM and other Process Improvement Models-Software Configuration Management-Introduction to Six Sigma - Case Studies: Indian Software Industry in Perspective.

Textbooks / References:

1. Daniel Galin, “Software Quality Assurance: From theory to Implementation”, Pearson Education, 2008
2. Nina Godbole, “Software Quality Assurance, Principles and Practice”, NarosaPublications, 2011.
3. William Perry, “Effective Methods of Software Testing”, Third Edition, Wiley, 2006.

26CSA569	WEB SERVICES	L-T-P-C: 3-0-0-3

Course Description

This course deals with how students will be able to define a web service, deploy a web service within WSDL, understand the SOAP protocol, read and understand SOAP messages passed between server and client. It will also explain how to register and discover the service.

Course Objectives

- To Understand Web Services and implementation model for SOA.
- To Understand the SOA, its Principles and Benefits.
- To Understand XML concepts and paradigms needed for testing Web Services.

Course Outcomes

Cos	Description
CO1	Describe architecture of web services
CO2	Illustrate web service architecture and characteristics.
CO3	Use SOAP for inter application communication.
CO4	Implement framework using WSDL
CO5	Describe Registering and Discovering Services

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	-	1	-	-	-
CO2	3	-	-	-	-	-	-	-
CO3	-	-	3	-	2	-	-	-
CO4	-	3	1	-	2	-	-	-
CO5	1	-	2	-	1	-	-	-

Prerequisites

Programming languages

Syllabus

Unit I

Evolution and Emergence of Web Services – Evolution of distributed computing. Core distributed computing technologies – client/server, CORBA, JAVA RMI, Microsoft DCOM, Challenges in Distributed Computing, Introduction to Web Services – The definition of web services, basic operational model of web services, tools and technologies enabling web services, benefits and challenges of using web services.

Unit II

Web Service Architecture – Web services Architecture and its characteristics, core building blocks of web services, standards and technologies available for implementing web services, web services communication, basic steps of implementing web services. Brief Overview of XML – XML Document structure, XML namespaces, Defining structure in XML documents, Reuse of XML schemes, Document navigation and transformation.

Unit III

SOAP: Simple Object Access Protocol, Inter-application communication and wire protocols, SOAP as a messaging protocol, Structure of a SOAP message, SOAP envelope, Encoding, Service Oriented Architectures, SOA revisited, Service roles in a SOA, Reliable messaging, Message Exchange Patterns, Message Exchange

Formats.

Unit IV

Describing Web Services – WSDL introduction, nonfunctional service description, WSDL1.1 Vs WSDL 2.0, WSDL document, WSDL elements, WSDL binding, WSDL tools, WSDL port type, limitations of WSDL.

Unit V

Registering and Discovering Services: The role of service registries, Service discovery, Universal Description, Discovery, and Integration, UDDI Architecture, UDDI Data Model, Interfaces, UDDI Implementation.

Textbooks / References:

1. Web Services & SOA Principles and Technology, Second Edition, Michael P. Papazoglou.
2. Developing Java Web Services, R. Nagappan, R. Skoczylas, R.P. Sriganesh, Wiley India
3. Thomas Erl, “Service Oriented Architecture”, Concepts, Technology and Design”, Prentice Hall of India, 2005.

26CSA570	**SOFTWARE PROJECT MANAGEMENT**	**L-T-P-C: 3-0-0-3**

Course Description

This course provides a comprehensive understanding of software project management principles, processes, and practices. It introduces the concepts of software projects, their importance, and their relationship with programs and portfolios. The course covers project management process groups and knowledge areas as defined by PMBOK, along with common problems encountered in software projects. Students will learn project evaluation techniques, project integration, scheduling, resource, communication, and risk management. Practical exposure using software project management tools and case studies enables students to apply theoretical concepts to real-world software projects.

Course Objectives

- To understand the fundamentals and significance of software project management
- To familiarize students with PMBOK process groups and knowledge areas
- To analyze cost–benefit and feasibility of software projects
- To apply scheduling, resource, communication, and risk management techniques
- To develop skills in planning, monitoring, controlling, and closing software projects
- To use project management tools for practical project execution and analysis

Course Outcomes (COs)

COs	**Description**
CO1	Explain the concepts, importance, and challenges of software project management
CO2	Apply project management processes and knowledge areas to software projects
CO3	Evaluate software projects using cost–benefit and feasibility techniques
CO4	Develop project charters, project management plans, and schedules
CO5	Manage project resources, communication, and risks effectively
CO5	Use project management tools to plan, monitor, and control software projects

CO–PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	3	-	-	1	-	-	3
CO2	3	2	2	-	-	-	-	-
CO3	-	2	2	2	2	-	-	-
CO4	3	2	1	1	2	-	-	-
CO5	2	2	2	2	1	-	-	-

Syllabus

Unit I: Introduction to Software Project Management

Definition and overview of software projects,Importance of software project management, Relationship between project, program, and portfolio, Key components of a software project, Project management process, process groups, and knowledge areas, Problems with software projects (Bob Hughes).

Unit II: Project Evaluation and Integration Management
Evaluation of individual projects, Cost–benefit evaluation techniques (Bob Hughes), Project charter and project management plan, Organizational process assets, Project integration management overview

Unit III: Project Integration Management Processes
Develop project charter, develop project management plan, Direct and manage project work, Monitor and control project work, Integrated change control, Project closure

Unit IV: Project Schedule, Resource, and Communication Management
Schedule management,Plan schedule management: Define and sequence activities,Estimate activity durations, develop schedule, Resource management: Plan and estimate resources, Acquire resources, Develop and manage teams, Control resources, Project communication management overview

Unit V: Project Risk Management
Plan risk management, Identify risks,Qualitative risk analysis, Risk response planning, Monitor and control risks

Text Books / References

1. PMBOK® Guide, Sixth Edition, Part 1
2. Mike Cotterell & Bob Hughes, *Software Project Management*, Fifth Edition, Tata McGraw-Hill, 2010
3. Software project management tools for case studies (e.g., ProjectLibre)

ELECTIVES – 4 Credits

AI & DS STREAM

26CSA631 DATA MODELLING AND VISUALIZATION L-T-P-C: 3-0-2-4

Course Description

Data Modeling and Visualization is delivered to explore the domain of data analysis using the methods of data visualization and data modelling countering the challenges and risks associated with data analytics. The learners of this course are engaged with extensive learning methods and evaluation techniques to gain theoretical and practical knowledge of the concepts.

Course Objectives

- To understand the concepts from fundamental to the advanced level of data analysis aided by the strategies of data visualization and data modelling.
- The students are familiarized with the tools of R programming for data analysis and acquire competence in problem-solving and decision-making.
- To learn and implement various data modelling algorithms.

Course Outcomes

COs	**Description**
CO1	Understand data and its processing techniques using Python.
CO2	Apply data modeling techniques and visualize data using R.
CO3	Design and Apply data visualization using Tableau or Power BI
CO4	Analyze and visualize data modelling using different types of data sources using various techniques.

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	1	-	2	1	2
CO2	3	2	3	1	1	2	1	2
CO3	3	2	3	2	1	2	1	2
CO4	3	2	3	-	2	2	1	2

Syllabus

Unit I

Data-Analytic Thinking: Knowing your data, Data pre-processing, Storytelling with data. Data Visualization using Python: Introduction to Python programming, Visualization using Python, Transformation using Python, exploratory data analysis.

Unit II

Data Visualization using R: Introduction to R programming, Visualization using R, Transformation using R, exploratory data analysis. Data Modeling: Linear regression, Logistic regression, K-nearest neighbors, K-means clustering, Performance measure, Implementation of some modelling algorithms using python.

Unit III

Data Visualization using Tableau and Power BI: Introduction to Tableau, data import and management, data type and operations, Different types of data visualizations, dashboards, storytelling, Understanding of the concepts of dynamic/interactive data visualization and report generation.

Unit IV

Data Modeling from Different Data Sources for Visualization: Understanding structured, unstructured and semi-structured data sources, Data modelling and creating visualization charts/dashboards from structured data like databases (SQL and NoSQL), Data modelling and creating visualization charts/dashboards from semi-structured data like CSV files, XML, JSON and others, Data modelling and creating visualization charts/dashboards from live streaming data.

Textbooks / References:

1. Data Analysis and Visualization Using Python: Analyze Data to Create Visualizations for BI Systems by Embarak, Apress.
2. Jiawei Han, Micheline Kamber and Jian Pei, “Data mining concepts and Techniques”, Third Edition, Elsevier Publisher, 2006.
3. K.P.Soman, Shyam Diwakar and V.Ajay, “Insight into data mining Theory and Practice”, Prentice Hall of India, 2006.
4. Data Science with R: A Step-by-Step Guide with Visual Illustrations & Examples, Andrew Oleksy.
5. Practical Data Science with R, Nina Zumel and John Mount, Dreamtech/Manning, 2014
6. R Programming for Data Science, Roger D. Peng, Lean Publishing, 2015.
7. Tableau Desktop Cookbook, Lorna Brown, O’Reilly

26CSA632	EXPLORATORY DATA ANALYSIS	L-T-P-C: 3-0-2-4

Course Description

This course covers the essential exploratory techniques for summarizing data that are typically applied before formal modelling commences and for the development of more complex statistical models. Exploratory techniques are also important for sharpening potential hypotheses about the real-world phenomena that can be addressed by the data.

Course Objectives

- To understand the challenges involved in working with real-time datasets.
- To learn Python programming tools and algorithms for data analysis and problem solving.
- To explore data visualization techniques for effective interpretation of data.

Course Outcomes

COs	Description
CO1	Use python/R libraries for exploratory computing and explore data wrangling techniques
CO2	Examine Cumulative Distribution functions and Probability Density functions
CO3	Conceive and apply the knowledge of data aggregation and group operations
CO4	Explore and solve problems on various supervised and unsupervised learning algorithms

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	1	1	2	-	2
CO2	2	3	3	3	1	1	-	1
CO3	3	3	3	3	1	2	-	1
CO4	4	4	3	3	-	1	-	1

Syllabus

Unit I

Introduction to exploratory data analysis and and Automated EDA what it is used for – Introduction to machine learning – about the dataset, data Preprocessing techniques, Data wrangling: Join, combine and reshape, numerical summarization, visualization – statistical learning and model selection, Prediction accuracy, cross-validation.

Unit II

Cumulative Distribution functions: Percentiles, CDF's, Percentile based statistics, Modeling Distributions: Exponential Distributions, Normal distributions, Normal probability plot, long normal distributions, Descriptive statistics– location, spread, Probability Density functions: PDFs, Kernel density estimation, Distribution framework, Skewness, Relationship between variables: Correlation, covariance, Pearson's correlation, Non-linear relationship, Estimation: Sampling distribution, sampling Bias, Exponential distributions, Hypothesis testing, Regression.

Unit III

Data aggregation and group operations – group by Mechanics, Data aggregation, group-wise operations and transformations, Pivot tables and cross-validation. Time series: Date Ranges, frequencies and shifting, Time zone handling, Period arithmetic, Resampling and frequency conversion and moving window function.

Unit IV

Supervised Learning algorithms – Classification, Forecasting, prediction and regression, Linear Models, SVM, KNN, Decision tree classifier, Artificial neural networks, Ensemble methods, Deep neural networks. Unsupervised learning algorithms: K-means, association rule mining, reinforcement learning, Instance-based learning.

Textbooks / References:

1. Practical machine learning for data analysis using python, Abdul Hamid Subasi, Elsevier Publication
2. Think Stats exploratory data analysis, Allen B. Downey, 2nd edition
3. Python for data analysis, Oreily, Wes McKinney.

26CSA633	DATA MINING AND APPLICATIONS	L-T-P-C: 3-0-2-4

Course Description

This course provides an introduction to data mining, covering the fundamental concepts, algorithms, principles, implementation techniques and applications of data mining. The course also explores applications of data mining across various domains to understand how data mining concepts can be used to extract meaning patterns and knowledge.

Course Objectives

- To understand concepts of pattern discovery
- To familiarise with data preprocessing and mining algorithms
- To understand prediction algorithms and cluster analysis

Course Outcomes

Cos	Description
CO1	Understand key concepts, methods, and applications emphasizing data preprocessing techniques for effective data mining.
CO2	Identify and apply efficient pattern mining association methods and rules
CO3	Learn pattern-based classifications and prediction techniques
CO4	Understand and apply classifier evaluation methods
CO5	Study the fundamental concepts, methods, and applications of cluster analysis

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	1	1	1	2	-	1	1	2
CO2	3	3	3	2	-	1	2	2
CO3	3	3	3	2	1	1	1	2
CO4	3	3	3	2	1	1	1	2
CO5	3	3	3	2	1	1	1	2

Syllabus

Unit I

Introduction: Evolution and Importance of Data Mining-Types of Data and Patterns Mined-Technologies-Applications-Major issues in Data Mining. Knowing about Data-Data Preprocessing: attribute type, Basic statistical descriptions of data, measuring data similarity and dissimilarity, Cleaning– Integration–Reduction–PCA, Data Transformation and Discretization.

Unit II

Mining Frequent Patterns: Basic Concept – Frequent Item Set Mining Methods – Mining Association Rules – Association to Correlation Analysis.

Unit III

Classification and Prediction: Issues - Decision Tree Induction - Bayesian Classification – Rule-Based Classification – k-Nearest-Neighbour Classification - Linear SVM - Regression – Linear, Logistic - Accuracy and Error measures –Introduction to Ensemble methods.

Unit IV

Classifier Evaluation methods- Introduction to Classifier Evaluation, Evaluation Metrics for Classification, Probabilistic and Threshold-Based Measures, Model Validation Techniques, Evaluation for Imbalanced Datasets, Statistical Significance Testing

Unit V

Clustering: Overview of Clustering – Types of Data in Cluster Analysis – Major Clustering Methods-Partitioning Methods- k-Means, k-Medoids. Hierarchical Methods: Agglomerative and Divisive hierarchical clustering-single linkage, complete linkage, average linkage. Density-Based Methods- DBSCAN, Graph-based clustering (CHAMELEON), Grid-based Clustering: CLIQUE, probabilistic Model-Based Clustering-EM algorithm. Data mining trends and research frontiers- Mining complex Data types- Mining other kinds of data-data mining applications.

Textbooks / References:

1. Jiawei Han, Micheline Kamber and Jian Pei, “Data mining concepts and Techniques”, Third Edition, Elsevier Publisher, 2006.
2. K.P.Soman, Shyam Diwakar and V.Ajay, “Insight into data mining Theory and Practice”, Prentice Hall of India, 2006.
3. Yanchang Zhao, “R and Data Mining”, Elsevier, 2013 4. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow, O'Reilly Media, 2017 5. Itay Lieder, Yehezkel Resheff, Tom Hope, Learning TensorFlow, O'Reilly Media, 2017.

26CSA634 MACHINE LEARNING L-T-P-C: 3-0-2-4

Course Description

This course will enable students to understand the basic concepts of machine learning and apply different machine learning models to real-world problems. It covers essential machine learning paradigms including supervised, unsupervised and basic reinforcement learning techniiques.

Course Objective(s)

- To understand the fundamentals of machine learning and its applications.
- To familiarize the machine learning models like linear and logistic regression
- To understand different classifiers and clustering algorithms.

Course Outcomes

COs	Description
CO1	Explain the fundamental concepts, goals, and applications of machine learning, and differentiate between supervised, unsupervised, and reinforcement learning paradigms.
CO2	Select and apply supervised algorithms and analyze the models using different evaluation parameters.
CO3	To build ML models using ANN
CO4	To Apply different clustering algorithms for the unlabeled data and interpret the results.
CO5	Ability to recognize and implement various ways of selecting suitable model parameters for different machine learning techniques

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	2	-	-	1	-	2	-
CO2	2	3	2	1	2	-	-	-
CO3	2	3	2	2	2	-	-	-
CO4	2	2	2	2	3	-	-	1
CO5	2	2	3	2	3	1	2	2

Syllabus

Unit I: Introduction to Machine Learning

Introduction to Machine Learning: definition, goals, and applications. Types of learning: Supervised, Unsupervised, and Reinforcement learning. Aspects of developing a learning system: training, validation, and testing data. Machine learning workflow: data collection, preprocessing, model training, validation, and testing.

Unit II: Supervised Learning

Linear regression: assumptions, Sum of Squared Errors (SSE), Mean Squared Error (MSE), and gradient descent optimization. Performance metrics: R^2 score. Bias–variance trade-off, overfitting and underfitting. Regularization techniques: Ridge (L2) and Lasso (L1) regression. Logistic regression and decision boundaries. Probabilistic learning: fundamentals of probability theory and Bayes' rule. Naïve Bayes classifier. Instance-based and margin-based learning: k-Nearest Neighbors (k-NN) and Support Vector Machines (SVM). Tree-based learning: Decision Trees, Model evaluation metrics: accuracy, precision, recall, F-measure, and Receiver Operating Characteristic (ROC) curve.

Unit III: Neural Networks

Artificial Neural Networks: Perceptron model, multilayer feedforward neural networks, and backpropagation algorithm. Activation functions and loss functions.

Unit IV: Unsupervised Learning

Unsupervised learning and learning from unclassified data. Clustering techniques: k-means clustering (partition-based clustering), Hierarchical clustering, Density-based clustering, Expectation Maximization (EM) for soft clustering

Unit V: Model Selection, Ensembles, and Modern Trends

Model selection and validation: Model selection criteria and minimum description length principle. Feature engineering and feature selection: Dimensionality reduction: Principal Component analysis (PCA) and Linear Discriminant Analysis (LDA). Ensemble learning methods: Bagging, Boosting, Random Forests, and AdaBoost classifier. Modern Gradient Boosting Frameworks: Advanced study of Gradient Boosted Decision Trees (GBDT). Practical implementation and hyper-parameter tuning of XGBoost, LightGBM, and CatBoost for high-performance classification and regression on structured data. Current challenges in machine learning. Introduction to deep learning and its applications.

Textbooks/References

1. Kevin P. Murphey, “Machine Learning, a probabilistic perspective”, The MIT Press, 2012.
2. Tom Mitchael, “Machine Learning”, McGraw Hill, 1997.
3. EthemAlpaydin,”Introduction to Machine learning”, PHI learning, MIT Press, 2010, 2nd edition
4. John D. Killeher, Brian Mac, Namee, AoiFE D'Arcy, Fundamental of Machine Learning for Predictive Data Analytics, 2015 MITpress.
5. Alex Smola and SVN. Viswanathan, “Introduction to Machine Learning”, Cambridge University Press, 2008.
6. ShaiShalev-Shwartz and Shai Ben-David, “Understanding Machine Learning: From Theory to Algorithms”, Cambridge University Press,2014.
7. Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, MIT Press, 2016.
8. Duda, Richard, Peter Hart, and David Stork, “Pattern Classification” Second Edition, New York, NY: Wiley Interscience, 2000.
9. Hastie, T., R. Tibshirani, and J. H. Friedman, “The Elements of Statistical Learning: DataMining, Inference and Prediction”, New York, Springer, 2001
10. Christopher, M. Bishop. Pattern Recognition and Machine Learning, Springer-Verlag New York, 2016.
11. Jiawei Han, Micheline Kamber and Jian Pei, “Data mining concepts and Techniques”, Third Edition, Elsevier Publisher, 2006

26CSA635	Explainable AI	L-T-P-C: 3-0-2-4

Course Objectives:

This course provides an advanced understanding of the fundamental principles, need, and scope of explainability and interpretability in AI.

Course Outcome:

CO1	Describe the need and concepts of explainability in AI systems.
CO2	Apply intrinsic and model-agnostic techniques for interpreting machine learning models.
CO3	Implement explainability approaches for deep learning models.
CO4	Analyze AI systems for bias, fairness, and robustness.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	2	1	1	2	1	1	-
CO2	3	3	3	1	2	1	1	-
CO3	3	3	3	1	2	1	1	-
CO4	2	3	2	3	3	1	2	2

Syllabus

UNIT I

Introduction to Machine Learning and Explainable AI, Purpose, need, and limits of explainability and interpretability, Explainability within the model development pipeline, Intrinsic vs. Post-hoc explainability Global and local explanations, properties and evaluation metrics

UNIT II

Intrinsic Explainable Models: Linear/Logistic Regression, Decision Trees, K-Nearest Neighbors. Model transparency and loss function behavior. Model-Agnostic Techniques: Surrogate models, feature importance (global explanations) LIME, SHAP, Kernel Explainer, Integrated Gradients (local explanations) Bagging and Boosting explainability, SHAP for boosted trees.

UNIT III

Explainability in Deep Neural Networks: Post-hoc and agnostic approaches: adversarial features, data augmentation, occlusion methods. Layer-wise analysis: CAM, Grad-CAM, DeepSHAP, DeepLIFT. Counterfactual and causal inference concepts; What-If Tool, Contrastive Explanation Method (CEM). Temporal and Domain-Specific Explainability: Time-series and LSTM attribution using integrated gradients. Explainability for Transformers and Large Language Models (LLMs)

UNIT IV

Bias detection and mitigation in AI models. Fairness and ethical aspects of explainable systems. Adversarial robustness: evasion attacks, preprocessing defenses, adversarial training. Evaluation and certification of robustness. Feature engineering and its role in interpretability. Practical applications and current research directions in XAI

Text/ Reference Books

[1] Explainable AI with Python, Antonio Di Cecco and Leonida Gianfagna, Springer, 2021
[2] Christoph Molnar, Interpretable Machine Learning, Lulu Press, 2022.
[3] Ankur Taly et al., Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Springer, 2019.
Reference
[1] Hands-On Explainable AI (XAI) with Python: Interpret, visualize, explain, and integrate reliable AI for fair, secure, and trustworthy AI apps, Denis Rothman, Packt publisher, 2020
[2] Interpretable Machine Learning with Python: Learn to build interpretable high-performance models with hands-on real-world examples, by SergMasís , Packt publisher, 2021

26CSA636	BIG DATA ANALYTICS	L-T-P-C: 3-0-2-4

Course Description

This course aims to introduce students to the fundamental concepts and characteristics of Big Data and Big Data Analytics. It enables students to understand and apply advanced analytical techniques for processing and analyzing large volumes of diverse datasets. The course familiarizes students with key Big Data technologies and frameworks. In addition, students gain exposure to emerging Big Data tools and ecosystems, along with an understanding of recent trends and real-world applications.

Course Objectives

- To provide an overview of the concepts, scope and significance of the rapidly growing field of big data analytics.
- To introduce essential tools required to manage and analyze Big Data
- To impart the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability.
- To equip students with analytical and technical skills required to address complex real-world problems and support effective decision-making

Course Outcomes

COs	**Description**
CO1	Describe the characteristics of Big Data and use cases from selected business domains.
CO2	Apply NoSQL concepts for big data storage and management.
CO3	Install, configure, and operate Hadoop and HDFS.
CO4	Implement map-reduce analytics using Hadoop.
CO5	Use Hadoop-related tools for big data analytics and understand recent trends in Big Data analytics.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	1	-	-	1	-	-	1
CO2	2	1	2	1	-	-	-	1
CO3	1	2	3	1	-	-	-	-
CO4	1	1	3	1	-	-	-	-
CO5	2	2	3	1	-	-	-	-

Syllabus

Unit I

Introduction to Big Data: What is big data, why big data, the convergence of key trends, unstructured data, industry examples of big data, web analytics, big data and marketing, fraud and big data, risk and big data, credit risk management, big data and algorithmic trading, big data and healthcare, big data in medicine, advertising and big data, big data technologies, introduction to Hadoop, open-source technologies, cloud and big data, mobile business intelligence, Crowd-sourcing analytics, inter and trans firewall analytics.

Unit II

No SQL: Introduction to NoSQL, aggregate data models, aggregates, key-value and document data models, relationships, graph databases, schema less databases, materialized views, distribution models, sharding, master-slave replication, peer-peer replication, sharding and replication, consistency, relaxing consistency, version stamps, map-reduce, partitioning and combining, composing map-reduce calculations.

Unit III

Hadoop: Data format, analyzing data with Hadoop, scaling out, Hadoop streaming, Hadoop pipes, design of Hadoop distributed file system (HDFS), HDFS concepts, Java interface, data flow, Hadoop I/O, data integrity, compression, serialization, Avro, file-based data structures.

Unit IV

MapReduce: MapReduce workflows, unit tests with MRUnit, test data and local tests, anatomy of MapReduce job run, classic Map-reduce, YARN, failures in classic Map-reduce and YARN, job scheduling, shuffle and sort, task execution, MapReduce types, input formats, output formats. Scalable Processing with Apache Spark: Transitioning from classic MapReduce to Spark in-memory processing and Spark SQL

Unit V

Recent Trends in Big Data Analytics: HBase, data model and implementations, HBase clients, HBase examples, praxis. Cassandra, Cassandra data model, Cassandra examples, Cassandra clients, Hadoop integration, Hive, data types and file formats, HiveQL data definition, HiveQL data manipulation, HiveQL queries.

Textbooks / References:

1. Big Data Analytics, Introduction to Hadoop, Spark, and Machine-Learning, Raj Kamal, Preeti Saxena, McGraw Hill, 2018.
2. Big Data, Big Analytics: Emerging Business intelligence and Analytic Trends for Today's Business, Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, John Wiley & Sons, 2013.
3. Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013
4. Hadoop: The Definitive Guide, Tom White, Third Edition, O'Reilly, 2012.
5. Hadoop Operations, Eric Sammer, O'Reilly, 2012.
6. Programming Hive, E. Capriolo, D. Wampler, and J. Rutherglen, O'Reilly, 2012.
7. HBase: The Definitive Guide, Lars George, O'Reilly, 2011.
8. Cassandra: The Definitive Guide, Eben Hewitt, O'Reilly, 2010.
9. Programming Pig, Alan Gates, O'Reilly, 2011.

E-Books:

1. http://index-of.co.uk/Big-Data-Technologies/Data%20Science%20and%20Big%20Data%20Analytics.pdf

26CSA637	NATURAL LANGUAGE PROCESSING	L-T-P-C: 3-0-2-4

Course Description

This course provides a theoretical and methodological foundation in the most widely used and effective current techniques, strategies and toolkits for natural language processing. The ability to harness, employ and analyze linguistic and textual data effectively is highly valuable in academic research, government and the private sectors.

Course Objectives

- Students will be able to comprehend the importance of natural language processing when resolving issues in the real world.
- Enables students to apply and match the proper processing technique to a given situation.
- Students will be able to exhibit the necessary design abilities for large collection sets. Additionally, capable of understanding and presenting cutting-edge, sophisticated NLP research materials to an audience.

Course Outcomes

COs	Description
CO1	Discern the concepts of POS tagging and CFG in English language processing.
CO2	Comprehend the Vector Representation of words and skip-gram models
CO3	Explore semantic analysis algorithms and deep learning techniques to apply them in various NLP applications.
CO4	Gain proficiency in Mathematical and programming tools for implementing NLP applications.

CO-PO Mapping

PO/PSO							PO7	PO8
CO	PO1	PO2	PO3	PO4	PO5	PO6		
CO1	3	2	2	3	1	2	-	1
CO2	2	3	3	-	1	1	3	2
CO3	1	3	3	3	2	2	2	1
CO4	2	3	3	3	1	1	3	1

Syllabus

Unit I

Basics of Machine Learning, Python Programming language, Basics of Probability, Introduction - terminologies - empirical rules – Statistical Properties of words – Probability and NLP – Vector Space Models - Pre-processing-Tokenization, Parts-Of-Speech (POS) tagging, chunking, syntax parsing, Dependency parsing.

Unit II

Vector Representation of words – Contextual Understanding of text – Cooccurrence of matrix – N-grams – Dense Word Vector. Word2Vec – CBOW and Skip-gram Models – One-word learning architecture- Forward pass for Word2Vec – Reduction of complexity – subsampling and negative sampling. Continuous Skip-Gram Model, GloVe, BERT, XLNet.

Unit III

NLP Applications: Named Entity Recognition, Sentiment analysis, Text categorization using Machine learning algorithms, SVD and Latent semantic Indexing, Probabilistic Latent Semantic Indexing (pLSI) and Latent Dirichlet Allocation (LDA).

Deep Learning for NLP: Deep Learning with Hugging Face library; Neural Networks Basics, Feedforward Neural Network, Recurrent Neural Networks, LSTM, An Introduction to Transformers and Sequence-to-Sequence Learning.

Unit IV

Historical Approaches to Machine Translation – Statistical Machine Translation – Translation Models – Healthcare Data analysis and Text visualization: Summarizing lengthy blocks of narrative text, such as a clinical note or academic journal article. Answering unique free-text queries that require the synthesis of multiple data sources. Introduce Mathematical and programming tools to visualize a large collection of text documents.

Textbooks / References:

1. C.D. Manning et al, "Foundations of Statistical Natural Language Processing," MIT Press. MIT Press, 1999. isbn: 9780262133609.
2. James Allen, "Natural Language Processing with Python", O'Reilly Media, July 2009.
3. NiladriSekhar Dash and S. Arulmozi, Features of a Corpus. Singapore: Springer Singapore,2018, pp. 17–34. ISBN: 978-981-10-7458-5.
4. Ian Goodfellow, YoshuaBengio, and Aaron Courville, Deep Learning, http://www.deeplearningbook.org. MIT Press, 2016.
5. NitinIndurkhya and Fred J Damerau," Handbook of natural language processing," Chapman and Hall/CRC, 2010.
6. Daniel Jurafsky and James H. Martin" Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition," 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000. ISBN: 0130950696.

26CSA638	LARGE LANGUAGE MODELS	L-T-P-C: 3-0-2-4

Course Description

This course provides an in-depth exploration of large language models, focusing on understanding their architecture, training and fine-tuning process, applications, and practical use cases. Students will gain hands-on experience with language model tools and libraries.

Course Objectives

- Understand the architecture, components, and training processes of large language models (LLMs).
- Explore techniques for fine-tuning, evaluating, and optimizing LLMs for specific tasks.
- Gain practical experience using LLM tools, frameworks, and libraries for real-world applications.
- Apply LLMs to various domains, including text generation, classification, summarization, and conversational AI.
- Analyze the strengths, limitations, and ethical considerations associated with deploying LLMs.

Course Outcomes

COs	Description
CO1	Explore fundamental principles of Large Language Model (LLM) architectures
CO2	Evaluate the strengths and weaknesses of various LLM architectures for critical/insightful analysis.
CO3	Apply and analyse techniques for training and fine-tuning LLMs for specific tasks
CO4	Apply LLM expertise to address real-world challenges effectively.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	3	2	3	0	0	2	2
CO2	3	3	2	3	0	2	2	2
CO3	3	3	2	3	1	2	2	2
CO4	3	3	2	3	1	2	2	2

Prerequisites:

- Basic knowledge of machine learning and natural language processing
- Familiarity with Python programming.

Syllabus

Unit 1

Computational linguistics: Overview of NLP, Introduction, syntax, semantics, morphology, Word representation: One-hot encoding, Bag-of-Words (BoW), Dictionary: TF-IDF, Embedding: Word2vec, Glove, Fasttext, Language Model: n-gram, Sequences and sequential data. Neural Networks and Deep Learning, Transformer Architecture- Pre-training and Fine-tuning

Unit 2

Introduction to Large Language Models – Decoder-only LLMs: A deep dive into GPT, Encoder Only LLMs-BERT, prompting – different prompting strategies– Instruction tuning – fine tuning – parameter-efficient fine-tuning – quantized fine tuning. Small Language models. Training LLMs using reinforcement Learning.

Unit 3

Evaluating LLMs: Retrieval Augmented Generation, Benchmarks, evaluation frameworks and popular leaderboards. Applications/Case study of Large Language Models - Text Generation, Translation, and Summarization -Question Answering, Sentiment Analysis, Chatbots, Application of LLMs in healthcare and Code generation, latest advancements in LLMs.

Text Books/References

1. Hands-On Large Language Models by Jay Alammar, Maarten Grootendorst, December 2024 Publisher(s): O'Reilly Media.
2. ‘Deep Learning for Natural Language Processing: Develop Deep Learning Models for your Natural Language Problems (Ebook)’, Jason Browlee, Machine Learning Mastery, 2017.
3. Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT by Sudharsan Ravichandiran, Packt Publishing Limited January 2021
4. Latest research papers on LLM
5. Comprehensive Overview of LLMs- A survey paper: https://arxiv.org/pdf/2307.06435
6. ‘Foundations of Statistical Natural Language Processing’, Christopher Manning and Hinrich Schütze, MIT press, 1999
7. ‘Natural Language Processing with Python’, Steven Bird, Ewan Klein and Edward Loper, O'Reilly Media, Inc.", 2009.
8. ‘Speech & language processing’, Daniel Jurafsky, James H Martin, preparation [cited 2020 June 1] Available from: https://web. stanford. edu/~ jurafsky/slp3 (2018).

26CSA639	COMPUTER VISION	L-T-P-C: 3-0-2-4

Course Description

This course is a broad introduction to computer vision covering fundamental concepts and techniques for analysing and interpreting visual data. It includes image types and formats, image processing and transformations, feature extraction for image representation, and video analysis.

Course Objectives

- To understand image types and techniques for image conversion
- Learn and apply different operations on images
- To explore methods for image representation and video analysis

Course Outcomes

COs	Description
CO1	Understand the fundamentals of computer vision, its applications, image types and basic operations
CO2	Apply various image processing techniques to binary images and understand different color spaces
CO3	Demonstrate image representation using different feature extraction algorithms
CO4	Analyze videos using appropriate computer vision methods and techniques

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	1	1	1	-	-	-	-	1
CO2	2	2	2	1	-	-	-	1
CO3	2	2	2	2	-	-	-	2
CO4	2	2	2	2	-	-	-	2

Syllabus

Unit I

Introduction to computer vision. Image processing v/s computer vision. Applications of computer vision. Types of images: binary, greyscale, color image. Image channels, splitting and merging channels, and manipulating color pixels. Mathematical operations on images.

Unit II

Binary Image Processing: Thresholding, Erosion and Dilation, opening and closing. Connected component analysis, contour analysis. Color spaces: RGB, HSV, CMYK, Y'CbCr, Y'UV. Image filtering, smoothing and gradient

Unit III

Feature Extraction: Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT. Gabor filters, LBP, GLCM.

Unit IV

Video Analysis**:** Motion Estimation using Optical Flow, Video stabilization, object tracking, Kalman filter, MeanShift and CamShift**.** Background Subtraction and Modelling**.** Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Textbooks / References:

1. Computer Vision: Algorithms and Applications by Richard Szeliski
2. Computer Vision: A Modern Approach (Second Edition) by David Forsyth and Jean Ponce
3. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press

26CSA641	ETHICAL HACKING AND INFORMATION SECURITY	L-T-P-C: 3-1-0-4

Course Description

This course introduces the fundamental concepts of ethical hacking and information security, focusing on understanding cyber threats, vulnerabilities, and attack methodologies used by malicious actors. It provides learners with practical knowledge of ethical hacking techniques, security tools, legal and ethical frameworks, and defensive strategies used to protect information systems and networks. The course emphasizes responsible hacking practices, compliance with cyber laws, and the application of security controls to safeguard digital assets, preparing students for entry-level roles in cybersecurity and ethical hacking.

Course Objectives

- To understand the fundamentals of ethical hacking and information security
- To familiarize students with common cyber threats, attacks, and vulnerabilities
- To introduce ethical hacking tools, techniques, and methodologies
- To impart knowledge of legal, ethical, and professional standards in cybersecurity
- To develop skills in identifying, analyzing, and mitigating security risks
- To understand defensive strategies and security best practices for information systems

Course Outcomes

COs	Description
CO1	Analyze the ethical, legal, and architectural foundations of a secure penetration testing environment.
CO2	Apply scanning and system hacking methodologies to identify and exploit host vulnerabilities.
CO3	Examine network traffic and session management flaws to detect sniffing and hijacking attacks.
CO4	Evaluate the impact of social engineering and Denial of Service (DoS) attacks on organizational continuity.
CO5	Construct exploits for web applications and wireless networks using standard frameworks like Metasploit.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	-	-	-	3	-	-
CO2	2	3	-	-	3	-	-	-
CO3	-	3	-	2	3	-	-	-
CO4	-	2	3	-	-	2	-	-
CO5	-	-	3	2	3	-	1	-

Syllabus

Unit 1

Ethical Hacking Basics: Introduction to ethical hacking, information security overview, and the "Cyber Kill Chain". Legal & Ethical Frameworks: Information security laws, standards, and the impact of unethical hacking. Penetration Testing Environments: Architecture of a hacking lab; Virtualization, Sandboxing, and isolation of Attacker/Victim systems. Information Gathering: Footprinting methodologies (Active vs. Passive), Google Dorking, Whois lookup, and DNS interrogation.

Unit 2

Scanning & Enumeration: Port scanning, banner grabbing, and vulnerability scanning techniques. System Hacking: Password cracking (Dictionary, Rainbow tables), Privilege Escalation, application execution, and covering tracks. Malware: Concepts of Trojans, Viruses, Worms, Rootkits, and Ransomware analysis.

Unit 3
Packet Sniffing: Active vs. Passive sniffing, MAC Flooding, DHCP Starvation, and ARP Spoofing/Poisoning. Traffic Analysis: Inspecting protocols (HTTP, FTP, Telnet) for clear-text data using analyzers like Wireshark. Session Hijacking: Spoofing vs. Hijacking, TCP/IP Hijacking, and mitigation strategies for session management.

Unit 4
Social Engineering: Phishing, Spear Phishing, Vishing, Pretexting, and Insider Threats. Denial of Service (DoS): DoS vs. DDoS, Botnets, SYN Flooding, Smurf Attacks, and mitigation techniques. Perimeter Security: Introduction to Firewalls, IDS, and IPS evasion techniques.

Unit 5
Web Application Hacking: OWASP Top 10 overview, SQL Injection (Blind vs. Error-based), and Cross-Site Scripting (XSS). Exploitation Frameworks: Architecture of Metasploit, Payloads (Bind vs. Reverse Shell), and Encoders. Vulnerability Analysis: Analyzing web application vulnerabilities and operating system flaws.

Textbooks / References

1. Michael T. Simpson, Kent Backman, James E. Corley, *Hands-On Ethical Hacking and Network Defense*, 2nd Edition, CENGAGE Learning.
2. Jon Erickson, *Hacking: The Art of Exploitation*, 2nd Edition.
3. Alan T. Norman, *Computer Hacking Beginners Guide*.
4. Data and Computer Communications -- W. Stallings.

26CSA642	SYSTEM SECURITY	L-T-P-C: 3-0-2-4

Course Description

This course provides a comprehensive introduction to system security, focusing on protecting computer systems from a wide range of threats and attacks. It covers the fundamental principles of securing operating systems, hardware, and system-level software against vulnerabilities, malware, unauthorized access, and misuse. Students will gain practical insight into common system attacks and corresponding defense mechanisms, including access control, authentication, system hardening, monitoring, and incident response. The course equips learners with the knowledge required to design, implement, and manage secure computing environments in both standalone and networked systems.

Course Objectives

- To understand the fundamentals of computer system security and its importance
- To study common system-level attacks and vulnerabilities
- To learn and implement defense mechanisms against malware, unauthorized access, and other threats
- To gain practical knowledge of access control, authentication, and system hardening techniques
- To develop skills for monitoring, detecting, and responding to system security incidents
- To apply best practices for securing operating systems and system-level software

Course Outcomes

COs	Description
CO1	Describe security goals and principles which is used in designing a secure system.
CO2	Demonstrate the exploitation of Access control vulnerabilities and develop its mitigation.
CO3	Explain the basics of system organization, assembly language and Linux system calls.
CO4	Demonstrate buffer overflow attack, format string attack and return to libc attack with examples.
CO5	Understand the preventive mechanisms for different exploits.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	-	-	-	-	-
CO2	3	2	1	-	-	-	-	-
CO3	3	-	-	1	-	-	-	-
CO4	3	1	-	-	-	-	-	-
CO5	3	1	1	1	-	-	-	-

Syllabus

Unit I
Security Goals, Secure Design Principles

Unit II
Authentication, Linux Password scheme, Password Security, Authorization - Access control, MAC, DAC, ACL, Capabilities, Information flow control, Privilege Escalation Attacks, constraining and sandboxing users and applications.

Unit III
Assembly Primer, Shell coding, ELF File Format.

Unit IV
Memory Exploits – Buffer Overflow, off by one overflow, Format String Attacks, Integer Overflow, Return to Libc, Heap Overflow, Exploit prevention mechanisms: stack canaries

Unit V
Data Execution Prevention, Address Space Layout Randomization, bypassing DEP & ASLR. Trusted Execution Environment - Case Study on IntelSGX. Fuzzing - Types of fuzzers, Bug detection, Case study - AFL fuzzer. Vulnerability and exploit analysis: spectre, meltdown, foreshadow, dirty COW.

Textbooks / References:

1. Neil Daswani, Christopher Kern, Anita Kesavan, "Foundations of Security, What Every Programmer Needs to Know", Apress, 2007
2. Jon Ericson, "Hacking: The Art of Exploitation", Second Edition, No Starch Press, 2008
3. Gary McGraw, John Viega, "Building Secure Software", Addison-Wesley Professional, 2001.
4. Michael Sutton, Adam Greene, Pedram Amini, "Fuzzing Brute Force Vulnerability Discovery".

26CSA643	WEB APPLICATION SECURITY	L-T-P-C: 3-0-2-4

Course Objectives

Students will learn an overview of web application architectures and the associated security vulnerabilities and defenses. By the end of the course, students will be confident to understand how to secure web applications.

Course Outcomes

Cos	Description
CO1	Understand the security requirements in web applications.
CO2	Understand the various attack against web applications
CO3	Implement secure coding practices.
CO4	Get trained in responsible vulnerability disclosure.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	-	-	-	-	-	-	-
CO2	3	1	-	2	-	-	-	-
CO3	3	-	2	1	-	-	-	-
CO4	1	-	-	1	-	-	-	-

Syllabus

Unit I

Introduction - Overview of web architecture, Protocols, Client-server architecture, P2P architecture, DNS, etc.
Understanding the browser: Same origin policy, Cookies, Cache, authentication.
Website development basics, understanding server-side languages like HTML, PHP and Database languages such as SQL. Understanding the frontend, backend, database paradigm of web application development.

Unit II

Injection attacks: SQL injection, OS command injection.
File upload vulnerability: LFI, RFI, secure a file inclusion vulnerability.
Request forgery vulnerability: Server-side request forgery, Client-side request forgery.
Cross-site scripting attacks: Reflected XSS, Stored XSS, how to properly secure against XSS attacks.
DOS & DDOS attacks, Phishing attacks.
Automating vulnerabilities: SQL map, Burp Suite.

Unit III

OWASP Top 10: Broken Authentication, Sensitive Data Exposure, XML External Entities, Broken Access Control, Security Misconfiguration, Insecure Deserialization, Using Components with Known Vulnerabilities, Insufficient Logging & Monitoring.

Unit IV

Responsible vulnerability disclosure: CVE's, CVEmitre, Exploit-db, SearchSploit, bug bounty.
Secure coding practices: blacklisting, whitelisting, user input validation, automated testing, sanitizing HTML.

Textbooks / References:

1. Peter Yaworski, "Real-World Bug Hunting: A Field Guide to Web Hacking"
2. Michal Zalewski, "The Tangled Web: A Guide to Securing Modern Web Applications"
3. Dafydd Stuttard and Marcus Pinto, "The Web Application Hacker's Handbook" Second edition, 2011
4. OWASP, "Web Security Testing Guide", Fourth edition.

26CSA644	ADVANCED NETWORK SECURITY	L-T-P-C: 3-0-2-4

Course Description

This course provides a comprehensive study of network security principles, practices, and technologies, focusing on both defensive and offensive aspects of securing modern networks. It bridges the gap between theoretical networking concepts and real-world security operations. Students will explore contemporary network architectures, including SASE (Secure Access Service Edge) and SD-WAN (Software-Defined Wide Area Networks), and learn how to secure them against evolving threats. The course emphasizes hands-on experience in detecting, analyzing, and mitigating network attacks such as ARP poisoning, DDoS, and other intrusion attempts. By the end of the course, students will be equipped with the skills to design, monitor, and defend enterprise-grade networks effectively

Course Objectives

- To transition students from theoretical networking to applied defensive and offensive operations.
- To master modern network frameworks like SASE and SD-WAN which are dominant in current industry settings.
- To provide hands-on experience in detecting and mitigating real-world network attacks (ARP poisoning, DDoS).

Course Outcomes

COs	Description
CO1	Analyze modern network architectures (Cloud/SASE) to identify potential security gaps.
CO2	Execute vulnerability assessments using industry-standard tools and interpret the results.
CO3	Defend wireless and IoT infrastructures against specialized protocol attacks.
CO4	Configure and manage complex defense systems like NG-Firewalls and IDS/IPS.
CO5	Learn how to find weaknesses in a computer network and test them using various attack methods to see how well a company's defenses work

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	-	2	3	1	1	1	3	3
CO2	-	2	3	2	2	1	3	3
CO3	3	2	3	2	3		3	
CO4	3	3	3	3	3		3	
CO5	3	2	2	2	3		3	

Syllabus

Unit 1

Application Security & Traffic Analysis: Information Security Awareness: The evolution of Bad Actors (State-sponsored vs. Insiders); Deep dive into Web and HTTP security; Content Distribution Networks (CDNs); Web Application Firewalls (WAF) vs. traditional firewalls.

LAB: Web Header Analysis: Analyzing HTTP security headers and cookies. Socket Security: Implementing secure socket communication using Python/Java. Email Analysis: Email Header Analysis and investigating SMTP/IMAP vulnerabilities.

Unit 2
Infrastructure & Cloud-Edge Security - Evolution of Network Security; Secure Access Service Edge (SASE); Cloud Security models; SD-WAN (Software-Defined Wide Area Network); Endpoint Security and EDR (Endpoint Detection and Response).

Unit 3
Wireless & IoT Security - Wi-Fi Technologies: Evolution of Wi-Fi (NSE 2 standards); WPA3 vs. WPA2 security; IoT Protocols: Bluetooth and Zigbee architecture; Working mechanisms and specific security flaws in low-power protocols.

LAB: Traffic Analysis: Capturing and decrypting wireless handshakes. Exploitation Demo: Demonstration of Wi-Fi exploitation (De-authentication attacks) and Bluetooth pairing vulnerabilities.

Unit 4
Defensive Architecture (Firewalls, IDS/IPS) - Modern Firewall architectures (Next-Gen Firewalls); NSE 2 Firewall concepts; Role of Threat Intelligence Services; IDS and IPS: signature-based vs. anomaly-based detection; Placement and design of IDS/IPS in a network.

LAB: IDS Deployment: Setting up and tuning an IDS (e.g., Snort or Suricata) to reduce false positives. Firewall ules: Implementing egress and ingress filtering to stop data exfiltration.

Unit 5
Offensive Security & Vulnerability Management - Network Risk and Vulnerability Management; Types of Vulnerability Assessments (Internal vs. External); Network Attack Vectors: DNS Poisoning, ARP Poisoning, Replay attacks, and Privilege Escalation; Malware & DDoS mechanics.

LAB: Information Extraction: Advanced reconnaissance using NMAP and script scanning (NSE). Attack Simulation: Demonstration of MAC Spoofing, ARP Poisoning, and Switch Port Stealing. Vulnerability Scanning: Using professional tools (e.g., Nessus or OpenVAS) to generate a risk report.

Text/References:

1. Michael Goodrich, Roberto Tamassia, Introduction to Computer Security: Pearson publications, 2nd edition, 2021, ISBN-13: 978-0133575477.
2. William Stallings, *Computer Networking with Internet Protocols and Technology*, Pearson Education, 2011.
3. Behrouz A. Forouzan, *Data Communications and Networking*, McGraw-Hill, 2017.
4. Harlie Kaufman, Radia Perlman, and Mike Speciner, *Network Security: Private Communication in a Public World*, Prentice Hall, 2002
5. Sagar Rahalkar, Quick Start Guide to Penetration Testing: With NMAP, OpenVAS and Metasploit, 1st Edition, Apress publications, 2019, Softcover ISBN: 978-1-4842-4269-8.

26CSA645	VULNERABILITY AND PENETRATION TESTING	L-T-P-C: 3-0-2-4

Course Description

This course introduces students to the principles and practices of Vulnerability Assessment and Penetration Testing (VAPT). It focuses on identifying, analyzing, and exploiting security weaknesses in systems, networks, and applications to assess their resilience against cyber threats. Students will learn industry-standard tools and methodologies for vulnerability scanning, penetration testing, and risk assessment. The course emphasizes practical skills, including report generation, communicating findings, and recommending remediation steps, enabling students to confidently perform security assessments for organizations or product teams.

Course Objectives

The student will be confident to perform vulnerability and penetration testing for any organization or product team, generate a report and communicate remediation steps.

Course Outcomes

COs	Description
CO1	Gain the skills to analyze and identify weaknesses in network systems and web services.
CO2	Understand the importance of strong passwords and be able to ethically penetrate systems by leveraging weak password practices.
CO3	Use various tools to analyze network traffic and packets for security purpose
CO4	Perform different types of injection attacks, understanding their impact and potential for exploitation.
CO5	Develop the ability to document their penetration testing activities by creating comprehensive reports that detail vulnerabilities discovered and recommendations for remediation.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	-	-	-	-	-	-	-
CO2	3	1	-	-	-	-	-	-
CO3	3	-	-	3	-	-	-	-
CO4	3	-	-	1	-	-	-	-
CO5	1	-	1	1	-	-	-	-

Syllabus

Unit I

Active information gathering, General vulnerability scanners, Port-based, Service-based, Banner grabbing, Web application scanners, General application flaw scanners, Directory listing/Brute forcing, Web server version/Vulnerability identification, Technology stacks and packages version detection, Network vulnerability scanners/Specific protocols, VPN, Manual direct connections.

Unit II

Passive information gathering, Metadata analysis, Traffic monitoring, ARP/MAC cache overflow, Etherleak, Misconfigured clusters or load balancers

Public Research: Vulnerability databases, Vendor advisories, Exploit databases and framework modules, Common/default passwords, Hardening guides/common misconfigurations, Private Research, Fuzzing. Wi-Fi password hacking, aircracking,

Unit III

DNS Cache poisoning/Spoofing, Recon tools: NS enumeration tools, Nmap, Netcat, Tcpdump, Wireshark, Directory enumeration tools, Google hacking, Shodan.

Unit IV

SQL injection Case study: SQLi, File upload vulnerabilities, Case study: SSRF, Reverse shell, Password brute-forcing using shadow file, Hashcat, John the ripper, Hydra, Medusa, Ncrack, Cross site scripting, XSS, Client-side request forgery, IDOR, Metasploit.

Unit V

XML attacks, Case study: XXE, Vulnerability Exploitation and Generating PoC, Vulnerability assessment, Pen Test Report Generation.

Textbooks / References:

1. OWASP Web Security Testing Guide V4
2. Bugcrowd, “The Ultimate Guide to Penetration Testing”, 2020 edition
3. HackerOne, “Web hacking 101”

26CSA646	COMPUTER NETWORKS FOR CYBER SECURITY	L-T-P-C: 3-0-2-4

Course Description

This course provides a strong foundation in computer networking concepts and focuses on securing modern networked systems. It introduces students to network architectures, protocols, vulnerabilities, attacks, and defense mechanisms relevant to today’s cyber security landscape, including cloud, wireless, and IoT networks.

Course Objectives – Network Security Fundamentals

- To understand the fundamental concepts of computer networking and network architectures
- To study common network vulnerabilities, threats, and attack vectors
- To learn defense mechanisms and security controls for wired, wireless, cloud, and IoT networks
- To gain knowledge of protocols, network monitoring, and intrusion detection techniques
- To develop practical skills in securing networked systems and analyzing security incidents
- To prepare students for further studies or careers in cybersecurity and network defense

Course Outcomes

CO	Course Outcome
CO1	Explain fundamental computer networking concepts, architectures, and protocols with emphasis on their role and relevance in cyber security.
CO2	Analyse data link and network layer functionalities to identify vulnerabilities, attacks, and secure routing mechanisms in networked systems.
CO3	Evaluate transport layer protocols and mechanisms to understand performance, reliability, and security issues in network communication.
CO4	Analyse application layer protocols and security mechanisms to protect network services such as DNS, web, and email against cyber attacks.
CO5	Examine emerging and modern network technologies such as IoT, SDN, and NFV, and assess their security challenges.

CO-PO Mapping

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	–	1	–	–	–
CO2	2	3	2	2	2	–	–	–
CO3	2	2	2	–	1	–	–	–
CO4	2	3	3	2	2	1	–	–
CO5	2	2	3	2	3	1	1	–

Syllabus

Unit 1

Computer Network Fundamentals - Network concepts and uses, packet switching vs circuit switching, Internet architecture, OSI and TCP/IP models, network devices and their security roles.

Packet Sniffing: Capturing live traffic using Wireshark to identify OSI layer headers. Network Topology Discovery using ipconfig/ifconfig and traceroute to map local network paths.

Unit 2

Data Link and Network Layer Functionalities - Error detection and correction, framing and flow control, IP addressing and subnetting, CIDR, ARP, ICMP, routing algorithms, IP spoofing, routing attacks, and secure routing concepts.
Subnetting and IP Planning—Designing a secure network addressing scheme. ARP Spoofing Simulation—Demonstrating a Man-in-the-Middle (MitM) attack in a controlled environment. Network Reconnaissance—Using Nmap to identify active hosts and potential vulnerabilities.

Unit 3

Transport Layer - UDP, TCP, connection management, flow control, congestion control, Introduction to quality of service, TCP vulnerabilities.
Transport Layer Analysis — Analysing the TCP 3-way handshake and termination process.

Unit 4

Application Layer - DNS, Web, HTTP, email, DNS architecture and attacks, DNSSEC, HTTP vs HTTPS, SSL/TLS, email security, authentication and encryption mechanisms. Capture packets and analyze the protocols, query types, dig, nslookup,

Unit 5

Modern Network Security Paradigms - Internet of Things (IoT) networks – architecture, communication protocols (6LoWPAN, RPL, CoAP, MQTT), security challenges. Software Defined Networking (SDN) and Network Function Virtualization (NFV).

Textbooks / Reference

1. William Stallings, "Computer Networking with Internet Protocols and Technology", Pearson.
2. James F. Kurose and Keith W. Ross, *Computer Networking: A Top-Down Approach*, Pearson Education, Latest Edition.
3. William Stallings, "Network Security Essentials", Pearson.
4. Behrouz A. Forouzan, "Data Communications and Networking", McGraw-Hill.
5. Olivier Hersent, David Boswarthick, Omar Elloumi , The Internet of Things: Key Applications and Protocols, 2nd Edition, Wiley

REGULAR STREAM

26CSA651	COMPLEX NETWORK ANALYSIS	L-T-P-C: 3-0-2-4

Course Description

Network science is an evolving field which focuses on the study of patterns of connection in a wide range of physical and social phenomena. The exponential increase in data sets derived from social, economic, and biological networks, along with modern computational power, has increased its relevance. The goal of this course is to provide a mathematical foundation for understanding and analyzing the structure of complex networks. The subject material is interdisciplinary, with topics of graph theory, probability theory, statistical physics, and computer science.

Course Objectives

- To understand and explain the workings of systems built upon complex networks
- To impart fundamental and advanced concepts in the areas of complex networks and network science that focus on study of the models and behavior of networked systems.

Course Outcomes

COs	Description
CO1	Describe the fundamental concepts of graph theory and network mathematics along with properties.
CO2	Use various measures and metrics for analyzing networks.
CO3	Implement the concept of large-scale networks, communities and community detection algorithms in various applications.
CO4	Differentiate random graphs and models of network growth.
CO5	Explore and describe the processes taking place in Networks.

CO-PO Mapping

PO/ PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	2	2	-	-	-	-	1
CO2	3	2	2	-	-	-	-	1
CO3	3	2	2	1	1	-	-	1
CO4	2	2	1	-	-	-	-	1
CO5	2	2	2	-	-	-	-	1

Prerequisites

- Proficiency in programming languages
- Basic knowledge in graph theory

Syllabus

Unit I

Graphs and Networks- Review of basic graph theory, Examples of real-world networks, networks and their representation, the adjacency matrix, weighted networks, directed networks, hypergraphs, bipartite networks, trees, planar networks, degree, paths, components, independent paths, connectivity and cut sets, the graph Laplacian, random walks, Properties of Networks.

Unit II

Measures and Metrics: Degree centrality, eigenvector centrality, Katz centrality, page rank, hubs and authorities, closeness centrality, betweenness centrality, groups of vertices, transitivity, reciprocity, signed edges and structural balance, similarity, homophily and assortative mixing.

Unit III

The large-scale structure of Networks, Basic concepts of network communities, community structures, network navigation, Modularity, Girvan-Newman Algorithm, Spectral Bisection Algorithm, Radicchi Edge Clustering Algorithm, Wu-Hubermann Algorithm, Random Walk based Algorithm.

Unit IV

Generalized random graphs, Poisson random graphs- the configuration model, generating functions, power-law degree distribution, Models of Network Growth-Price model, Barabasi & Albert model, other growth models, vertex copying models, Bipartite Network.

Unit V

Processes on Networks: Percolation theory and network resilience, Epidemiological processes, Cascades and information spread, Cohesiveness, Cliques, Clans, Clubs, Plex, Equivalence of ties, Ego-centric networks, Cascade formation and information diffusion in social media. Search on networks, exhaustive network search, guided network search, network navigation; network visualization and semantic zooming, Temporal network, Multilayer networks, Interdependent networks, Controllability of complex networks, Economic and financial network analytics.

The lab experiments/ case studies shall be implemented using any suitable tool such as Python/ R/ MATLAB.

Textbooks / References:

1. M.E.J. Newman, "Networks: An Introduction", Oxford University Press, 2010
2. The structure and function of complex networks. https://epubs.siam.org/doi/10.1137/S003614450342480
3. Statistical mechanics of complex networks, Rev. Mod. Phys., 74(1), 2002.
4. Complex Graphs and Networks, by F. Chung and L. Lu
5. Dougles West, "Introduction to Graph Theory", Second Edition, PHI Learning Private Limited, 2011.
6. Guido Caldarelli, "Scale-Free Networks", Oxford University Press, 2007.
7. Alain Barrat, Marc Barthelemy and Alessandro Vespignani, "Dynamical processes on Complex networks", Cambridge University Press, 2008.
8. Reuven Cohen and Shlomo Havlin, "Complex Networks: Structure, Robustness and Function", Cambridge University Press, 2010.

26CSA652 CONNECTED INTERNET OF THINGS DEVICES L-T-P-C: 3-0-2-4

Course Description

Internet of Things (IoT) aims to connect Everything to the Internet. Sensors, vehicles, home appliances, and other items fall into this category. This course was designed to prepare students for developing, deploying, and managing IoT products and services. Building IoT products necessitates the integration of multiple platforms, often using a variety of technologies and programming languages. Using such well-developed and well-supported technologies makes it easier to deploy the IoT network in a cost-effective and secure manner.

Course Objectives

To learn to specify, design and program modern connected electronic systems based on commodity smartphones and sensor networks.

Course Outcomes

COs	Description
CO1	Identify the elements of IoT system
CO2	Describe IoT Protocols
CO3	Demonstrate Internet of Things architecture
CO4	Recognize the concepts of Web of Things
CO5	Identify the application areas where Internet of Things can be applied

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	2	2	-	-	-	1	-
CO2	2	3	2	-	-	-		-
CO3	3	1	2	-	-	-		-
CO4	2	1		-	1	-	1	-
CO5	2	3	2					

Syllabus

Unit I

IOT - What is the IoT and why is it important? Elements of an IoT ecosystem, Technology drivers, Business drivers, Trends and implications, Overview of Governance, Privacy and Security Issues.

Unit II

IOT PROTOCOLS - Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFID Protocols – Issues with IoT Standardization – Unified Data Standards – Protocols – IEEE802.15.4– BACNet Protocol– Mod bus – KNX – Zigbee– Network layer – APS layer – Security.

Unit III

IOT ARCHITECTURE - IoT Open source architecture (OIC)- OIC Architecture & Design principles- IoT Devices and deployment models- IoTivity : An Open source IoT stack - Overview- IoTivity stack architecture- Resource model and Abstraction.

Unit IV

WEB OF THINGS - Web of Things versus Internet of Things – Two Pillars of the Web – Architecture Standardization for WoT– Platform Middleware for WoT – Unified Multitier WoT Architecture – WoT Portals and Business Intelligence.

Unit V

IOT APPLICATIONS - IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications. Study of existing IoT platforms /middleware, IoT- A, Hydra etc.

Textbooks / References:

1. Honbo Zhou, “The Internet of Things in the Cloud: A Middleware Perspective”, CRC Press, 2012.
2. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), “Architecting the Internet of Things”, Springer, 2011.
3. David Easley and Jon Kleinberg, “Networks, Crowds, and Markets: Reasoning About a Highly Connected World”, Cambridge University Press, 2010.
4. Olivier Hersent, David Boswarthick, Omar Elloumi , “The Internet of Things – Key applications and Protocols”, Wiley, 2012.
5. Vijay Madisetti and ArshdeepBahga, “Internet of Things (A Hands-on-Approach)”,1st Edition, VPT, 2014
6. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013
7. CunoPfister, Getting Started with the Internet of Things, O‟Reilly Media, 2011, ISBN: 978-1- 4493-9357-1.

26CSA653	**IoT AND CLOUD COMPUTING**	**L-T-P-C: 3-0-2-4**

Course Description

Cloud Computing and IoT are one of the most trending technologies in today’s world. Although these are two different technologies, it is very interesting to learn about the interdependence of cloud computing and IoT. In the near future almost all devices and appliances will include IoT modules which will use sensor data collection and control/management based on Clouds.

Course Objectives

- This course provides an overview of the Internet of Things (IoT) and Cloud Computing concepts, infrastructures and capabilities.
- This will help students gain the necessary knowledge to construct IoT systems and use cloud services for processing and storage of the data produced by the IoT devices.

Course Outcomes

COs	**Description**
CO1	Describe the IoT characteristics, applications and challenges.
CO2	Describe general concepts of IoT and recognize different devices.
CO3	Determine the proper sensors and communication protocols to use in a particular IoT system.
CO4	Identify the architecture and infrastructure of cloud computing and resource management fundamentals.
CO5	Develop IoT applications and implement different solution approaches in Cloud and evaluate the security issues.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	-	-	-	-	-	-	-
CO2	-	-	3	-	-	-	-	-
CO3	-	3	-	-	-	-	-	1
CO4	2	-	-	-	-	-	-	1
CO5	-	-	2	1	-	-	-	1

Syllabus

Unit I

Introduction to IoT: Definition and Characteristics of IoT, IoT services and applications, IoT in Indian scenario, Challenges in IoT implementation.

Unit II

IoT architecture and design: IoT Layers and components, IoT device platforms (Arduino, Raspberry Pi, ESP8266).

Unit III

IoT Network and communication protocols – Networking architectures, Networking protocols (TCP/IP, 6LoWPAN, RPL, Thread), IoT Devices Application-Level Protocols (MQTT, CoAP, REST).

Unit IV

Cloud Computing Fundamentals, Cloud Computing Architectures, Cloud Types and Services, Virtualization and Resource Management.

Unit V

Application of IoT & Cloud

IoT and cloud integration, Application development and cloud processing, Security and Privacy for IoT/Cloud Computing.

Textbooks / References:

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things”, 1st Edition, Pearson Education (Cisco Press Indian Reprint). (ISBN: 978-9386873743)
2. Andy King, “Programming the Internet of Things: An Introduction to Building Integrated, Device-to-Cloud IoT Solutions”, 1st Edition, O'Reilly
3. Vibha Soni, “IoT for Beginners: Explore IoT Architecture, Working Principles, IoT Devices, and Various Real IoT Projects (English Edition)”, 1st Edition, BPB Publications
4. Srinivasa K G, “Internet of Things”, CENGAGE Leaning India, 2017

26CSA654 COMPUTER GRAPHICS AND VISUALIZATION L-T-P-C: 3-0-2-4

Course Description

Computer graphics and visualisation applications range from art and entertainment to science and engineering, biology, and medicine. This course prepares students for work in complex modelling and application development fields. It covers the fundamental principles, concepts, and algorithms of computer graphics and visualisation. Students will learn mathematical and computational techniques for modelling, representing, and displaying geometric objects, as well as how to use these techniques for data visualisation.

Course Objectives

To introduce the use of a graphics system and become familiar with building approach of graphics system components and algorithms related with them. Also covers the basic principles of 2D and 3D graphics. And an understanding of how to scan convert the basic geometrical primitives, how to transform the shapes to fit them as per the picture definition.

Course Outcomes

COs	Description
CO1	Describe various types of video displays and colour models.
CO2	Implement various drawing primitives using OPENGL.
CO3	Explain fundamental concepts within computer graphics such as geometrical transformations.
CO4	Describe various 3D object representation techniques.
CO5	Describe the concept of illumination models, removal of hidden surfaces and rendering.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	2	2	-	-	-	2	-
CO2	2	2	2	-	-	-	2	-
CO3	3	2	2	-	-	-	2	-
CO4	3	2	2	-	-	-	2	-
CO5	2	2	2	-	-	-	2	-

Syllabus

Unit I

Computer Graphics Fundamentals: Overview of CG - Video Displays -Color Models

Unit II

Output Primitives. Introduction to OPENGL- Points, Lines – Specifying a 2D World Coordinate Reference Frame in OpenGL- OpenGL Point Functions, Line Functions Polygon Fill Area Functions, Vertex Arrays - Line Drawing Algorithms - Circle Generation Algorithm Filled Area Primitives OpenGL fill Area Functions - Scan Line Polygon Filling Algorithms – Boundary Fill - Flood Fill Algorithms
Attributes of Output Primitives.

Unit III

Geometric Transformations: Basic 2Dtransformations-Other Transformations- Reflection and Shearing. OpenGL Geometric Transformation Functions.

Unit IV
3D Object Representation: Fractals - Geometrical Transformation for - 3D Objects – Viewing and Clipping 2D Viewing Functions Clipping Operations. Three-Dimensional Viewing: Viewing Pipeline, Viewing Coordinates. Projections: Parallel Projections, Perspective Projections. OpenGL Two-Dimensional and Three-Dimensional Viewing Functions-OpenGL Animation.

Unit V
Visible Surface Detection and Illumination Models: Visible Surface Detection Methods –Illumination Methods and Surface Rendering – Polygon. Rendering Methods: Constant Intensity Shading, Gouraud Shading, Phong Shading. OpenGL Illumination and Surface Rendering Functions, GUI – OpenGL Interactive Input Device Functions. The User Dialog –Interactive Picture Construction Techniques – Color Models - Computer Animation.

Textbooks / References:

1. Donald Hearn and Pauline Baker, ―Computer Graphics with OpenGL ‖, Third Edition, Prentice Hall of India, 2009.
2. Roy A. Plastock and Gordon Kalley, ―Schaum's Outline Series - Theory and Problems ofComputer Graphics‖, Second Edition, Tata McGraw-Hill, 2000.
3. Foley J.D, Van Dam A, Eiener S.K. and Hughes J.F., ―Computer Graphics Principles andPractice‖, Second Edition, Pearson Education, 1996.
4. Rajiv Chopra ―Computer Graphics – A Practical Approach‖.

26CSA655	**DevOps**	**L-T-P-C: 3-0-2-4**

Course Description

This course deals with as a professional, you will be able to transform areas of code automation, configuration management, version controlling and monitoring of different applications. On successful completion of the course, a candidate is entitled to earn a certificate of achievement, which is proof of the merit of the candidate.

Course Objectives

- Understand the concepts of DevOps and the issues it resolves.
- Implement Automated Installations.
- Learn to Develop automation using Maven.
- Understand Continuous Delivery & Continuous Deployment

Course Outcomes

COs	**Description**
CO1	Describe traditional software development vs DevOps.
CO2	Recognize deployment and flow management in DevOps
CO3	Illustrate the Measurement, collaboration and visualizing using DevOps.
CO4	Demonstrate the Common DevOps Roles.
CO5	Define common DevOps practices and techniques

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	3	1	-	-	-	-	-	-
CO2	1	-	3	-	-	-	-	-
CO3	3	-	-	-	-	-	-	-
CO4	-	2	-	-	2	-	-	-
CO5	1	-	3	-	1	-	-	-

Syllabus

Unit I

Introducing DevOps, Emergence of DevOps, History of DevOps, Transformation with DevOps and Agile, Business Case for DevOps, Benefits of DevOps, Agile Practices, Focus on Products and Service, Autonomy of Teams, Introducing CALMS, Culture, Team Behaviors, Team Agility, Cross-functional Delivery Teams, Job Satisfaction, Servant Leadership.

Unit II

Automation, Continuous Integration, Environment Management, Release Management , Test Automation, Deployment, Data and Data Management. . Lean, Flow Optimization, Work In Progress (WIP) , Constraint Management, Reducing Waste, Customer Focus.

Unit III

Measurement, Aligning Goals, Delivery Metrics, Operational Metrics, Metric Analysis, Lead and Cycle Time, Sharing, Collaboration, Feedback Loops, Visualizing, Business and IT Work Alignment, Education and

Learning.

Unit IV
Common DevOps Roles, DevOps Evangelist, Automation Architect, Cloud Infrastructure Engineer, Software Developer, Software Test, Security Engineer, Database Administrator, Product Owner.

Unit V
Common DevOps Practices and Techniques, Continuous Integration, Testing and Deployment Infrastructure as Code, Test-Driven Deployment, Integrated Toolchains, Distributed Version Control, Production Monitoring, Public, Private and Hybrid Cloud Technologies
Relevant Methods and Approaches for DevOps Teams, DevOps Topologies and Target Operating Models, Scrum Development Delivery, Kanban Workflow, Transformational Leadership, Full-Stack Engineering, Collective Ownership, Continuous Experimentation.

Textbooks / References:

1. The DevOps Handbook - Book by Gene Kim, Jez Humble, Patrick Debois, and Willis.
2. What is DevOps? - by Mike Loukides.

26CSA656	DIGITAL IMAGE PROCESSING	L-T-P-C: 3-0-2-4

Course Description

Image processing deals with methods to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. This course provides basic knowledge about digital images, Imaging geometry, Image transforms, Image enhancement and filtering, Image restoration, Image segmentation, and morphological operations which are useful in any computer vision applications. Image Compression – need for image compression, Huffman, run-length encoding, shift codes, Vector quantization, Transform coding, JPEG standard, MPEG.

Course Objectives

- To introduce students to the basics of digital image processing applicable to binary, gray scale and colour images.
- To familiarize students to various algorithms in spatial and frequency domain relevant to image enhancement and restoration.
- To provide an opportunity to learn image compression and segmentation and its applications.

Course Outcomes

COs	Description
CO1	Describe the fundamental concepts of digital image processing and perform basic operations on pixels.
CO2	Implement image transformation and image enhancement techniques in spatial and frequency domain to devise algorithms or mathematical models for real time image enhancement problems.
CO3	Implement various techniques used for image restoration.
CO4	Use morphological processing on images for simple image processing applications.
CO5	Implement segmentation and compression algorithms on Images and analyze their performance.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	1	-	-	-	-	-	-
CO2	3	2	2	1	-	-	-	-
CO3	2	2	2	1	-	-	-	-
CO4	2	2	1	1	1	-	-	-
CO5	2	2	2	1	1	-	-	-

Syllabus

Unit I

Digital Image Fundamentals: Elements of Visual Perception- Simple Image Formation Model -Image Sensing and Acquisition-Image Sampling and Quantization – Basic Relationships between Pixels - Image interpolation.

Unit II

Intensity Transformations and Filtering: Basic Intensity transformation Functions – Histogram Processing – Fundamentals of Spatial Filtering –Smoothing and Sharpening Spatial Filters. Filtering in Frequency Domain: 2D Discrete Fourier Transforms - Basics of filtering - Image Smoothing and Image Sharpening Using Frequency Domain Filters - Selective Filtering.

Unit III

Image Restoration: Noise Models – Restoration using Spatial Filters – Periodic Noise Reduction by Frequency Domain Filters.

Unit IV

Morphological Image Processing: Erosion – Dilation – Opening – Closing – Hit-or-Miss Transform - Extraction of Connected Components.

Unit V

Image Segmentation: Fundamentals – Point, Line and Edge Detection – Thresholding-Region Based Segmentation – Region Growing – Region Splitting and Merging.

Introduction to Color image processing.

Image Compression – need for image compression, Huffman, Run-length encoding, shift codes, Vector quantization, Transform coding, JPEG standard, MPEG.

The lab experiments/ Case studies shall be done using MATLAB/ Python.

Textbooks / References:

1. Rafael C. Gonzalez and Richard E. Woods," Digital Image Processing", 4th Edition, Pearson, 2018.
2. A K. Jain, Fundamentals of digital image processing, Prentice Hall of India, 1989.
3. Al Bovik, The Essential Guide to Image Processing, Academic Press, 2009.
4. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis, and Machine Vision, Thomson Learning, 2008.
5. S Jayaraman, S Esakkirajan and T Veerakumar, Digital Image Processing, McGraw Hill Education, 2009.
6. Arthur R. Weeks, Jr., "Fundamentals of Electronic Image Processing", First Edition, PHI,1996.

26CSA657 ADVANCED COMPUTER NETWORKS L-T-P-C: 3-0-2-4

Course Description

The field of computer networking is rapidly changing. It is critical to consider not only what computer networks are today, but also why and how they are designed in the manner that they are. The goal of this course is to provide a solid conceptual foundation for computer networks and the principles that govern their design. The course covers the various protocols and the working of the Internet and its design which help students to contribute to research work.

Course Objectives

- The primary course objective is to provide the foundation of basics in computer networks in the digital era.
- Enable the student to understand the fundamental networking principles, standards, protocols and technologies.
- The course also provides insights into networking concepts in each layer of the protocol model.
- The course will enrich the students with hands-on experience in configuring networking devices using Packet Tracer and analyzing the protocols using Wireshark.

Course Outcomes

COs	Description
CO1	Discuss concepts of the core network and layered approach.
CO2	Describe routed network and design Network Models using Simulation tools.
CO3	Describe application protocols and its analysis using simulation tools
CO4	Describe IP Addressing and subnetting.
CO5	Demonstrate a real time network and study and network troubleshooting commands

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	1	1	-	-	-	-	-
CO2	2	2		3	-	-	1	-
CO3	1	1		2	-	-		-
CO4	2	2		2	-	-	-	1
CO5	3	2		3	-	-	1	

Syllabus

Unit I

Protocol layers -The Network Edge- The Network Core– Delay– Loss and Throughput in Packet Switched Networks

Unit II

IPV4, IPV6, Routing algorithm – Interior and Exterior routing. ICMP, Classless and Subnet Address Extensions (CIDR), Internet Multicasting. NAT Routing protocol design and architectures for RIP, OSPF, BGP, RIP.

Unit III

Application layer protocols – HTTP- DNS – PPP file sharing Introduction to Transport Layer Services - Connectionless Transport- UDP - Principles of Reliable Data Transfer- Connection Oriented Transport- TCP Traffic Control: Packet Scheduling, TCP Congestion Control, - Leaky Bucket, Token Bucket-Internet protocol

Unit IV

Internet Layer-Class full Addressing – Classless addressing – Private Addresses – Subnets – Subnet masks –ARP – ICMP-Routing & Forwarding -

Unit V
Global Internet– RIP – OSPF – BGP – Broadcast & Multicast routing.
ifconfig, nw.js - netcat - netstat - DNS - dhcp and monitoring tool Wireshark Network simulator.

Textbooks / References:

1. James F. Kurose and Keith W. Ross, “Computer Networking: A Top-Down Approach”, 6th Edition, Addison Wesley, 2008.
2. Larry Peterson and Bruce Davie, “Computer Networks: A Systems Approach”, Fourth Edition, Morgan Kaufmann, 2007.
3. Richard Stevens, Bill Fenner and Andrew M. Rudoff, “UNIX Network Programming”, Volume 1: “The Sockets Networking API”, Third Edition, Addison Wesley, 2004.
4. Andrew S.Tanenbaum, “Computer Networks”, Fifth Edition, Prentice Hall of India, 2011.

26CSA658	ADVANCED WEB TECHNOLOGIES AND MEAN STACK	L-T-P-C: 3-0-2-4

Course Description

This course deals with as a professional, you will be able to transform areas of code automation, configuration management, version controlling and monitoring of different applications. On successful completion of the course, a candidate is entitled to earn a certificate of achievement, which is proof of the merit of the candidate.

Course Objectives

This course helps the students to proficient in Javascript and use HTML, CSS and Javascript to handle front-end operations and back-end server scripting. MEAN is a full-stack development toolkit used to develop a fast and robust web application.

Course Outcomes

After completing this course, students will be able to:

COs	Description
CO1	Describe latest web application development trends in the IT industry
CO2	Get equipped with principles, knowledge, and skills for the design and construction of web-enabled internet applications
CO3	Design, Implement and deploy an in-house project using MongoDB, Express.js, Angular, and Node.js
CO4	Recognize and explore the REST architecture
CO5	Demonstrate the use of MongoDB and CRUD operations

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	3	3	-	-	-	-	2
CO2	2	3	3	-	-	-	-	2
CO3	2	3	3	-	-	-	-	2
CO4	2	3	3	-	-	-	-	2
CO5	2	3	3	-	-	-	-	2

Prerequisites

- Basic Web design concept
- Programming knowledge HTML, XML, CSS, Javascript
- Concept of database

Syllabus

Unit I

Basics of HTML, CSS, and JavaScript HTML, CSS, Bootstrap, JavaScript basics – Variables, functions, and scopes, Logic flow and loops, Events, and Document object model, Handling JSON data, Understanding JSON callbacks.

Unit II

Building Single Page Applications with Angular Single Page Application – Introduction, Two-way data binding (Dependency Injection), MVC in Angular, Controllers, getting user input, Loops, Client-side routing – Accessing URL data, Various ways to provide data in Angular – Services and Factories, Working with filters, Directives and

Cookies.

Unit III
Introduction to Node JS Installation, Callbacks, installing dependencies with npm, Concurrency and event loop fundamentals, Node JS callbacks, Building HTTP server, Importing and exporting modules

Unit IV
Building REST services using Node JS REST services, Installing Express JS, Express Node project structure, Building REST services with Express framework, Routes, filters, template engines - Jade, ejs.

Unit V
MongoDB Basics and Communication with Node JS Installation, CRUD operations, Sorting, Projection, Aggregation framework, MongoDB indexes, connecting to MongoDB with Node JS, Introduction to Mongoose, connecting to MongoDB using mongoose, Defining mongoose schemas, CRUD operations using mongoose.

Textbooks / References:

1. Simon Holmes, "Getting MEAN with Mongo, Express, Angular, and Node, Second Edition, Manning Publications; 1 edition (31 October 2015)
2. Jeff Dickey, "Write Modern Web Apps with Mean Stack, Peachpit press, 2015
3. Angular: From Theory to Practice by Asim Hussain, CodeCraft 1st edition
4. Beginning Angular with Typescript, Greg Lim
5. Mithun Satheesh, "Web development with MongoDB and Node JS", Packt Publishing Limited; 2nd Revised edition (30 October 2015).

26CSA659 MOBILE APPLICATION DEVELOPMENT L-T-P-C: 3-0-2-4

Course Description

Due to the growing popularity of technology and the digitization of everything, the Mobile App Development course is gaining popularity. It assists aspirants in learning how to develop smartphones and other mobile devices. The course teaches students how to create applications.

Course Objective

This Course provides a comprehensive overview of how to integrate mobile technology. Students learn how to create applications for mobile devices such as smartphones and tablets. They are introduced to current mobile operating systems and mobile application development environments. They will be able to create mobile applications with more than one user interface and more than one system component.

Course Outcomes

COs	Description
CO1	Describe characteristics of mobile communication and different application development environments
CO2	Demonstrate Android application development environment.
CO3	Implement user interfaces for interacting with apps and triggering actions for App development.
CO4	Implement mobile apps to solve real world problems using maps and google APIs
CO5	Demonstrate performance of android applications and role of permissions and security

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	3		-		-	-	2
CO2	2	3		1	-	-	-	2
CO3	2	3		2	1	-	-	
CO4	2	3		2		-	-	
CO5	2	3		-		-	-	

Syllabus

Unit I

Introduction-Mobile vs. Desktop devices -App Store, Google Play, Windows Store –Development Environments-Phone GAP- Native vs. web applications – Mobile Connectivity Evolution. Characteristics and advantages of mobile communication, types of mobile applications – development approaches, overview of mobile strategy and designing mobile solutions.

Unit II

Introduction to the Android Platform(Android Studio), Android Platform and Development Environment, Application Fundamentals, The Activity Class. Get started, Build your first app, Install Tools, Create HelloWorld App, Activities, Testing, debugging and using support Libraries.

Unit III

User Interaction Application Development, Testing UI, Background Tasks, Triggering, scheduling and optimizing tasks.

Unit IV

Data Storage and accessing the mobile data with different databases, Preferences and Settings, storing data using SQLite, sharing data with content providers, loading data using Loaders. Google APIs for Android - Maps, Cloud Messaging, Authentication, Storage, Hosting and Google Play services.

Unit V

Different level of security in mobile application, Solution of attacks, malware, permission, Firebase and Recent Trends.

Textbooks / References:

1. Google Developer Training, "Android Developer Fundamentals Course – Concept Reference", Google Developer Training Team, 2017.

2. Brian Fling, "Mobile Design and Development" O'Reilly Media,2009.

3. Maximiliano Firtman "Programming the Mobile Web", O'Reilly Media, 2010.

4. Erik Hellman, "Android Programming – Pushing the Limits", 1st Edition, Wiley India Pvt Ltd, 2014.

5. Valentino Lee, Heather Schneider, and Robbie Schell, "Mobile Applications: Architecture, Design and Development", Prentice Hall , 2004.

26CSA660 MULTIVARIATE STATISTICS L-T-P-C: 3-1-0-4

Course Description

The course will provide basic understanding of the multivariate statistics used in many areas. The course introduces some very powerful statistics behind solving real world problems from data reduction to forecasting.

Course Objectives

- To understand the concept of multivariate distributions
- To understand the computations of multivariate calculus.
- To explore the use of multivariate calculus in real life.

Course Outcomes

COs	Description
CO1	Describe basics of probability, random variables and distribution functions.
CO2	Discuss standard distributions and their properties.
CO3	Describe basics of multivariate distributions.
CO4	Explain PCA and its application on clustering.
CO5	Describe simple linear regression and its estimation.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	1	1	-	1	-	-	-	-
CO2	2	2	2	-	-	-	-	-
CO3	2	3	3	-	-	-	-	-
CO4	2	3	3	-	-	-	-	-
CO5	2	3	3	-	-	-	-	-

Syllabus

Unit I

Review of probability concepts-Conditional probability –Bayes Theorem, Introduction to Random variables: Discrete and Continuous random variables and its distribution –mathematical expectations.

Unit II

Some standard distributions –Binomial, Multinomial, Poisson, Uniform, exponential, Weibull, Gamma, Beta, Normal, Mean, variance, properties and application of these distributions.

Unit III

Introduction to multivariate random variables and distribution functions, variance-covariance matrix, correlation matrix, Bivariate normal distribution, Multivariate normal density.

Unit IV

Principal component Analysis, Dimensionality Reduction, Cluster Analysis: Hierarchical clustering and divisive clustering methods.

Unit V

Simple linear regression, properties, least squares estimation of parameters, Hypothesis test in simple linear regression.

Textbooks / References:

1. S.C Gupta and V.K Kapoor Fundamentals of Mathematical Statistics
2. Anderson T.W (1983): An introduction to multivariate statistical analysis, #rd Ed, Wiley
3. Ronald E.Walpole, Raymond H Myers, Sheron L Myers and Kreying Ye. Probability and Statistics for Engineers and Scientists, Eighth Edition, Pearson Education Asia 2007
4. Douglas C. Montgomery and Elizabeth A Peck and G Geoffrey Vining. "Introduction to linear regression Analysis", Third Edition, John Wiley and Sons, I

26CSA661 GRAPH THEORY AND COMBINATORICS L-T-P-C: 3-1-0-4

Course Description

This course demonstrates the knowledge of fundamental concepts in graph theory. Students will be able to use graphs to solve real life problems. This helps students to develop efficient algorithms for graph related problems in different domains of engineering and science.

Course Objectives

- To understand and apply the fundamental concepts in graph theory.
- To learn computational algorithms.
- To develop fundamental knowledge of combinatorics and complexity.

Course Outcomes

Cos	Description
CO1	Describe basic concepts of graphs.
CO2	Discuss the concepts of Trees and algorithms on trees.
CO3	Describe concepts of planar graphs and vertex colorings.
CO4	Implement concepts of principle of inclusion and exclusions.
CO5	Solve concepts of Polya's Enumeration Formula for enumeration problems.

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2		-	-	-	-	-
CO2	2	1	2	2	-	-	-	1
CO3	2	3	-	-	-	-	-	2
CO4	2	3	-	-	-	-	-	1
CO5	2	3	-	-	-	1	-	2

Prerequisites

- Discrete Mathematics
- Linear Algebra
- Mathematical proof technique (induction, proof by contradiction)

Syllabus

Unit I

Graphs and Sub graphs, isomorphism, matrices associated with graphs, degrees, walks, paths, connected graphs, Euler graph, Hamilton graphs, shortest path algorithm.

Unit II

Trees: Trees, cut-edges and cut-vertices, spanning trees, Cayley's formula, minimum spanning trees, DFS, BFS algorithms. Connectivity: Graph connectivity, k-connected graphs and blocks.

Unit III

Colorings: Vertex colorings, greedy algorithm and its consequences. Edge-colorings, Vizing theorem on edge-colorings. Planar graphs: Euler formula.

Unit IV

Some Essential Problems, Binomial Coefficients, Multinomial Coefficients, Pigeonhole Principle, Principle of Inclusion and Exclusion.

Generating Functions, Double Decks, Counting with Repetition, Fibonacci Numbers, Recurrence Relations.

Unit V

Polya's Theory of Counting, Permutation Groups, Burnside's Lemm, Cycle Index. Polya's Enumeration Formula, deBruijn's generalization.

Textbooks / References:

1. J. A. Bondy and U. S. R. Murty, Graph Theory and Applications, Springer, 2008.
2. Richard A. Brualdi, Introductory Combinatorics, Pearson, 2012
3. D. B. West, Introduction to Graph Theory, P.H.I. 2010.
4. J. H. van Lint and R. M. Wilson, A Course in Combinatorics, Cambridge University Press, 2001.
5. Bollobás, B. Modern Graph Theory (Graduate Texts in Mathematics). New York, NY: Springer-Verlag, 1998.

26CSA662 OPERATIONS RESEARCH AND OPTIMIZATION L-T-P-C: 3-1-0-4

Course Description

The course is intended to impart knowledge in concepts and tools of Operations Research, to understand mathematical models used in Operations Research and to apply these techniques constructively to make effective business decisions.

Course Objectives

This course aims to introduce students to use quantitative methods and techniques for effective decisions–making; model formulation and applications that are used in solving business decision problems.

Course Outcomes

Cos	Description
CO1	Describe concepts of linear programming, duality and methods for solving a linear programming problem.
CO2	Explain mathematical formulation of transportation and assignment problems and solution methods.
CO3	Solve simple games using various techniques.
CO4	Solve nonlinear unconstrained optimization problems.
CO5	Describe problem of sequencing and integer programming problems.

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	1	-	-	2	-	-
CO2	3	2	1	-	-	2	-	-
CO3	3	2	1	-	-	-	-	-
CO4	3	2	1	-	-	-	-	-
CO5	3	1	-	-	-	3	-	-

Syllabus

Unit I

Linear Programming: Introduction - Mathematical Formulations - Solutions – Graphical Method - Simplex Method - Artificial Variables- Big M - Two Phase Methods - Variants in Simplex Method - Duality Theory and Problems.

Unit II

Transportation and its Variants: Definition - Transportation Algorithms and Solutions -Assignment Model - Hungarian Method, Simulation -Types of Simulations - Monte Carlo Simulation.

Unit III

Game Theory: Competitive Games - Rectangular Game - Saddle point - Minmax (Maxmin)Method of Optimal Strategies - Value of the Game. Solution of Games with Saddle Points -Dominance Principle. Rectangular Games without Saddle Point – Mixed Strategy for 2 X 2Games.

Unit IV

Single Variable Non-Linear Unconstrained Optimization

One dimensional Optimization methods, Uni-modal function, Region elimination methods - interval halving, Fibonacci search, Golden section search, point estimation method - successive quadratic search, Gradient based Methods-Newton's method, secant method.

Unit V
Problem of sequencing, n jobs through two machines – two jobs through m machines - n jobs through m machines. Integer Programming Algorithms: Branch and Bound Algorithms and Cutting Plane Algorithm.

Textbooks / References:

1. Kantiswarup, P. K. Gupta and Manmohan, "Operations Research", Seventh Edition, Sultan Chand, 1991.
2. S.S. Rao, "Optimization Theory and Applications", Second Edition, New Age International (P) Limited Publishers, 1995.
3. Hamdy A. Taha (1987): Operations Research– An Introduction, 4/e, Prentice Hall of India, Private Ltd, New Delhi.
4. Kapoor V.K. (2008):Operations Research, 8/e, Sultan Chand & Sons.
5. Kalyanmoy Deb, "Optimization for Engineering Design Algorithms and Examples", Prentice Hall of India, New Delhi, 2004.

26CSA663	Programming for Problem Solving	L-T-P-C: 3-0-2-4

Course Description

This course is designed to strengthen students' logical reasoning and programming abilities through structured problem-solving exercises. It focuses on developing computational thinking, algorithmic problem-solving, and coding skills essential for technical assessments and interviews. Students will gain hands-on experience in writing efficient code, understanding logic-based challenges, and applying programming concepts to real-world scenarios. The course prepares learners to tackle competitive programming problems, technical assessments, and industry-oriented coding tasks effectively.

Course Objective

To develop students' logical reasoning and programming skills through hands-on problem solving, preparing them for coding assessments and technical interviews.

Course Outcomes

COs	Description
CO1	Apply fundamental programming constructs such as loops, conditionals, functions, and recursion to solve basic computational problems.
CO2	Analyze problems using decomposition techniques and implement solutions using arrays, strings, and hashing.
CO3	Solve algorithmic challenges using techniques like sliding window, two-pointer, and frequency maps.
CO4	Demonstrate proficiency in solving coding problems from online platforms under time constraints.
CO5	Develop clean, efficient, and optimized code for real-world applications, considering constraints and edge cases.
CO6	Design and implement a mini-project using command-line interface (CLI) tools that integrates learned programming concepts.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	1	-	-	1	-
CO2	3	3	2	1	-	-	1	-
CO3	2	3	3	2	-	-	2	-
CO4	2	3	3	2	2	-	2	-
CO5	1	2	2	2	2	-	1	-
CO6	1	2	2	3	2	-	2	-

Unit 1: Foundations of Programming

Introduction to logic building, Control structures: loops, conditionals, Functions and modular programming, Recursion basics, Input/output operations, Debugging and error handling.

Unit 2: Problem-Solving Techniques

Pattern recognition and problem decomposition, Mathematical problems (number theory, combinatorics), String manipulation techniques, Array-based problem solving, Hashing and frequency maps, Sliding window and two-pointer techniques.

Unit 3: Programming Challenges

Daily coding problems (easy to medium level), solving problems from platforms like HackerRank, LeetCode, CodeChef, Group coding sessions and peer reviews, Timed coding tests and contests.

Unit 4: Real-World Applications

Writing clean and efficient code, understanding constraints and edge cases, optimizing solutions. Mini-project: Build a simple CLI tool or game (e.g., quiz app, calculator, to-do list).

Textbooks/References:

1. Let Us C" by Yashavant Kanetkar – A classic book for foundational programming concepts and exercises.
2. Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein – For deeper understanding of algorithmic techniques (selected chapters).
3. Cracking the Coding Interview" by Gayle Laakmann McDowell – Excellent for preparing students for coding interviews and challenges.

26CSA664	ADVANCED PROGRAMMING USING PYTHON	L-T-P-C: 3-0-2-4

Course Description

This course aims to build strong programming foundations in Python, emphasizing both core and advanced features. Students will develop problem-solving skills using Python, focusing on logical reasoning, data structures, and algorithmic thinking. The course covers object-oriented programming concepts, advanced functionalities such as decorators, generators, file handling, and regular expressions. Through hands-on exercises and practical coding challenges, students will learn to write efficient, maintainable, and scalable Python programs for real-world applications

Course Objective

To equip students with strong programming foundations using Python, focusing on core logic building, object-oriented programming, advanced features like decorators and generators, file handling, regular expressions, and problem-solving patterns using Python.

Course Outcomes

COs	Description
CO1	Apply core and object-oriented programming principles in Python.
CO2	Use advanced Python features such as iterators, generators, decorators, and context managers.
CO3	Develop robust Python applications using modules, error handling, and unit testing. CO4: Implement algorithmic problem-solving techniques using Python.
CO4	Handle text data using regular expressions and work with file systems efficiently

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	3	-	-	-	1	-
CO2	3	3	3	1	-	-	1	-
CO3	2	3	3	2	1	-	2	-
CO4	3	3	3	2	1	-	2	-
CO5	1	1	1	2	2	-	1	-

Syllabus

Unit 1: Introduction & Object-Oriented Programming
Python syntax and semantics, data types, control flow, functions, modules, packages, recursion and lambda functions, file handling, exception handling, and debugging techniques. Classes and Objects, Constructors, inheritance, polymorphism, Magic methods (__init__, __str__, __len__, etc.), Composition vs Inheritance, class and static methods.

Unit 2: Advanced Python Programming Concepts, Iterators and Generators, Decorators and closures, Context managers (with statement), Modules and packages (standard & custom), Dynamic typing and introspection.

Unit 3: File and Text Processing
Text, binary file handling, Directory handling using os, shutil, glob, working with JSON and CSV, Regular Expressions using the re module (search, match, replace, extract patterns), Use- case: Log file analyzer

Unit 4: Problem-Solving Patterns in Python
Sorting, searching, recursion, Hashing, frequency counters, sliding window, Backtracking problems, Basic graph and tree problems using dictionaries/lists, Competitive programming style challenges (Leetcode-style)

Unit 5: Testing and Best Practices
Writing testable code, Unit testing with unittest or pytest, Code documentation and style (PEP8), Error logging and debugging, Packaging Python applications. Git and GitHub basics, writing clean, maintainable code, and creating a portfolio of Python projects.

Text Books/ References

1. Python Programming: An Introduction to Computer Science by John Zelle
2. Fluent Python by Luciano Ramalho
3. Test-Driven Development with Python by Harry J.W. Percival

OPEN LABS

26CSA585 PYTHON SCRIPTING FOR SECURITY L-T-P-C: 0-0-2-1

Course Objectives

In this course students will learn to program in python, using an object-oriented approach. Students will learn and write short and long programs to use python to write programs that automate common security tasks.

Course Outcomes

COs	Description
CO1	Explain the fundamentals of computing with python.
CO2	Solve tasks with OOP python scripts.
CO3	Implement Python OS module functions
CO4	Build networking concepts with Python
CO5	Describe Data processing and Visualization libraries in Python

CO-PO Mapping

PO/PSO	PO 1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	1	3	-	-	-	2	-
CO2	2	2	3	-	-	-	2	-
CO3	1	3	3	-	-	-	2	-
CO4	2	1	3	-	-	-	3	-
CO5	-	-	1	-	-	-	-	-

Syllabus

Unit I: Python Fundamentals & Data Structures

1. Write a Python program to demonstrate the use of variables, input/output operations, and basic data types.
2. Develop programs using conditional statements and looping constructs to solve simple problems (e.g., number classification, factorial).
3. Implement user-defined functions and recursion to solve problems such as Fibonacci series and GCD calculation.
4. Write programs to perform operations on lists, tuples, dictionaries, and sets.
5. Demonstrate mutable vs immutable data types using suitable examples.
6. Implement list comprehensions and dictionary comprehensions for data transformation tasks.

Unit II: Object-Oriented Programming & File Handling

7. Design a Python program using classes and objects to model a real-world application (e.g., Bank Account / Student Management System).
8. Implement inheritance and method overriding in Python.
9. Write a program to serialize and deserialize objects using JSON.
10. Develop a program to serialize and deserialize Python objects using Pickle.
11. Implement exception handling mechanisms and demonstrate debugging techniques.

12. Create and use custom Python modules and packages.

Unit III: Scripting, File System & Process Management

13. Write Python scripts to create, delete, and traverse files and directories using os and pathlib.
14. Develop a program to automate file organization based on file types.
15. Execute system commands using os.system() and subprocess module.
16. Implement process creation and management using os.fork() and os.exec().
17. Write a program to demonstrate inter-process communication and process termination using os.kill().

Unit IV: Networking & Cryptography

18. Develop a client-server application using socket programming for data transmission.
19. Implement a secure socket communication using SSL.
20. Write programs to generate cryptographically secure random numbers using secrets module.
21. Implement hashing using hashlib and verify data integrity.
22. Demonstrate message authentication using MAC and HMAC.
23. Implement symmetric encryption and decryption using Fernet.

Unit V: Web Automation, Data Processing & Visualization

24. Develop a Python program to automate HTTP requests using the requests library.
25. Scrape data from a website using Beautiful Soup.
26. Automate web interactions such as form filling and navigation using Selenium.
27. Process and analyze datasets using NumPy and Pandas.
28. Visualize data using Pandas and Seaborn (bar chart, line graph, heatmap).
29. Mini Project: Develop an end-to-end Python application integrating web data extraction, processing, and visualization.

Textbooks / References:

1. Wesley J. Chun, “Core Python Applications Programming”, 3rd Edition, Pearson Education, 2016.
2. https://automatetheboringstuff.com/ (free online version)
3. realpython.com (free articles only)
4. https://jakevdp.github.io/PythonDataScienceHandbook/ (free online version)

26CSA586	ETHICAL HACKING LAB	L-T-P-C: 0-0-2-1

Course Description

This course introduces students to a wide range of topics related to ethical hacking and penetration testing tools. The course provides an in-depth understanding of how to effectively protect computer networks by simulating attacks in a controlled environment.

Course Objectives

- To Understand the core foundations of ethics in regard to computer security.
- Learn about the hacker mindset and the history of hackers
- Understand basic networking and security technologies
- Gain a basic understanding of security policy
- Learn about basic system defense infrastructure

Course Outcomes

CO	Description
CO1	Build a secure virtualized environment for ethical hacking simulations.
CO2	Demonstrate proficiency in gathering intelligence and enumerating network services.
CO3	Analyze network packets to detect vulnerabilities and intercept sensitive data.
CO4	Execute system and web-based exploits using industry-standard frameworks like Metasploit and Burp Suite.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	-	-	-	3	-	-	-
CO2	-	2	-	-	3	-	-	-
CO3	-	2	-	3	3	-	-	-
CO4	-	-	3	-	3	-	1	-

Syllabus

Unit I: Introduction to Ethical Hacking & Information Security

1. Study and demonstration of Information Security concepts: threats, attack vectors, and security controls.
2. Case study analysis on the impact of unethical hacking and discussion on ethical hacking roles (White Hat, Grey Hat, Black Hat).
3. Identification and classification of information security laws, standards, and compliance frameworks (ISO 27001, IT Act).
4. Hands-on demonstration of penetration testing phases and ethical hacking methodologies.

Unit II: Reconnaissance & Social Engineering

5. Perform passive reconnaissance using search engines and public information (Google hacking basics).
6. Conduct footprinting using WHOIS and DNS lookup tools.
7. Demonstrate network scanning techniques using authorized scanners (e.g., Nmap).
8. Packet capturing and analysis using sniffing tools (e.g., Wireshark).
9. Social engineering attack scenarios—analysis and defense strategies (email phishing simulation).

Unit III: TCP/IP Vulnerabilities & Cryptography

10. Demonstrate TCP/IP vulnerabilities using simulated lab scenarios (IP spoofing and SYN flood analysis).
11. Study and implementation of ICMP-based attack detection and mitigation.
12. Analyze encryption algorithms and popular ciphers (AES, DES, RSA).
13. Perform password strength analysis and cracking demonstration using controlled datasets.
14. Implement cryptographic hashing and verify data integrity.

Unit IV: Attacks, Vulnerabilities & Incident Handling

15. Simulate session hijacking scenarios and analyze prevention techniques.
16. Demonstrate denial-of-service attack detection and mitigation strategies.
17. Study buffer overflow vulnerabilities using secure coding practices.
18. Analyze web application vulnerabilities (SQL Injection, XSS) using test environments.
19. Identify operating system vulnerabilities and apply security patches.
20. Perform incident response and handling using a simulated security breach scenario.

Textbooks / References:

1. Jon Erickson, this world-famous hacking book has two editions, one which was published in 2003 and the other in 2008.
2. Alan T. Norman, Computer Hacking Beginners Guide.

26CSA587 PYTHON PROGRAMMING L-T-P-C: 0-0-2-1

Course Description

The course aims to equip the students with a comprehensive study of Python Programming. The course includes collection data types, Python functions, Python exception handling mechanism and Object-Oriented paradigm in Python programs.

Course Objectives

The main objective of this course is to familiarize the student with Python programming concepts, syntax, semantics, and the runtime environment, as well as with general coding techniques and object-oriented programming.

Course Outcomes

Cos	Description
CO1	Identify the structure, syntax, and semantics of the Python language.
CO2	Solve real world problems by applying the Python Data Structures, Objects, string functions
CO3	Implement functions and modules in python
CO4	Implement the basics of data analysis using advanced Python libraries.
CO5	Describe object-oriented programming in python

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	2	2	-	-	-	1	
CO2	2	2	2	-				1
CO3	2	2	2	-	1			
CO4	2	2	2					
CO5	2	2	3					

Syllabus

1. a. Write a program to get the list of even numbers upto a given number.
   b. Write a program to get the ascii distance between two characters.
   c. Write a program to get the binary form of a given number.
   d. Write a program to convert base36 to octal.
2. a. Write a program to get the number of vowels in the input string (No control flow allowed)
   b. Write a program to check whether a given number has even number of 1's in its binary representation (No control flow, the number can be in any base)
   c. Write a program to sort given list of strings in the order of their vowel counts.

3. a. Write a program to return the top ‘n’ most frequently occurring chars and their respective counts. E.g. aaaaaabbbbcccc, 2 should return [(a 5) (b 4)]

   b. Write a program to convert a given number into a given base.

   Note: Convert the given number into a string in the given base.

   Valid base is 2 <= base <= 36

   Raise exceptions similar to how int (“XX”, YY) does (play in the console to find what errors it raises). Handle negative numbers just like bin and oct do.

4. a. Write a program to sort words in a file and put them in another file. The output file should have only lowecase words, so any upper case words from source must be lowered. (Handle exceptions)

   b. Write a program return a list in which the duplicates are removed, and the items are sorted from a given input list of strings.

5. Write a program to test whether given strings are anagrams are not.

6. a. Write a class Person with attributes name, age, weight (kgs), height (ft) and takes them through the constructor and exposes a method get_bmi_result() which returns one of “underweight”, “healthy”, “obese”

   b. Write a program to convert the passed in positive integer number into its prime factorization form.

   Note: If number = a1 ^ p1 * a2 ^ p2 ... where a1, a2 are primes and p1, p2 are powers >=1 then we represent that using lists and tuples in python as [(a1, p1), (a2, p2), ...]

   e.g.[(2,1), (5,1)] is the correct prime factorization of 10

Textbooks / References:

Mark Lutz & David Ascher, “Learning Python”, Oreilly Publications, 5th edition.

Web References:

docs.python.co

26CSA588	C#.NET	L-T-P-C: 0-0-2-1

Course Description

It belongs to "C" family and inherently has lots of things carried from C programming language. It is the ideal choice of all .net developers because Microsoft has developed C# with features of popular languages to develop different types of .net applications. It has SIMPLICITY of Java, POWER of C++ and PRODUCTIVITY of VB.

Course Objectives

To gain a thorough understanding of the philosophy and architecture of .NET and acquire a working knowledge about the .NET programming model along with database connectivity to develop application programs.

Course Outcomes

Cos	Description
CO1	Learn to use .NET frame work and basic programming concepts in C#.
CO2	Develop programs to solve real world problems using OOPS concepts in C#.
CO3	Understand the Window Programming and event driven programming.
CO4	Use ADO.net to store and retrieve data from database.

CO-PO Mapping

PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO7	PO8
CO								
CO1	2	-	-	-	-	-	-	-
CO2	-	3	1	-	-	-	-	-
CO3	2	-	-	-	1	-	-	-
CO4	-	-	2	-	-	-	-	-

Syllabus

Unit I: .NET Framework & C# Fundamentals

1. Write a C# program to demonstrate the .NET Framework architecture, CLR, MSIL, and assemblies.
2. Develop C# programs using predefined data types and demonstrate value types vs reference types.
3. Create a C# application using classes, objects, constructors, and methods.
4. Implement conditional statements and looping constructs in C#.
5. Write programs to demonstrate arrays, indexers, and properties.

Unit II: OOPS, Strings, Delegates & Exception Handling

6. Write a C# program demonstrating string class methods and properties.
7. Implement enumerations, boxing, and unboxing in C#.
8. Develop programs demonstrating OOPS concepts: encapsulation, inheritance, and polymorphism.
9. Implement interfaces, method overriding, and operator overloading.
10. Write programs using static class members.
11. Develop applications using delegates and events.
12. Implement structured exception handling in C# using try–catch–finally blocks.

Unit III: Windows Programming using Windows Forms

13. Develop a Windows Forms application using basic controls (Label, TextBox, Button, CheckBox, RadioButton).
14. Create a Windows Forms application using ComboBox, ListBox, and ProgressBar controls.
15. Implement menus, MDI forms, and ImageList components.
16. Develop applications using Timer component and demonstrate its functionality.
17. Implement mouse and keyboard event handling in Windows Forms applications.

Unit IV: ADO.NET & Database Connectivity

18. Write a C# program to demonstrate the ADO.NET object model and System.Data namespace.
19. Develop database applications using the connected architecture of ADO.NET.
20. Implement the disconnected architecture using DataSet and DataAdapter.
21. Design data-bound applications using data-bound controls.
22. Develop a CRUD (Create, Read, Update, Delete) application using .NET Data Providers.

Textbooks / References:

1. C# 4.0 the Complete Reference by Herbert Schildt
2. C# by Balaguruswamy
3. Latest version of Andrew Trolsens C# text from Apress (Pro C# 5.0 and the .NET Framework 4.5)
4. Robert Powel, Richard Weeks, C# and the .NET Framework, Techmedia

26CSA589 ANDROID PROGRAMMING L-T-P-C: 0-0-2-1

Course Description

Android Programming course is gaining importance in today’s digital era. Due to the increasing popularity of technology and everything getting digitized, the Android Programming course is winning attention. It helps the aspirants in learning the development of smartphones and other mobile devices. The course helps students learn the development of applications. It expounds the topics like software development, web applications development, app safety and security, app development, app functionality and many more.

Course Objectives

Through this course students will learn skills for creating and deploying Android applications, with particular emphasis on software engineering topics including software architecture, software process, usability, and deployment..The student will learn the basics of Android platform and get to understand the android application development with Firebase database.

Course Outcomes

COs	**Description**
CO1	Identify the Android OS Architecture.
CO2	Interpret the different views, layouts and resource files.
CO3	Apply the different UI components.
CO4	Implement Android Notifications, Android Preferences API and Services.
CO5	Implement Applications with Firebase.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	3	2	-	-	-	1	
CO2	2	2	2	-			1	
CO3	2	3	2	-			2	
CO4	2	3	2					2
CO5	2	2	3				2	

Prerequisite:

- Java Programming
- Windows/Linux OS

Syllabus

Unit I: Introduction to Android

1. Study and report on Android OS architecture, Android components, Dalvik Virtual Machine, and APK file structure.
2. List and compare Android API levels with their version names and features.

Unit II: Android Java Basics & First Application

3. Install and configure Android Studio and Android SDK; study project folder structure.
4. Create a simple Android application and generate APK using Android Studio.
5. Create and configure Android Virtual Device (AVD) and run applications on the emulator.
6. Set up debugging environment and debug applications using Logcat and breakpoints.
7. Deploy and debug an Android application on a physical mobile device.

Unit III: User Interface, Activities, Intents & Notifications

8. Design UI using Views, Layouts, Drawable resources, and input controls.
9. Implement input events and display messages using Toasts.
10. Develop applications using ListView, GridView, Adapters, and Action Bar.
11. Implement menus: options menu, context menu, and sub-menu.
12. Develop applications using Pickers (DatePicker and TimePicker) and Spinners.
13. Create applications demonstrating Activity lifecycle and Fragment lifecycle.
14. Implement explicit and implicit intents for navigation and data passing between activities.
15. Develop an application using Navigation Drawer for screen navigation.
16. Implement Android notifications using Toast, Dialogs (Alert, Progress, DatePicker, TimePicker), and Notification Manager.

Unit IV: Firebase Integration

17. Configure Firebase with Android Studio and connect an Android application.
18. Develop an Android application performing CRUD operations using Firebase Realtime Database / Firestore.

Textbooks / References:

1. Head first Android Development.
2. Android Programming: Pushing the Limits, Wiley By Erik Hellman
3. Android Application Development Black Book, Dreamtech Press, Pradeep Kothari, KLSI

26CSA680	UI/ UX DESIGN	L-T-P-C: 0-0-2-1

Course Description

The increasing possibilities with interactive technology as opened to virtual classrooms for teaching and educating the students. Research has proven that interactive teaching using such visual technologies is much more effective than the traditional methods which help students understand and gain knowledge better. Virtual reality is used in many training scenarios as it consists of a wide range of benefits for academia and industrial needs.

Course Objectives

The aim of the UI/UX course is to provide students with the knowledge of user- centered design, user -centered methods in design, graphic design on screens, simulation and prototyping techniques, usability testing methods, interface technologies and user centered design in corporate perspective. The course is organized around a practical project with iterative design of a graphical user interface to organize information about users into useful summaries with affinity diagrams, to convey user research findings with personas and scenarios and to learn the skill of sketching as a process for user experience design. The students will be given exposure to wireframing and Prototyping software in the various UI/UX Design tools.

Course Outcomes

COs	Description
CO1	Create Graphic Design artworks of your own.
CO2	Explain the functionality of different design related software.
CO3	Solve problems of various layouts using various skills.
CO4	Use skill and knowledge for a better workflow.
CO5	Sketch the best design for a given project.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	2	2	-	-	-	-	-
CO2	2	2	2	-	1	-	-	-
CO3	2	2	2	-	1	-	-	-
CO4	2	2	2	-	1	-	-	-
CO5	2	2	3	-	1	-	-	-

Syllabus

Unit I: Fundamentals of UI Design

1. Study and document the evolution of user interface design and the relationship between UI and UX with real-world examples.

2. Identify and analyze interface conventions and screen-based UI approaches used in popular applications.
3. Design a sample screen applying formal and active elements of interface design.
4. Create a basic UI layout demonstrating visual communication principles and interface composition.

Unit II: User Experience Design Basics

5. Analyze good and poor UX designs of two existing applications and present findings.
6. Conduct a basic user study to understand user needs and expectations.
7. Design a functional layout applying UX principles, interaction design, and navigation patterns.
8. Prepare a UX design flow covering ideation, design, testing, and release stages.

Unit III: UI/UX Design Tools & Prototyping

9. Conduct user interviews and document user insights.
10. Create user personas and device personas based on collected data.
11. Design low-fidelity wireframes using wireframing tools (paper/Figma/Balsamiq).
12. Develop high-fidelity wireframes and polished UI screens using UI design tools.
13. Create an interactive prototype using prototyping tools and export/share the design.

Unit IV: UX Research & Information Architecture

14. Perform user-centered research and define user goals and business goals.
15. Prepare psychographic and demographic profiles of target users.
16. Develop scenarios, task analysis, and user stories for a chosen application.
17. Create mind maps, information architecture, and detailed wireframes.
18. Design information visuals and simple data visualization layouts.

Unit V: UX Projects & Usability Evaluation

19. Design a complete web or mobile UI project (dashboard / product interface).
20. Design interfaces separately for Android and iOS platforms following design guidelines.
21. Conduct heuristic evaluation using usability rules and document findings.
22. Design and perform usability tests (low-fidelity and high-fidelity prototype testing).
23. Analyze usability test results and prepare a UX evaluation report.
24. Perform A/B testing and iterate designs based on insights.
25. Conduct remote usability testing and document feedback.

Textbooks / References:

1. A Project Guide to UX Design: For user experience designers in the field or in the making (2nd. ed.). Russ Unger and Carolyn Chandler. New Riders Publishing, USA, 2012.
2. The Elements of User Experience: User-Centered Design for the Web and Beyond, Second Edition Jesse James Garrett, Pearson Education. 2011.
3. The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, Third Edition Wilbert O. Galitz , Wiley Publishing, 2007. 4. The UX Book Process and Guidelines for Ensuring a Quality User Experience, Rex Hartson and Pardha S. Pyla, Elsevier, 2012
4. Weathers David. (2021). “UX/UI Design 2021 For Beginners: A Simple Approach to UX/UI Design for Intuitive Designers” (ISBN-13 : 979-8719605470)
5. Branson Steven (June 2020) “UX / UI Design: Introduction Guide To Intuitive Design And User-Friendly Experience” (ISBN-13 : 979-8653877315)
6. Anderson Gail. (2016). “The Typography Idea Book: Inspiration from 50 Masters” (ISBN-10 : 1780678495, ISBN-13 : 978-1780678498)
7. Slade-Brooking Catharine (2016). “Creating a Brand Identity: A Guide for Designers: (Graphic Design Books, Logo Design, Marketing”. (ISBN-10 : 1780675623, ISBN-13 : 978-1780675626)

26CSA681	LINUX PROGRAMMING	L-T-P-C: 0-0-2-1

Course Description

This course deals with the basic Linux commands to the internal working of Linux operating system. Also gives insights to shell programming and other administration commands.

Course Objectives

It helps the students to learn utilities available in Linux for daily use to write scripts to manage and administer the system.

Course Outcomes

COs	Description
CO1	Explain Linux commands and file/directory management commands.
CO2	Explain the general-purpose utilities available in Linux
CO3	Solve tasks using shell scripts
CO4	Describe the process management and synchronization techniques in Linux.

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	1	1	-	-	-	1	-
CO2	2	1	2	-	-	-	2	-
CO3	3	2	1	-	-		2	-
CO4	3	2	1	-	-	-	2	-

Syllabus

Unit I: File and Directory Management

1. Practice Linux file and directory management commands: ls, pwd, cd, mkdir, rmdir.
2. Create, remove, and modify directories and verify the current working directory.
3. Demonstrate file permissions using chmod and verify access rights.
4. Practice file ownership operations using chown, lchown, and fchown.
5. Create and manage **hard links** and **soft links** using link, symlink, and unlink.

Unit II: General Purpose Linux Utilities

6. Practice user and group management commands: useradd, groupadd, userdel, groupdel, passwd.
7. Use system and session information commands: who, whoami, tty, uname, clear.
8. Use text processing utilities: echo, printf, bc, ls.
9. Practice **input/output redirection** and pipelines using <, >, >>, |.
10. Perform text searching and editing using grep, sed, and awk.

Unit III: Shell Programming (BASH)

11. Write and execute a basic shell script and understand shell responsibilities.
12. Demonstrate shell meta characters and file-name substitution.
13. Use shell variables and command substitution in scripts.

14. Implement control structures: if, case, for, while, until.
15. Perform arithmetic operations using shell scripts.
16. Demonstrate pipes, redirection, and here documents.
17. Write shell scripts using test command and quoting mechanisms.
18. Implement interrupt processing using signal handling (trap).
19. Debug shell scripts using debugging options.

Unit IV: Process Management & IPC

20. Demonstrate process management commands: ps, top, kill, bg, fg.
21. Implement **pipes** and **named pipes (FIFOs)** for inter-process communication.
22. Write programs demonstrating **synchronization and locking techniques**.
23. Implement **shared memory** based inter-process communication.
24. Demonstrate **socket programming** for client–server communication.

Textbooks / References:

1. Unix and Shell Programming, B.A.Forouzan&R.F.Gilberg,Cengage Learning
2. Linux System Programming, Robert Love, O'Reily, SPD.
3. Begining Linux Programming, 4th Edition, N. Matthew, R.Stones, Wrox, Wiley India Edition.

26CSA682	COMPETITIVE PROGRAMMING	L-T-P-C: 0-0-2-1

Course Description

This course helps the students to apply familiar algorithms to solve complex problems as well as to write efficient code, which is necessary for a programmer.

Course Objectives

In this course students will learn how to apply algorithms in order to solve complex problems. The goal of this course is to teach students how to apply familiar algorithms to non-intuitive problems. Along the way students will also gain useful skills for which competitive programmers are so highly valued by employers: ability to write efficient, reliable, and compact code, manage your time well when it‘s limited, apply basic algorithmic ideas to real problems, etc.

Course Outcomes

COs	Description
CO1	Demonstrate knowledge of algorithms and programming languages.
CO2	Solve real world problems.
CO3	Describe competitive programming.
CO4	Describe approaches applied at the world competitions.
CO5	Implement programming concepts with competitive up solving contest.

CO-PO Mapping

PO CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO1	1	2	1	-	-	-	-	-
CO2	2	2		1	1	-	-	
CO3	1	1	2	1	-	-	-	-
CO4	2	2	2	-	1	-	-	-
CO5	1	2	1		1	-	-	-

Syllabus

Unit I: Data Structures and Libraries

1. Implement and analyze sorting algorithms: Bubble, Selection, Insertion, Merge, and Quick sort.
2. Implement dynamic arrays using standard libraries and analyze resizing strategies.
3. Implement and use iterators for traversing data structures.
4. Implement Binary Tree creation and traversal (Inorder, Preorder, Postorder).
5. Implement Binary Search Tree (BST) with insertion, deletion, and search operations.
6. Implement a Trie data structure and perform word insertion, search, and prefix matching.

Unit II: Problem Solving Paradigms

7. Solve problems using Divide and Conquer (Binary Search, Merge Sort).
8. Implement Greedy algorithms (Activity Selection, Fractional Knapsack).
9. Implement Dynamic Programming problems:
   - Fibonacci sequence
   - Longest Common Subsequence
   - Matrix Chain Multiplication
   - 0/1 Knapsack

Unit III: Graph Algorithms

10. Implement Depth First Search (DFS) and Breadth First Search (BFS) using adjacency lists.
11. Apply DFS and BFS to detect connected components and cycle detection.
12. Implement Minimum Spanning Tree using Kruskal’s Algorithm.
13. Implement Shortest Path Algorithms:

- Dijkstra’s Algorithm
- Bellman-Ford Algorithm
- Floyd-Warshall Algorithm

14. Implement Maximum Flow using Edmonds-Karp Algorithm.
15. Implement algorithms for special graphs (DAG shortest path, bipartite graph checking).

Unit IV: Mathematics & String Processing

16. Implement Number Theory algorithms: GCD, LCM, prime number generation (Sieve of Eratosthenes).
17. Implement programs for factorial computation using recursion and dynamic programming.
18. Implement Combinatorics problems (nCr using DP).
19. Solve problems based on Probability theory simulations.
20. Implement Linear Algebra operations: Matrix addition, multiplication, and transpose.
21. Implement String Processing Algorithms: Pattern matching (Naive, KMP, Rabin-Karp).

Unit V: Computational Geometry

22. Implement Graham’s Scan Algorithm for Convex Hull computation.
23. Implement Line Segment Intersection detection algorithms.
24. Solve geometric problems involving points, orientation, and distance calculations.

Textbooks / References:

1. Competitive Programming 3 by Felix Halim and Steven Halim.
2. Guide to Competitive Programming: Learning and Improving Algorithms Through Contests by Antti Laaksonen.
3. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald Rivest, and Clifford Stein.
4. The Algorithm Design Manual by Steven Skiena.
5. Concrete Mathematics by Donald Knuth, Oren Patashnik, and Ronald Graham, 648 pages.
6. Computational Geometry: Algorithms and Applications by Marc van Kreveld, Mark de Berg, and Otfried Cheong.

26CSA683	EDGE COMPUTING	L-T-P-C: 0-0-2-1

Course Description

The course is intended to develop the student's knowledge and abilities of how edge computing and Internet of Things (IoT) can be used as a way to meet application demands in intelligent IoT systems. This includes an understanding and use of the IoT architecture with its entities and protocols, from the IoT devices, via middle layers like edge and fog, up to the cloud. It also includes the understanding of the computing and communication technologies used for IoT, as well as the analysis of their constraints, as e.g. perfor• mance, power efciency, memory size, and communication bandwidth. The course also includes the security and privacy issues related to the area of edge computing, IoT, and big data. Further, it is intended to provide the possibility for the student to, from the basis of relevant literature, reect over and discuss current research and development regarding highly demanding streaming applications, like advanced sen• sing or machine learning, at the edge of an IoT system. The student should be able under supervision to implement an edge and IoT systems.

Course Objectives

- build a basic IoT system which includes edge computations
- investigate, discuss, and compare architectural design options regarding the tradeoff between computations and communication in an IoT system, depending on application demands and resource constraints
- identify, read, and understand relevant scientfic publications; review, discuss, and summarize them, and present the findings both orally and in written form.
-

Course Outcomes

After completing this course, students will be able to:

COs	Description
CO1	Describe basic requirements of edge computing.
CO2	Discuss architectures and applications in fog and edge computing.
CO3	Use fog and edge computing services.
CO4	Demonstrate tools and its usages
CO5	Implement software using standard open-source fog and edge computing software for data analytics.

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	-	-	-	-	-	-
CO2	3	1	-	-	-	-	-	-
CO3	1	-	3	-	-	-	-	-
CO4	-	-	3	-	-	-	-	-
CO5	1	3	-	-	-	-	-	-

Syllabus

Unit I: IoT & Edge Computing Fundamentals

1. Study and demonstration of IoT and Edge Computing use cases (Smart home, Healthcare, Smart agriculture).
2. Comparative analysis of Edge vs Fog vs Cloud Computing using real-world scenarios.
3. Implementation of M2M communication models using simulation tools.
4. Design of an Edge computing architecture for a given use case (block diagram and workflow).

Unit II: IoT Architecture & Case Studies

5. Design and implementation of a basic IoT architecture showing devices, gateways, edge, and cloud.
6. Analysis of IoT vs M2M vs SCADA with suitable examples.
7. Case study implementation: Telemedicine / Palliative Care IoT System – requirement analysis and deployment architecture.
8. Implementation of a sample IoT deployment and evaluation of performance at edge vs cloud.

Unit III: Raspberry Pi Basics & Device Interfacing

9. Familiarization with Raspberry Pi hardware, GPIO pin configuration, and OS installation.
10. Establish SSH-based remote access to Raspberry Pi.
11. Interface DHT11/DHT22 temperature and humidity sensor with Raspberry Pi.
12. Configure Raspberry Pi as a web server to display sensor data.
13. Interface Pi Camera for image and video capture.
14. Perform basic image and video processing using Raspberry Pi.

Unit IV: MQTT & Edge-to-Cloud Communication

15. Install and configure MQTT broker (Mosquitto) on Raspberry Pi.
16. Implement MQTT publish–subscribe model for sensor data transmission.
17. Study MQTT packet structure and state transitions through practical demonstration.
18. Implement MQTT 3.1.1 working example for edge-to-cloud data communication.
19. Analyze MQTT data formats and QoS levels.

Unit V: Edge Computing Applications

20. Develop an Edge computing application using Raspberry Pi (local data filtering/analytics).
21. Implement Edge-based alert system for industrial or commercial IoT use cases.
22. Comparative performance analysis of Edge vs Cloud processing (latency, bandwidth).
23. Mini Project: Industrial/Commercial IoT solution using Edge computing (Smart monitoring system).

Textbooks / References:

1. IoT and Edge Computing for Architects - Second Edition, by Perry Lea, Publisher: Packt Publishing, 2020, ISBN: 9781839214806
2. Raspberry Pi Cookbook, 3rd Edition, by Simon Monk, Publisher: O'Reilly Media, Inc., 2019, ISBN: 978149204322.
3. Fog and Edge Computing: Principles and Paradigms by Rajkumar Buyya, Satish Narayana Srirama, wiley publication, 2019, ISBN: 9781119524984.
4. David Jensen, “Beginning Azure IoT Edge Computing: Extending the Cloud to the Intelligent Edge, MICROSOFT AZURE

26CSA684 R PROGRAMMING L-T-P-C: 0-0-2-1

Course Description

The course is intended to develop the student's knowledge and abilities of how R programming can be used for data analysis and visualization. This course is also intended to get the idea of how it can be applied in various machine learning tasks.

Course Objectives

- The main objective is to provide information on R studio environment.
- It focuses on the basic commands and its syntax.
- Focuses on how to do exploratory data analysis using R programming
- Apply R programming on various ML models and its performance evaluation.

Course Outcomes

COs	Description
CO1	Explain the basic syntax of R programming language in RStudio environment.
CO2	Implement the Pre-processing of raw data in R for further analysis.
CO3	Conduct exploratory data analysis and create insightful visualizations to identify patterns.
CO4	Demonstrate machine learning algorithms for supervised and unsupervised learning.
CO5	Evaluate the performance of models and degree of certainty of predictions

CO-PO Mapping

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	-	2	-	1	-	-	
CO2	1	1	-	1	-	-		
CO3	2	1	-	-	1	-		
CO4	3		2			-	-	
CO5	3						1	1

Syllabus

Unit I: Data Science Process & Data Handling in R

1. Introduction to the Data Science lifecycle with a real-world case study.
2. Loading datasets in R using read.csv(), read.table(), readxl, and data() functions.
3. Exploratory Data Analysis (EDA) using summary statistics and basic plots.
4. Data visualization using ggplot2 for univariate and bivariate analysis.
5. Data cleaning and preprocessing: handling missing values, outliers, and data transformation.
6. Data management in R: data frames, merging datasets, subsetting, and reshaping data.

Unit II: Modeling Methods

7. Implement Supervised Learning models:
   - Linear Regression
   - Logistic Regression
8. Implement Classification algorithms:
   - Decision Trees

    - k-Nearest Neighbors (k-NN)
9. Implement Unsupervised Learning techniques:
    - k-Means clustering
    - Hierarchical clustering
10. Implement Ensemble Models:
    - Random Forest
    - Boosting techniques
11. Model evaluation and comparison using accuracy, precision, recall, RMSE, and cross-validation.

Unit III: Delivering Results & Visualization

12. Create effective visualizations using ggplot2, lattice, and base R graphics.
13. Generate statistical summaries and dashboards using R Markdown.
14. Document and deploy data science results using reports and presentations.
15. Mini Project: End-to-end data science workflow—from data loading to model building and result reporting.

Textbooks / References:

1. "R for Data Science", Hadley Wickham and Garett Grolemund, , O'Reilly, 2017
2. "Data Mining for Business Analytics: Concepts, Techniques and Applications in R", GalitShmueli, et al, Wiley India, 2018.
3. "Practical Data Science with R", Nina Zumel and John Mount, Dreamtech/Manning, 2014

26CSA685 MATLAB PROGRAMMING L-T-P-C: 0-0-2-1

Course Description

This course helps the students with how to use the MATLAB software for image processing. This also explains how Matlab can be used for Mathematical Modeling, Linear Algebra, Numerical Analysis. The various tool boxes in MATLAB along with designing GUI applications are also covered in this course.

Course Objectives

- To impart the knowledge to the students with MATLAB software.
- To introduce students the use of a high-level programming language, Matlab.

Course Outcomes

COs	Description
CO1	Define the use MATLAB software to the students.
CO2	Discuss the MATLAB technical computing environment.
CO3	Explain the use of high-level programming language options available in MATLAB to students.
CO4	Discuss the various toolboxes available in MATLAB.
CO5	Apply the different tools for developing GUI based applications in MATLAB.

CO-PO Mapping

PO/P SO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO7	PO8
CO								
CO1	2	1	1	-	-	-	1	-
CO2	2	2	1	-	-	-	-	-
CO3	2	2	2	-	-	-	-	-
CO4	2	2	3	-	-	-	-	-
CO5	3	3	3	-	-	-	2	-

Syllabus

Unit I: MATLAB Fundamentals

1. Familiarization with MATLAB environment: Desktop, Command Window, Workspace, and Current Directory.
2. Study of MATLAB data types and variable operations.
3. Writing basic MATLAB programs using arithmetic and logical operations.
4. Implementation of input–output operations in MATLAB.

Unit II: Matrices, Control Structures & Plotting

5. Matrix creation and manipulation using built-in functions.
6. Implementation of control flow statements: if, switch, for, and while.
7. String operations and formatting in MATLAB.
8. Implementation of array operations and matrix algebra.
9. Solving systems of linear equations using MATLAB.
10. Basic plotting and graphics: 2D and 3D plots using plot, subplot, bar, and hist.

Unit III: Numerical Methods & M-files

11. Writing and executing M-file scripts and functions.
12. Debugging MATLAB programs using breakpoints and debugging tools.
13. Computation of Eigenvalues and Eigenvectors of a matrix.
14. Polynomial operations and root-finding methods (Bisection, Newton-Raphson).
15. Interpolation techniques and least squares regression.

Unit IV: MATLAB Toolboxes

16. Statistical data analysis using Statistics and Machine Learning Toolbox.
17. Implementation of basic machine learning models using MATLAB.
18. Image processing operations using Image Processing Toolbox (image enhancement, filtering).
19. Text preprocessing and analysis using Text Analytics Toolbox.
20. Introduction to Deep Learning Toolbox and simple neural network implementation.

Unit V: GUI Design & Application Development

21. Design of a Graphical User Interface (GUI) using MATLAB App Designer.
22. Implementation of GUI components: buttons, sliders, text boxes, and panels.
23. Writing callback functions for GUI components.
24. Designing dialog boxes and menu-driven applications.
25. Mini Project: MATLAB GUI-based application integrating computation, visualization, or data analysis

Textbooks / References:

1. http://www.eng-tips.com/threadminder.cfm?pid=575
2. http://www.matlabtutorials.com/mathforum/
3. http://www.mathworks.in/matlabcentral/
4. http://www.cfd-online.com/Forums/tags/matlab.html
5. http://diydrones.com/forum/topic/listForTag?tag=Matlab
6. MATLAB Manuals and Handbooks
7. Duane Hanselman, Bruce Little Field ―Mastering MATLAB 7, Pearson Education India

26CSA686 DEEP LEARNING LAB L-T-P-C: 0-0-2-1

Course Objectives

Introduce major deep learning algorithms, the problem settings, and their applications to solve real world problems.

Course Outcomes

COs	Description
CO1	Demonstrate deep learning and the main research in this field.
CO2	Design and implement deep neural network systems.
CO3	Use neural networks for various application domains.
CO4	Use deep Learning technologies throughout most of machine learning pipeline.
CO5	Develop algorithms for resolving real world problems.

CO-PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1	2	2	2	-	2	-	-	-
CO2	2	2			2	-	1	-
CO3	2	2	2	-	2	-	-	-
CO4	2	2				-	1	-
CO5	2	2	2			-	-	-

Syllabus

1. CNNs for Hand-written digit recognition using Tensor flow.
2. CNNs for Hand-written digit recognition using Keras.
3. Simple image classification with Inception Model.
4. Demonstrate use of GoogleNet and Hyper-parameter Optimization..
5. Demonstrate use of AlexNet and Hyper-parameter Optimization.
6. Create CONV layer of a CNN.
7. Display details of CONV layer of a CNN.
8. Demonstrate use of Stride and Pad for CONV layers of a CNN.
9. Neuron view of the convolution layer.
10. RELU in CNNs.
11. Pooling and fully connected layers in CNNs
12. Classify movie reviews — binary classification using Keras.
13. Python Code: RNNs for Hand-written digit recognition using Tensorflow
14. Python Code: Bi-directional RNNs for Hand-written digit recognition using Tensorflow
15. Python Code: Next word prediction using RNNs

Textbooks / References:

1. Domingos, Pedro. "A few useful things to know about machine learning." Communications of the ACM 55.10 (2012): 78-87.
2. Li Fei-Fei (Stanford), Rob Fergus (NYU), Antonio Torralba (MIT), “Recognizing and Learning Object Categories” (Awarded the Best Short Course Prize at ICCV 2005).

3. Baydin, AtilimGunes, Barak A. Pearlmutter, and Alexey AndreyevichRadul. "Automatic differentiation in machine learning: a survey." arXiv preprint arXiv:1502.05767 (2015).
4. Bengio, Yoshua. "Practical recommendations for gradient-based training of deep architectures." Neural Networks: Tricks of the Trade. Springer Berlin Heidelberg, 2012. 437-478.
5. LeCun, Yann A., et al. "Efficient backprop." Neural networks: Tricks of the trade. Springer Berlin Heidelberg, 2012. 9-48.

26CSA687	**SQLite**	**L-T-P-C: 0-0-2-1**

Course Objective

The course will help the students to understand how an embedded database engine can be used for database-related applications.

Course Outcomes

COs	Description
CO1	Examine the usage, features, installation of SQLite
CO2	Implement operations in database, tables
CO3	Use complex queries like joins, Triggers and keys

CO-PO mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO								
CO1		1			1			
CO2	3	2	2					
CO3	2	2			2			

Syllabus

1 Install and set up the database software (e.g., SQL Server, MySQL, or SQLite). Explore database features and basic syntax.

2 Practice basic datatypes and operators in SQL. Write queries using arithmetic, logical, and comparison operators.

3 Write SQL expressions and evaluate them using SELECT statements.

4 Create a new database and attach/detach it. Explore database management commands.

5 Create tables with various datatypes and constraints (PRIMARY KEY, NOT NULL).

6 Drop existing tables safely using DROP TABLE.

7 Insert records into tables using INSERT INTO.

8 Update existing records using UPDATE with conditions.

9 Delete records from tables using DELETE with conditions.

10 Write SELECT queries with WHERE clause and different conditions (>, <, =, LIKE, IN, BETWEEN).

11 Perform JOIN operations: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN.

12 Create and use triggers for automatic actions on table events (INSERT, UPDATE, DELETE).

13 Work with DATE and TIME functions: CURRENT_DATE, NOW(), DATEADD, DATEDIFF, etc.

14 Define and use keys in tables: PRIMARY KEY, FOREIGN KEY, UNIQUE, and AUTO_INCREMENT/IDENTITY.

15 Mini project: Create a database for a small business or student management system with multiple tables, keys, triggers, and CRUD operations.

References:

1. Using SQLite by Jay A Kreibich
2. The Definitive Guide to SQLite 2nd ed. Edition by Grant Allen and Mike Owens
3. Android SQLite Essentials by Sunny Kumar Aditya and Vikash Kumar Karn