# COURSE STRUCTURE
# AND
# DETAILED SYLLABUS

**For**

M.Tech., COMPUTER SCIENCE

*(Applicable for batches admitted from 2025-2026)*

**(V25 Regulations)**

08818-284577, 284355 Ext: 321; Fax: 08818-284577 Visit us at: www.srivasaviengg.ac.in

SRI VASAVI ENGINEERING COLLEGE

(Autonomous)

(Sponsored by Sri Vasavi Educational Society)

(Approved by AICTE, New Delhi & Permanently affiliated to JNTUK, Kakinada)

(Accredited by NAAC with A Grade ,Recognized by UGC under section 2(f) & 12(B))

Pedatadepalli, **TADEPALLIGUDEM – 534 101.** W.G.Dist**. (A.P)**

**JAWAHARLALNEHRUTECHNOLOGICALUNIVERSITY: KAKINADA**

*(Established by Andhra Pradesh Act No.30 of 2008)*

**Kakinada–533 003,Andhra Pradesh(India)**

**ACADEMIC REGULATIONS** R25 FOR M.Tech(REGULAR) PG **DEGREE COURSE**

Applicable for the students admitted to M. Tech (Regular) Course from the Academic Year 2025-26 and onwards. The M. Tech Degree of Jawaharlal Nehru Technological University Kakinada shall be conferred on candidates who are admitted to the program and who fulfill all the requirements for the award of the Degree.

## 1.0 ELIGIBILITY FOR ADMISSIONS

Admission to the above program shall be made subject to eligibility, qualification and specialization as prescribed by the University/affiliating bodies from time to time.

Admissions shall be made on the basis of merit/rank obtained by the candidates at the qualifying Entrance Test conducted by the State Govt./University or on the basis of any other order of merit as approved by the State Govt./University, subject to reservations as laid down by the Govt. from time to time.

## 2.0 AWARD OF M. Tech DEGREE

A student shall be declared eligible for the award of the M. Tech Degree, if he pursues a course of study in not less than two and not more than four academic years.

**The student shall register for all 80 credits and secure all the 80 credits.**

The minimum instruction days in each semester are 90.

## 3.0 PROGRAMME OF STUDY

The following specializations are offered at present for the M.Tech Programme of study.

**M.Tech in**

1.Advanced Manufacturing Systems
2.Agricultural Engineering
3.AI&DS
4.AI&ML
5.Artificial Intelligence
6.Avionics
7.Bio-technology
8.CAD/CAM
9.Civil Engineering
10.Communication Systems
11.ComputerAided Design &Computer Aided Manufacture
12.Computer Aided Structural Analysis & Design
13.Computer Science
14.Computer Science& Engineering
15.Computer Science& Technology
16.CSE
17.CSE (AI&ML)
18.CSE (AI)
19.CSE(Cyber security)

20.CSE(Data Science)
21.CSE (DS)
22.CSE (IOT)
23.CSE(AI&ML)
24.Cyber Security
25.DataScience
26.Digital Electronics & Communication Systems
27.Digital Electronics and Communication Engineering
28.Digital Electronics and Communication Systems
29.Digital Systems & Computer Electronics
30.ECE (VLSI & Embedded Systems)
31.ElectricPowerSystem
32.ElectricVehicleTechnology
33.Electrical &Power Engineering
34.Electrical Machine & Drives
35.Electrical Power Systems
36.Electronics & Communication(VLSI Design)
37.Electronics and Communication(VLSI Design)
38.Embedded System &VLSI
39.Embedded System & VLSI Design
40.Embedded Systems
41.Environmental Engineering
42.Environmental Engineering & Management
43.Environmental Occupational Health & Safety
44.Highway Engineering
45.Information Technology
46.Internet of Things(IoT)
47.Machine Design
48.Mechanical(CAD/CAM)
49.Microwave &Communication Engineering
50.Mining Engineering
51.Nano-Technology
52.Power and Industrial Drives
53.Power Electronics
54.Power Electronics & Drives
55.Power Electronics & Electrical Drives
56.Power Electronics & Power Systems
57.Power Electronics and Systems
58.Power Electronics Drives
59.Power System Control & Automation
60.Power Systems
61.Power Systems & Automation
62.Power Systems Control & Automation Engineering
63.Remote Sensing
64.Renewable Energy
65.Robotics&Artificial Intelligence
66.Software Engineering

67.Soil Mechanics & Foundation Engineering
68.Structural Engineering
69.Thermal Engineering
70.Thermal Power Engineering
71.Thermal Sciences & Energy Systems
72.Transportation Engineering
73.VLSI
74.VLSI & Embedded Systems
75.VLSI & Embedded Systems Design
76.VLSI Design
77.VLSI Design &Embedded Systems
78.VLSI Embedded Systems
79.VLSI Systems Design

and another course as approved by AICTE/ University from time to time.

## 4.0 ATTENDANCE(SUBJECT WISE PROMOTION IS INTRODUCED)

A student shall be eligible to write University examinations if he acquires a minimum of 75% of attendance in aggregate of all the subjects / courses, and with minimum 75% in each and every course including Laboratories.

Condonation of shortage of attendance in aggregate up to 10% for only hospitalization (65% and above and below 75%) in each course / semester shall be granted by the College Academic Committee.(ONLY ONE TIME in the entire course work)

Shortage of Attendance **below** 65% in aggregate shall not be condoned and not eligible to write their end semester examination of that course.

Students whose shortage of attendance is not condoned in any semester/course are not eligible to write their end semester/course examination of that class.

A prescribed fee shall be payable towards condonation of shortage of attendance.

A student shall not be permitted to appear for the Semester End Examination(SEE) in a course unless they meet the prescribed attendance requirements for that course. Such students may re-register for the course in the subsequent semester when it is offered. The college must obtain permission from the University by submitting the list of students eligible for re-registration at least 30 days prior to the commencement of classwork

If any candidate fulfils the attendance requirement in the present semester/for the course, he/she shall not be eligible for re-admission/re-register into the same class.

## 5.0 EVALUATION

The performance of the candidate in each semester shall be evaluated course-wise, with a maximum of 100 marks for theory and 100 marks for practical, on the basis of Internal Evaluation and End Semester Examination.

For the theory courses 60 marks shall be awarded based on the performance in the End Semester Examination and 40 marks shall be awarded based on the Internal Evaluation. The continuous / internal evaluation shall be made based on the average of the marks secured in the two Mid Term-Examinations conducted-one in the middle of the Semester and the other immediately after the completion of instruction. Each midterm examination shall be conducted for a total duration of 120 minutes with 4 questions (without choice) each question for 10 marks. End semester examination is conducted for 60 marks for all FIVE (5) questions (one question from one unit) to be answered (either or).

For practical subjects, 60 marks shall be awarded based on the performance in the End Semester

Examinations and 40 marks shall be awarded based on the day-to-day performance as Internal Marks. The internal evaluation based on the day to day work-10 marks, record- 10 marks and the remaining 20 marks to be awarded by conducting an internal laboratory test. The end examination shall be conducted by the examiners, with breakup marks of Procedure-20, Experimentation-20, Results-10, Viva-voce-10.

For Seminar, a student under the supervision of a faculty member, shall collect the literature on a topic and critically review the literature and submit it to the department in a report form and shall make an oral presentation before the Project Review Committee consisting of Head of the Department, supervisor/mentor and two other senior faculty members of the department. For Seminar, there will be only internal evaluation of 100 marks. A candidate has to secure a minimum of 50% of marks to be declared successful.

A candidate shall be deemed to have secured the minimum academic requirement in a subject if he secures a minimum of 40% of marks in the End semester Examination and a minimum aggregate of 50% of the total marks in the End Semester Examination and Internal Evaluation taken together.
In case the candidate does not secure the minimum academic requirement in any subject (as specified in 5.4) he has to re-appear for the End semester Examination in that subject. A candidate shall be given **one** chance to re-register for each subject provided the internal marks secured by a candidate **are less than 50% and has failed in the end examination.** In such a case, the candidate must re-register for the subject(s) and secure the required minimum attendance. The candidate's attendance in the re-registered subject(s) shall be calculated separately to decide upon his eligibility for writing the end examination in those subject(s). In the event of the student taking another chance, his internal marks and end examination marks obtained in the previous attempt shall stands cancelled. For re-registration the candidates have to apply to the University through the college by paying the requisite fees and get approval from the University before the start of the semester in which re-registration is required.
In case the candidate secures less than the required attendance in any re-registered subject(s), he/she shall not be permitted to write the End Examination in that subject. He / She shall again re-register the subject when next offered.
Laboratory examination for M. Tech. courses must be conducted with two Examiners, one of them being the Laboratory Class Teacher or teacher of the respective college and the second examiner shall be appointed by the University from the panel of examiners submitted by the respective college.

Students shall undergo mandatory summer internship / industrial training (2 credits) for a minimum of eight weeks duration at the end of second semester of the Programme/Summer Break. A student will be required to submit a summer internship/industrial training report to the concerned department and appear for an oral presentation before the committee. The Committee comprises of a Professor of the department and two faculty. The report and the oral presentation shall carry 40% and 60% weightages respectively. For summer internship / industrial training, there will be only internal evaluation of 100 marks. A candidate has to secure a minimum of 50% of marks to be declared successful.

The objective of comprehensive viva-voce is to assess the overall knowledge of the student in the relevant field of Engineering/Specialization in the PG program. Viva will be conducted in 3$^{rd}$ semester. The duration of the viva will be around 30 min. The examination committee will be constituted by the HoD and consist of Professor of the department and two faculty. For comprehensive viva-voce, there will be only internal evaluation of 100 marks. A candidate has to secure a minimum of 50% of marks to be declared successful.

## 6.0 EVALUATION OF PROJECT / DISSERTATION WORK

Every candidate shall be required to submit a thesis or dissertation on a topic approved by the Project Review Committee.

A Project Review Committee (PRC) shall be constituted with Head of the Department and two other senior faculty members in the department.

Registration of Dissertation/Project Work: A candidate is permitted to register for the project work after satisfying the attendance requirement of all the subjects, both theory and practical.

After satisfying 6.2, a candidate has to submit, in consultation with his project supervisor, the title, objective and plan of action of his project work for approval. The student can initiate the Project work, only after obtaining the approval from the Project Review Committee (PRC).

If a candidate wishes to change his supervisor or topic of the project, he can do so with the approval of the Project Review Committee (PRC).However, the PRC shall examine whether or not the change of topic/supervisor leads to a major change of his initial plans of project proposal. If yes, his date of registration for the project work starts from the date of change of Supervisor or topic as the case may be.

Continuous assessment of Dissertation-Part A and Dissertation-Part B during the Semester(s) will be monitored by the PRC. *Dissertation-Part A* will be only internal evaluation by PRC for 100 marks. A candidate has to secure a minimum of 50% of marks to be declared successful.

The candidate shall submit a status report to the PRC in two stages, each accompanied by an oral presentation, with a minimum interval of three months between the two.

The work on the project shall be initiated at the beginning of the II year and the duration of the project is two semesters. A candidate is permitted to submit Project Thesis only with the approval of PRC not earlier than 40 weeks from the date of registration of the project work.

Three copies of the Project Thesis certified by the supervisor shall be submitted to the College/School/Institute.

The thesis shall be adjudicated by one examiner selected by the University. For this, the Principal of the College shall submit a panel of 5 examiners, eminent in that field, with the help of the guide concerned and head of the department.

If the report of the examiner is not favourable, the candidate shall revise and resubmit the Thesis, in the time frame as decided by the PRC. If the report of the examiner is not favourable again, the thesis shall be summarily rejected. The candidate has to reregister for the project and complete the project within the stipulated time after taking the approval from the University.

If the report of the examiner is favourable, Viva-Voce examination shall be conducted by a board consisting of the Supervisor, Head of the Department and the examiner who adjudicated the Thesis. The Head of the Department shall coordinate and make arrangements for the conduct of Viva-Voce examination. The Board shall jointly report the candidate's work for a maximum of 100 marks as one of the following:

A. Excellent
B. Good
C. Satisfactory
D. Unsatisfactory

If the report of the Viva-Voce is unsatisfactory (i.e., < 50 marks), the candidate shall retake the Viva-Voce examination only after three months. If he fails to get a satisfactory report at the second Viva-Voce examination, the candidate has to re- register for the project and complete the project within the stipulated time after taking the approval from the University.

**6.0** Cumulative Grade Point Average(CGPA)

| **Marks Range Theory/ Laboratory (Max – 100)** | **Marks Range Mini Project/ Project Work for Dissertation (Max – 100)** | **Letter Grade** | **Level** | **Grade Point** |
|---|---|---|---|---|
| ≥ 90 | ≥ 90 | O | Outstanding | 10 |
| ≥80 to <90 | ≥80 to <90 | S | Excellent | 9 |
| ≥70 to <80 | ≥70 to <80 | A | Very Good | 8 |
| ≥60 to <70 | ≥60 to <70 | B | Good | 7 |
| ≥50 to <60 | ≥50 to <60 | C | Fair | 6 |
| <50 | <50 | F | Fail | 0 |
| | | | Absent | 0 |

**Computation of SGPA**

- The following procedure is to be adopted to compute the Semester Grade Point Average(SGPA) and Cumulative Grade Point Average(CGPA):
- The **SGPA** is the ratio of sum of the product of the number of credits with the grade points scored by a student in all the courses taken by a student and the sum of the number of credits of all the courses undergone by a student, i.e

  **SGPA(Si) = ∑(CiX Gi) / ∑Ci**
- Where Ci is the number of credits of the ith course and Gi is the grade point scored by the student in the ith course.

**Computation of CGPA**

- The **CGPA** is also calculated in the same manner taking into account all the courses undergone by a student over all the semester of a Programme, i.e.

  **CGPA=∑(CiXSi)/∑Ci**
- Where Si is the SGPA of the $i^{th}$ semester and Ci is the total number of credits in that semester.
- The SGPA and CGPA shall be rounded off to 2 decimal points and reported in the transcripts.
- Equivalent Percentage= (CGPA)x10

## 7.0 AWARD OF DEGREE AND CLASS

After a student has satisfied the requirements prescribed for the completion of the program and is eligible for the award of M. Tech. Degree he shall be placed in one of the following four classes:

| **Class Awarded** | **CGPA to be secured** | |
|---|---|---|
| First Division with Distinction | ≥ 7.75(without supplementary History) | From the CGPA secured from80 credits |
| First Class | ≥6.75 | |
| Second Class | ≥5.0to <6.75 | |

The secured grade, grade points, status and credits obtained will be shown separately in the memorandum of marks.

If a student wants to leave the program / exit after successful completion of first two semesters, he/she will be awarded Post Graduate Diploma in the specialization concerned.

## 8.0 WITHHOLDING OF RESULTS

If the student is involved in indiscipline/malpractices/court cases, the result of the student will be withheld.

## 9. 0 GENERAL

Whereverthewords“he”,“him”,“his”,occurintheregulations,theyinclude“she”, “her”, “hers”.

The academic regulation should be read as a whole for the purpose of any interpretation.

In the case of any doubt or ambiguity in the interpretation of the above rules, the decision of the Vice-Chancellor is final.

The University may change or amend the academic regulations or syllabi at any time and the changes or amendments made shall be applicable to all the students with effect from the dates notified by the University.

## MALPRACTICES RULES

## DISCIPLINARYACTION FOR IMPROPER CONDUCT IN EXAMINATIONS

| | **Nature of Malpractices/Improper conduct** | **Punishment** |
|---|---|---|
| | *If the candidate:* | |
| 1. (a) | Possesses or keeps accessible in examination hall, any paper, note book, programmable calculators, Cell phones, pager, palm computers or any other form of material concerned with or related to the subject of the examination (theory or practical) in which he is appearing but has not made use of (material shall include any marks on the body of the candidate which can be used as an aid in the subject of the examination) | Expulsion from the examination hall and cancellation of the performance in that subject only. |
| (b) | Gives assistance or guidance or receives it from any other candidate orally or by any other body language methods or communicates through cell phones with any candidate or persons in or outside the exam hall in respect of any matter. | Expulsion from the examination hall and cancellation of the performance in that subject only of all the candidates involved. In case of an outsider, he will be handed over to the police and a case is registered against him. |
| 2. | Has copied in the examination hall from any paper, book, programmable calculators, palm computers or any other form of material relevant to the subject of the examination (theory or practical) in which the candidate is appearing. | Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted to appear for the remaining examinations of the subjects of that Semester/year.<br><br>The Hall Ticket of the candidate is to be cancelled and sent to the University. |
| 3. | Impersonates any other candidate in connection with the examination. | The candidate who has impersonated shall be expelled from examination hall. The candidate is also debarred and forfeits the seat. The performance of the original candidate who has been impersonated, shall be cancelled in all the subjects of the examination (including practicals and project work) already appeared and shall not be allowed to appear for examinations of the remaining subjects of that semester/year. The candidate is also debarred for two consecutive semesters from class work and all University examinations. The continuation of the course by the candidate is subject to the academic regulations in connection with forfeiture of seat. If the imposter is an outsider, he will be handed over to the police and a case is registered against him. |

| | | |
|---|---|---|
| 4. | Smuggles in the Answer book or additional sheet or takes out or arranges to send out the question paper during the examination or answer book or additional sheet, during or after the examination. | Expulsion from the examination hall and cancellation of performance in that subject and all the other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred for two consecutive semesters from class work and all University examinations. The continuation of the course by the candidate is subject to the academic regulations in connection with forfeiture of seat. |
| 5. | Uses objectionable, abusive or offensive language in the answer paper or in letters to the examiners or writes to the examiner requesting him to award pass marks. | Cancellation of the performance in that subject. |
| 6. | Refuses to obey the orders of the Chief Superintendent/Assistant – Superintendent / any officer on duty or misbehaves or creates disturbance of any kind in and around the examination hall or organizes a walkout or instigates others to walk out, or threatens the officer-in charge or any person on duty in or outside the examination hall of any injury to his person or to any of his relations whether by words, either spoken or written or by signs or by visible representation, assaults the officer in-charge, or any person on duty in or outside the examination hall or any of his relations, or indulges in any other act of misconduct or mischief which result in damage to or destruction of property in the examination hall or any part of the College campus or engages in any other act which in the opinion of the officer on duty amounts to use of unfair means or misconduct or has the tendency to disrupt the orderly conduct of the examination. | In case of students of the college, they shall be expelled from examination halls and cancellation of their performance in that subject and all other subjects the candidate(s) has (have) already appeared and shall not be permitted to appear for the remaining examinations of the subjects of that semester/year. The candidates also are debarred and forfeit their seats. In case of outsiders, they will be handed over to the police and a police case is registered against them. |
| 7. | Leaves the exam hall taking away answer script or intentionally tears of the script or any part thereof inside or outside the examination hall. | Expulsion from the examination hall and cancellation of performance in that subject and all the other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred for two consecutive semesters from class work and all University examinations. The continuation of the course by the candidate is subject to the academic regulations in connection with forfeiture of seat |

| | | |
|---|---|---|
| 8. | Possess any lethal weapon or firearm in the examination hall. | Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred and forfeits the seat. |
| 9. | If student of the college, who is not a candidate for the particular examination or any person not connected with the college indulges in any malpractice or improper conduct mentioned in clause 6 to 8. | Student of the colleges expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred and forfeits the seat.<br><br>Person(s) who do not belong to the College will be handed over to police and, a police case will be registered against them. |
| 10. | Comes in a drunken condition to the examination hall. | Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. |
| 11. | Copying detected on the basis of internal evidence, such as, during valuation or during special scrutiny. | Cancellation of the performance in that subject and all other subjects the candidate has appeared including practical examinations and project work of that semester /year examinations. |
| 12. | If any malpractice is detected which is not covered in the above clauses 1 to 11 shall be reported to the University for further action to award suitable punishment. | |

Malpractices identified by squad or special invigilators:

1. Punishments to the candidates as per the above guidelines.
2. Punishment for institutions:(if the squad reports that the college is also involved in encouraging malpractices)
   (i) A show cause notice shall be issued to the college.
   (ii) Impose a suitable fine on the college.
   (iii) Shifting the examination centre from the college to another college for a specific period of not less than one year.

# M.Tech(CS) Programme Course Structure

**(**With effect from **2025-26** Admitted Batch onwards**)**

## SEMESTER-I

| S.No. | Course Code | Course Title | L | T | P | C |
|---|---|---|---|---|---|---|
| 1. | V25CTT01 | Data Structures and Algorithm Analysis | 3 | 1 | 0 | 4 |
| 2. | V25CTT02 | Artificial Intelligence | 3 | 1 | 0 | 4 |
| 3. | V25CTT03 | Mathematical Foundations of Computer Science | 3 | 1 | 0 | 4 |
| 4. | V25CTT04 - V25CTT11 | Program Elective –I | 3 | 0 | 0 | 3 |
| 5. | V25CTT04 - V25CTT11 | Program Elective – II | 3 | 0 | 0 | 3 |
| 6. | V25CTL01 | Data Structures and Algorithms Analysis Lab | 0 | 1 | 2 | 2 |
| 7. | V25CTL02 | Artificial Intelligence Lab | 0 | 1 | 2 | 2 |
| 8. | V25CTL05 | Seminar-I | 0 | 0 | 2 | 1 |
| **Total Credits** | | | **15** | **5** | **6** | **23** |

## SEMESTER-II

| S.No. | Course Code | Course Title | L | T | P | C |
|---|---|---|---|---|---|---|
| 1. | V25CTT12 | Machine Learning | 3 | 1 | 0 | 4 |
| 2. | V25CTT13 | OS and UNIX Programming | 3 | 1 | 0 | 4 |
| 3. | V25CTT14 | Cloud Computing | 3 | 1 | 0 | 4 |
| 4. | V25CTT15 - V25CTT22 | Program Elective – III | 3 | 0 | 0 | 3 |
| 5. | V25CTT15 - V25CTT22 | Program Elective-IV | 3 | 0 | 0 | 3 |
| 6. | V25CTL03 | Machine Learning Lab | 0 | 1 | 2 | 2 |
| 7. | V25CTL04 | OS and Unix lab | 0 | 1 | 2 | 2 |
| 8. | V25CTL06 | Seminar–II | 0 | 0 | 2 | 1 |
| **Total Credits** | | | **15** | **5** | **6** | **23** |

## SEMESTER-III

| S.No. | Course Code | Course Title | L | T | P | C |
|---|---|---|---|---|---|---|
| 1. | V25MOOCS1 | Research Methodology and IPR/ *Swayam 12 week MOOC course – RM&IPR* | 3 | 0 | 0 | 3 |
| 2. | V25CTP01 | Summer Internship/Industrial Training(8-10weeks)* | - | - | - | 3 |
| 3. | V25CTL07 | Comprehensive Viva[#] | - | - | - | 2 |
| 4. | V25CTP02 | Dissertation Part–A[$] | - | - | 20 | 10 |
| **Total Credits** | | | **3** | **-** | **20** | **18** |

**Student attended during summer/ year break and assessment will be done in 3rd Sem.*

*# Comprehensive viva can be conducted courses completed up to second sem.*

*$ Dissertation–Part A, internal assessment*

## SEMESTER-IV

| S.No. | Course Code | Course Title | L | T | P | C |
|---|---|---|---|---|---|---|
| 1. | V25CTP03 | Dissertation Part–B[%] | - | - | 32 | 16 |
| **Total Credits** | | | **3** | **-** | **-** | **-** |

*%External Assessment*

## List of Program Elective Courses in I Semester(Electives I & II)

| S.No. | Course Code | Course Title |
|---|---|---|
| 1 | V25CTT04 | Full Stack Technologies |
| 2 | V25CTT05 | Computer Networks |
| 3 | V25CTT06 | Social Network Analysis |
| 4 | V25CTT07 | Automata Theory and Compiler Design |
| 5 | V25CTT08 | Object Oriented Software Engineering |
| 6 | V25CTT09 | Data Warehousing and Data Mining |
| 7 | V25CTT10 | Advanced Computer Architecture |
| 8 | V25CTT11 | Artificial Neural Networks |

## List of Program Elective Courses in II Semester(Electives III & IV)

| S.No. | Course Code | Course Title |
|---|---|---|
| 1 | V25CTT15 | Data Science |
| 2 | V25CTT16 | Quantum Computing |
| 3 | V25CTT17 | Object Oriented and Analysis Design |
| 4 | V25CTT18 | Cryptography and Network Security |
| 5 | V25CTT19 | Secure coding |
| 6 | V25CTT20 | DevOps |
| 7 | V25CTT21 | Web Application Security |
| 8 | V25CTT22 | Big Data Analytics |

| I SEM | **Data Structures and Algorithm Analysis** | Course Code: V25CTT01 | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|---|
| | | | 3 | 1 | 0 | 4 |

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain basic data structures like linked lists, stacks, queues, and trees. **(K2)**
**CO2:** Describe searching and sorting methods along with time and space complexity. **(K2)**
**CO3:** Explain how trees and graphs are represented and how they can be traversed. **(K2)**
**CO4:** Describe abstract data types like stacks, queues, and hash tables, and explain how collisions are handled. **(K2)**
**CO5:** Apply algorithms to implement and perform operations on data structures such as linked lists, heaps, and search trees. **(K3)**

**UNIT–I: Introduction to Data Structures** - Singly Linked Lists, Doubly Linked Lists, Circular Lists - Algorithms, Stacks and Queues - Algorithm Implementation using Linked Lists, time and space complexity, Asymptotic notations.

**UNIT–II: Searching** - Linear and Binary Search Methods, Sorting - Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Trees - Binary trees, Operations - Insertion, Deletion, Properties, Representation and Traversals (DFT, BFT), Expression Trees (Infix, prefix, postfix), Graphs - Basic Concepts, Storage structures and Traversals.

**UNIT–III:** Dictionaries, ADT, The List ADT, Stack ADT, Queue ADT, Hash Table Representation, Hash Functions, Collision Resolution - Separate Chaining, Open Addressing - Linear Probing, Double Hashing.

**UNIT–IV: Priority Queues** - Definition, ADT, Realising a Priority Queue Using Heaps, Definition, Insertion, Deletion, Search Trees - Binary Search Trees, Definition, ADT, Implementation, Operations - Searching, Insertion, Deletion.

**UNIT–V: Search Trees** - AVL Trees, Definition, Height of AVL Tree, Operations - Insertion, Deletion and Searching. Introduction to Red-Black and Splay Trees, B-Trees, Height of B-Tree, Insertion, Deletion and Searching, Comparison of Search Trees .

**Text Books**:


1. Data Structures: A Pseudo code Approach with C,2nd Edition, Richard F. Gilberg, Behrouz A. Forouzon, Cengage Learning, 2004
2. Data Structures, Algorithms and Applications in java, 2nd Edition,SartajSahni, University Press/Orient Black Swan, 2005


**Reference Books**:


1. Data Structures and Algorithm Analysis,2nd Edition, Mark Allen Weiss,Pearson,2002
2. Data Structures and Algorithms in C++,3rd Edition, Adam Drozdek, Cengage Learning, 2005
3. C and Data Structures: A Snap Shot Oriented Treatise Using Live Engineering Examples, 1st Edition, N.B. Venkateswarulu, E.V. Prasad, S Chand &Co,2009
4. Classic Data Structures,2nd Edition, Debasis Samantha, PHI Learning,2009

| I SEM | **Artificial Intelligence** | Course Code: V25CTT02 | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|---|
| | | | 3 | 1 | 0 | 4 |

**Syllabus Details**

## Course Outcomes: After completion of course, students would be able to

**CO1:** Describe the basics, history, and applications of Artificial Intelligence. **(K2)**
**CO2:** Apply game-playing methods and logic concepts to solve problems. **(K3)**
**CO3:** Use different knowledge representation techniques in AI. **(K3)**
**CO4:** Apply probability-based methods to handle uncertainty. **(K3)**
**CO5:** Explain and use fuzzy logic for reasoning with uncertainty. **(K3)**

**UNIT-I: Introduction to Artificial Intelligence:** Introduction, history, intelligent systems, foundations of AI, applications, tic-tac-toe game playing, development of AI languages, current trends in AI.
**Problem Solving – State-Space Search and Control Strategies:** Introduction, general problem solving, characteristics of problems, exhaustive searches, heuristic search techniques, iterative-deepening A*, constraint satisfaction.

**UNIT-II: Problem Reduction and Game Playing:** Introduction, problem reduction, game playing, alpha-beta pruning, two-player perfect information games.
**Logic Concepts:** Introduction, propositional calculus, propositional logic, natural deduction system, axiomatic system, semantic tableau system in propositional logic, resolution refutation in propositional logic, predicate logic.

**UNIT-III: Knowledge Representation:** Introduction, approaches to knowledge representation, knowledge representation using semantic networks, extended semantic networks for KR, knowledge representation using frames.
**Advanced Knowledge Representation Techniques:** Introduction, conceptual dependency theory, script structure, CYC theory, case grammars, semantic web.

**UNIT-IV: Uncertainty Measure – Probability Theory:** Introduction, probability theory, Bayesian belief networks, certainty factor theory, Dempster-Shafer theory, non-monotonic reasoning, TMS (Truth Maintenance System).

**UNIT-V: Fuzzy Sets and Fuzzy Logic:** Introduction, fuzzy sets, fuzzy set operations, types of membership functions, multi-valued logic, fuzzy logic, linguistic variables and hedges, fuzzy propositions, inference rules for fuzzy propositions, fuzzy systems.

**Text Books**:
1. Artificial intelligence, A modern Approach, 2nd ed, Stuart Russel, Peter Norvig, Prentice Hall
2. Artificial Intelligence, Saroj Kaushik, 1st Edition, CENGAGE Learning, 2011.

**Reference Books**:
1. Artificial intelligence, structures and Strategies for Complex problem solving, 5th Edition, George F Lugar, PEA
2. Introduction to Artificial Intelligence, Ertel, Wolf Gang, Springer, 2017
3. Artificial Intelligence, A new Synthesis, 1st Edition, Nils J Nilsson, Elsevier, 1998
4. Artificial Intelligence- 3rd Edition, Rich, Kevin Knight, Shiv Shankar B Nair, TMH
5. Introduction To Artificial Intelligence And Expert Systems, 1st Edition, Patterson, Pearson India, 2015

| I SEM | Mathematical Foundations for Computer Science | Course Code: V25CTT03 | L | T | P | C |
|---|---|---|---|---|---|---|
| | | | 3 | 1 | 0 | 4 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain probability concepts, random variables, and distribution functions in discrete and continuous cases.**(K2)**

**CO2:** Describe sampling methods, estimation techniques, and computation of statistical measures for grouped data. **(K2)**

**CO3:** Illustrate hypothesis testing procedures, error types, significance levels, and quality control methods in statistical analysis. **(K2)**

**CO4:** Apply algebraic properties, number theory algorithms, and modular arithmetic concepts to solve mathematical problems. **(K3)**

**CO5:** Construct graphs using matrix representations, paths, circuits, spanning trees, and graph colouring techniques. **(K3)**

**UNIT–I: Basic Probability and Random Variables:** Random Experiments, Sample Spaces, Events, the Concept of Probability, the Axioms of Probability, Some Important Theorems on Probability, Assignment of Probabilities, Conditional Probability, Theorems on Conditional Probability, Independent Events, Bayes Theorem or Rule. Random Variables, Discrete Probability Distributions, Distribution Functions for Random Variables, Distribution Functions for Discrete Random Variables, Continuous Random Variables.

**UNIT–II:** Sampling and Estimation Theory: Population and Sample, Statistical Inference, Sampling With and Without Replacement, Random Samples, Random Numbers, Population Parameters, Sample Statistics, Sampling Distributions, Frequency Distributions, Relative Frequency Distributions, Computation of Mean, Variance, and Moments for Grouped Data, Unbiased Estimates and Efficient Estimates, Point Estimates and Interval Estimates, Reliability, Confidence Interval Estimates of Population Parameters, Maximum Likelihood Estimates.

**UNIT–III:**Tests of Hypothesis and Significance: Statistical Decisions, Statistical Hypotheses, Null Hypotheses, Tests of Hypotheses and Significance, Type I and Type II Errors, Level of Significance, Tests Involving the Normal Distribution, One-Tailed and Two-Tailed Tests, P-Value, Special Tests of Significance for Large Samples, Special Tests of Significance for Small Samples, Relationship between Estimation Theory and Hypothesis Testing, Operating Characteristic Curves, Power of a Test, Quality Control Charts, Fitting Theoretical Distributions to Sample Frequency Distributions, The Chi-Square Test for Goodness of Fit, Contingency Tables, Yates' Correction for Continuity, Coefficient of Contingency.

**UNIT–IV:** Algebraic Structures and Number Theory: Algebraic Systems, Examples, General Properties, Semi Groups and Monoids, Homomorphism of Semi Groups and Monoids, Group, Subgroup, Abelian Group, Homomorphism, Isomorphism. Properties of Integers, Division Theorem, The Greatest Common Divisor, Euclidean Algorithm, Least Common Multiple, Testing for Prime Numbers, The Fundamental Theorem of Arithmetic, Modular Arithmetic (Fermat's Theorem and Euler's Theorem).

**UNIT–V:** Graph Theory: Basic Concepts of Graphs, Sub graphs, Matrix Representation of Graphs: Adjacency Matrices, Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian and Hamiltonian Graphs, Multi graphs, Planar Graphs, Euler's Formula, Graph Colouring and Covering, Chromatic Number, Spanning Trees, Algorithms for Spanning Trees (Problems Only and Theorems without Proofs).

**Text Books**:


1. Foundation Mathematics for Computer Science,1st edition, John Vince,Springer,2015
2. Probability&Statistics,3rd Edition, Murray R. Spiegel, John J. Schiller and R. Alu Srinivasan, Schaum's Outline Series, Tata McGraw-Hill Publishers, 2018
3. Probability and Statistics with Reliability,2nd edition,K.Trivedi,Wiley,2011
4. Discrete Mathematics and its Applications with Combinatorics and GraphTheory,7th Edition, H. Rosen, Tata McGraw Hill, 2003


**Reference Books**:


1. Probability and Computing: Randomized Algorithms and Probabilistic Analysis,1st edition, M. Mitzenmacher and E.Upfal, 2005
2. Applied Combinatorics,6th edition,AlanTucker,Wiley,2012

| I SEM | **Full Stack Technologies (Program Elective-I & Program Elective-II )** | Course Code: V25CTT04 | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|---|
| | | | 3 | 0 | 0 | 3 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain the concepts of internet protocols, web page structure, XML, and document models for web development. **(K2)**

**CO2:** Describe the basics of JavaScript, AngularJS expressions, and form validation techniques for client-side programming. **(K2)**

**CO3:** Apply PHP and Node.js concepts for server-side scripting and database-driven applications. **(K3)**

**CO4:** Use jQuery and MySQL queries to implement interactive web applications with database connectivity. **(K3)**

**CO5:** Develop and deploy applications using MongoDB, cloud platforms, and web services (SOAP, WSDL, REST). **(K3)**

**UNIT-I: Introduction to Web**- Internet and World Wide Web, Domain Name Service, Protocols: HTTP, FTP, SMTP, HTML5 concepts, CSS3, Anatomy of a Web Page, X**ML**- Document Type Definition, XML Schemas, Document Object Model, XSLT, DOM and SAX Approaches.

**UNIT-II: JavaScript**- The Basics of JavaScript: Objects, Primitives, Operations and Expressions, Control Statements, Arrays, Functions, Constructors, Pattern Matching using Regular Expressions. **Angular JavaScript** – AngularJS Expressions: Array, Objects, $eval, Strings, AngularJS Form Validation & Form Submission.

**UNIT-III: PHP Programming:** Back-end Scripts PHP, Node.js, Working with PHP – Using Variables, Using Constants, Data Types, Operators, Conditional & Control Statements, Arrays, Functions. Working with Forms and Databases such as MySQL. **Node.js** – Introduction, Advantages, Node.js Process Model, Node.js Modules.

**UNIT-IV: jQuery:** Introduction to jQuery, Syntax, Selectors & Events.
**MySQL**: Practice MySQL Queries, Aggregate Functions, Regular Expressions, Joins & Unions, Sub-Queries, Database Connectivity with MySQL.

**UNIT-V: MongoDB** – Introduction, Architecture, Features, Examples, Database Creation & Collection in MongoDB. **Deploying Applications** – Web Hosting & Domains, Deployment Using Cloud Platforms. **Web Services** – SOAP, WSDL and RESTful Architecture.

**Text Books:**


1. Programming the World Wide Web, 7th Edition, Robert W. Sebesta, Pearson, 2013
2. Web Technologies, 1st Edition, 7th Impression, Uttam K. Roy, Oxford, 2012
3. Pro Mean Stack Development, 1st Edition, Elad Elrom, Apress O'Reilly, 2016
4. JavaScript & jQuery: The Missing Manual, 2nd Edition, David Sawyer McFarland, O'Reilly, 2011
5. Web Hosting for Dummies, 1st Edition, Peter Pollock, John Wiley & Sons, 2013
6. RESTful Web Services, 1st Edition, Leonard Richardson, Ruby, O'Reilly, 2007


**Reference Books**:


1. Ruby on Rails Up and Running: Lightning-fast Web Development, 1st Edition, Bruce Tate, Curt Hibbs, O'Reilly, 2006
2. Programming Perl, 4th Edition, Tom Christiansen, Jonathan Orwant, O'Reilly, 2012
3. Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML and AJAX Black Book, 1st Edition, Dream Tech, 2009
4. An Introduction to Web Design, Programming, 1st Edition, Paul S. Wang, Sanda S. Katila, Cengage Learning, 2003


**Web Reference Links**

1. http://www.upriss.org.uk/perl/PerlCourse.html

| I SEM | Computer Networks (Program Elective-I & Program Elective-II ) | Course Code: V25CTT05 | L | T | P | C |
|---|---|---|---|---|---|---|
| | | | 3 | 0 | 0 | 3 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain network layer functions, routing algorithms, and congestion control techniques. **(K2)**

**CO2:** Describe IP protocols, addressing schemes, and internet control protocols. **(K2)**

**CO3:** Illustrate transport protocols like UDP, TCP, and SCTP, including their services and control mechanisms. **(K3)**

**CO4:** Apply wireless network technologies such as WLAN, Bluetooth, WiMAX, and cellular telephony in practical scenarios. **(K3)**

**CO5:** Use mobile computing concepts, sensor networks, mesh networks, and P2P protocols to design secure and efficient communication systems. **(K3)**

**UNIT-I:Network layer:** Network Layer design issues: store-and-forward packet switching, services provided transport layers, implementation of connectionless services, implementation of connection-oriented services, comparison of virtual-circuit and datagram subnets, Routing Algorithms - shortest path routing, flooding, distance vector routing, link state routing, hierarchical routing, congestion control algorithms - approaches to congestion control, traffic-aware routing, admission control, traffic throttling, choke packets, load shedding, random early detection, Quality of Service, application requirements, traffic shaping, leaky and token buckets.

**UNIT-II: Internetworking and IP protocols:** How networks differ, how networks can be connected, internetworking, tunneling, the network layer in the internet, IPV4 protocol, IP addresses, subnets, CIDR, classful and special addressing, network address translation (NAT), IPV6 address structure, address space, IPV6 advantages, packet format, extension headers, transition from IPV4 to IPV6, internet control protocols - ICMP, ARP, DHCP.

**UNIT-III: Transport Layer Protocols:** Introduction, services, port numbers, user datagram protocol (UDP): user datagram, UDP services, UDP applications; transmission control protocol (TCP): TCP services, TCP features, segment, a TCP connection, state transition diagram, windows in TCP, flow control and error control, TCP congestion control, TCP timers; SCTP: SCTP services, SCTP features, packet format, an SCTP association, flow control, error control.

**UNIT-IV: Wireless LANs:** Introduction, architectural comparison, access control; the IEEE 802.11 project: architecture, MAC sub-layer, addressing mechanism, physical layer; Bluetooth: architecture, Bluetooth layers; other wireless networks: WiMAX - services, IEEE project 802.16, layers in project 802.16; cellular telephony: operations, first generation (1G), second generation (2G), third generation (3G), fourth generation (4G); satellite networks: operation, GEO satellites, MEO satellites, LEO satellites.

**UNIT-V:Emerging trends in computer networks:** Mobile computing - motivation for mobile computing, protocol stack issues in mobile computing environment, mobility issues, security issues in mobile networks; MOBILE Ad Hoc Networks - applications, challenges, and issues in MANETs, MAC layer issues, routing protocols, transport layer issues, ad hoc network security; Wireless Sensor Networks - functioning, operating system support, characteristics, operation, sensor architecture, cluster management; Wireless Mesh Networks - design, issues; Computational grids - features, construction design, grid design features; P2P networks - characteristics, classification, Gnutella, BitTorrent; Session Initiation Protocol (SIP) - characteristics, addressing, components, establishment, security.

**Text Books**:


1. Data Communications and Networking 4th edition Behrouz A Fourzan,TMH-2007
2. Computer Networks 4th edition Andrew S Tanenbaum,Pearson,2012
3. Computer Networks, Mayank Dave, CENGAGE, First edition.2012


**Reference Books**:


1. Computer Networks, A System Approach,5thed,LarryL Peterson and Bruce S Davie, Elsevier-2012.

| I SEM | **Social Network Analysis (Program Elective-I & Program Elective-II )** | Course Code: V25CTT06 | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|---|
| | | | 3 | 0 | 0 | 3 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain key concepts in social network analysis including centrality, balance, and homophily.**(K2)**
**CO2:** Describe random graph models, network growth patterns, and cohesive subgroups. **(K2)**
**CO3:** Illustrate network topology, diffusion processes, and navigation strategies. **(K3)**
**CO4:** Apply small world models and clustering techniques to analyze real-world networks. **(K3)**
**CO5:** Use network algorithms, game theory concepts, and agent-based models to explore network structure and dynamics. **(K3)**

**UNIT-I: Social Network Analysis:** Preliminaries and definitions, Erdos Number Project, centrality measures, balance and homophily.

**UNIT-II: Random Graph Models:** Random graphs and alternative models, models of network growth, navigation in social networks, cohesive subgroups, multidimensional scaling, structural equivalence, roles and positions.

**UNIT-III: Network Topology and Diffusion:** Contagion in networks, complex contagion, percolation and information, navigation in networks revisited.

**UNIT-IV: Small World Models:** Small world experiments, small world models, origins of small world, heavy tails, small diameter, clustering of connectivity, the Erdos–Renyi model, clustering models.

**UNIT-V: Network Structure and Dynamics:** Important vertices and page rank algorithm, towards rational dynamics in networks, basics of game theory, coloring and consensus, biased voting, network formation games, network structure and equilibrium, behavioral experiments, spatial and agent-based models.

**Text Books:**


1. S. Wasserman and K. Faust. Social Network Analysis: Methods and Applications(Cambridge, Cambridge University Press, 1994)
2. D. Easley and J. Kleinberg, Networks, Crowds and Markets: Reasoning about a highly connected world-2010


**Reference Books:**


1. Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences) by Stanley Wasserman, Katherine Faust, 1994.

| I SEM | **Automata Theory and Compiler Design (Program Elective-I & Program Elective-II )** | Course Code: V25CTT07 | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|---|
| | | | 3 | 0 | 0 | 3 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1**: Interpret the structure and behavior of finite automata through alphabets, states, and transitions. **(K2)**

**CO2**: Analyze regular expressions and grammars to identify patterns and ambiguity in language processing. **(K2)**

**CO3**: Examine the computational power of pushdown and Turing machines, and their implications for decidability. **(K2)**

**CO4**: Construct lexical and syntax analyzers by integrating parsing algorithms and grammar rules. **(K3)**

**CO5**: Develop intermediate code and memory management strategies to enhance compiler performance. **(K3)**

**UNIT- I:Introduction to Finite Automata:** Structural representations, automata and complexity, the central concepts of automata theory – alphabets, strings, languages, problems.
Nondeterministic Finite Automata: Formal definition, an application, text search, finite automata with epsilon-transitions. Deterministic Finite Automata: Definition of DFA, how a DFA processes strings, the language of DFA, conversion of NFA with epsilon-transitions to NFA without epsilon-transitions, conversion of NFA to DFA.

**UNIT- II: Regular Expressions:** Finite automata and regular expressions, applications of regular expressions, algebraic laws for regular expressions, conversion of finite automata to regular expressions. Pumping Lemma for Regular Languages: Statement of the pumping lemma, applications of the pumping lemma. Context-Free Grammars: Definition of context-free grammars, derivations using a grammar, leftmost and rightmost derivations, the language of a grammar, parse trees, ambiguity in grammars and languages.

**UNIT- III: Push Down Automata:** Definition of the pushdown automaton, the languages of a PDA, equivalence of PDAs and CFGs, acceptance by final state.
Turing Machines: Introduction to Turing machine, formal description, instantaneous description, the language of a Turing machine. Undecidability: Undecidability, a language that is not recursively enumerable, an undecidable problem that is RE, undecidable problems about Turing machines.

**UNIT- IV: Introduction to Compiler Design:** The structure of a compiler, lexical analysis – the role of the lexical analyzer, input buffering, recognition of tokens, the lexical-analyzer generator Lex. Syntax Analysis: Context-free grammars, writing a grammar, top-down parsing, bottom-up parsing, introduction to LR parsing – simple LR, more powerful LR parsers.

**UNIT- V:Syntax-Directed Translation:** Syntax-directed definitions, evaluation orders for SDDs, syntax-directed translation schemes, implementing L-attributed SDDs. Intermediate-Code Generation: Variants of syntax trees, three-address code. Run-Time Environments: Stack allocation of space, access to nonlocal data on the stack, heap management.

**Text Books:**


1. Introduction to Automata Theory, Languages, and Computation, 3rd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, 2nd Edition, Pearson.
3. Theory of Computer Science–Automata languages and computation, Mishra and Chandra shekaran, 2nd Edition, PHI.


**Reference Books:**


1. Introduction to Formal languages Automata Theory and Computation, Kamala Krithivasan, Rama R, Pearson.
2. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
3. Lex & yacc–John R. Levine, Tony Mason, Doug Brown, O' reilly
4. Compiler Construction, Kenneth C. Louden, Thomson. Course Technology

| I SEM | **Object Oriented Software Engineering (Program Elective-I & Program Elective-II )** | Course Code: V25CTT08 | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|---|
| | | | 3 | 0 | 0 | 3 |

**Syllabus Details**

## Course Outcomes: After completion of course, students would be able to

**CO1:** Explain OOP concepts and UML basics. **(K2)**
**CO2:** Apply OOAD principles using UML diagrams. **(K3)**
**CO3:** Demonstrate test-driven development and software testing. **(K3)**
**CO4:** Describe software maintenance and refactoring techniques. **(K2)**
**CO5:** Apply advanced OOP concepts in software projects. **(K3)**

**UNIT-I: Introduction to Object-Oriented Programming:** Overview of software engineering, introduction to Object-Oriented Programming (OOP) concepts (classes, objects, inheritance, polymorphism), Unified Modelling Language (UML) basics, introduction to software development process and software development lifecycle (SDLC).

**UNIT-II: Requirements Analysis and Design:** Requirements analysis and specification, use cases and scenarios, Object-Oriented Analysis and Design (OOAD), design patterns, UML modelling techniques (class diagrams, sequence diagrams, state machine diagrams, activity diagrams).

**UNIT-III: Software Construction and Testing:** Software construction basics, Object-Oriented design principles, Object-Oriented programming languages (Java, C++, Python), software testing basics (unit testing, integration testing, system testing), test-driven development (TDD).

**UNIT-IV: Software Maintenance and Evolution:** Software maintenance basics, refactoring techniques, software version control, code review and inspection, software evolution and re-engineering.

**UNIT-V: Advanced Topics in Object-Oriented Software Engineering:** Model-driven engineering (MDE), aspect-oriented programming (AOP), component-based software engineering (CBSE), service-oriented architecture (SOA), agile software development and Scrum methodologies.

**Text Books:**


1. Craig Larman, An Introduction to Object-Oriented Analysis and Design and the Unified Process, 3rd Edition, Prentice-Hall.
2. Sachin Malhotra, Programming in Java, Oxford University Press.


**Reference Books:**


1. G. Kotonya, I. Sommerville, Requirements Engineering: Processes and Techniques, Wiley, 1998.
2. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns.
3. J. Rumbaugh, I. Jacobson, G. Booch, The Unified Modeling Language Reference Manual, Addison Wesley.


**Online Learning Resources:**

1. https://nptel.ac.in/courses/106105153 - NPTEL – Object-Oriented Analysis and Design
2. https://nptel.ac.in/courses/106105237 - NPTEL – Object-Oriented System Development
3. https://spoken-tutorial.org/course/ - IIT Bombay Spoken Tutorial – Java Programming
4. https://vlab.amrita.edu/?sub=3&brch=273 - Virtual Labs (Vlabs) – Object-Oriented Programming

| I SEM | Data Warehousing and Data Mining (Program Elective-I & Program Elective-II ) | Course Code: V25CTT09 | L | T | P | C |
|---|---|---|---|---|---|---|
| | | | 3 | 0 | 0 | 3 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Describe the architecture, modeling techniques, and implementation strategies of data warehouses and OLAP systems, including modern cloud-based approaches. **(K2)**

**CO2:** Apply statistical and visualization techniques to summarize datasets and perform preprocessing tasks such as cleaning, integration, reduction, and transformation. **(K3)**

**CO3:** Develop and evaluate classification models using decision trees, Bayesian classifiers, and rule-based methods to address predictive analytics problems. **(K2)**

**CO4:** Discover meaningful associations and sequential patterns in datasets using algorithms like Apriori, FP-Growth, and sequential pattern mining techniques. **(K3)**

**CO5:** Implement clustering techniques such as K-means, hierarchical clustering, and DBSCAN, and analyze advanced data mining methods for text, spatial, and graph data. **(K3)**

**UNIT-I: Data Warehousing and Online Analytical Processing:** Basic concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Cloud Data Warehouse; Data Mining Methodologies: CRISP-DM and SEMMA, Comparison of Data Mining Methodologies. Statistical Limits on Data Mining, Introduction to Predictive Analytics, Technologies, Applications, Major issues (Text Book- 1)

**UNIT-II:** Data Objects & Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity. **Data Preprocessing:** An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization. (Text Book- 1)

**UNIT-III: Classification:** General Approach to solving a classification problem, Decision Tree Induction: Attribute Selection Measures, Tree Pruning, Scalability and Decision Tree Induction, Visual Mining for Decision Tree Induction, Bayesian Classification Methods: Bayes Theorem, Naïve Bayes Classification, Rule-Based Classification, Model Evaluation and Selection. (Text Book- 2)

**UNIT-IV: Association Analysis:** Problem Definition, Frequent Itemset Generation, Rule Generation: Confidence-Based Pruning, Rule Generation in Apriori Algorithm, Compact Representation of frequent item sets, FP-Growth Algorithm. Sequential Patterns: Preliminaries, Sequential Pattern Discovery. (Text Book- 2)

**UNIT-V: Cluster Analysis:** Clustering techniques, Different Types of Clusters; K-means: The Basic K-means Algorithm, K-means Additional Issues, Bi-secting K Means, Agglomerative Hierarchical Clustering: Basic Agglomerative Hierarchical Clustering Algorithm, DBSCAN: Traditional Density Center-Based Approach, DBSCAN Algorithm, Strengths and Weaknesses. Mining rich data types: Mining text data, Spatial-temporal data, Graph and networks. (Text Book- 2)

**Text Books**:

1. Data Mining concepts and Techniques, 3rd edition, Jiawei Han, Michel Kamber, Elsevier, 2011.
2. Introduction to Data Mining: Pang-Ning Tan & Michael Steinbach, Vipin Kumar, Pearson, 2012.

**Reference Books**:

1. Data Mining: Vikram Pudi and P. Radha Krishna, Oxford Publisher.
2. Data Mining Techniques, Arun K Pujari, 3rd edition, Universities Press, 2013.

**Online Resources:**

1. http://onlinecourses.nptel.ac.in/noc17_mg24/preview - NPTEL course by Prof. Pabitra Mitra
2. http://www.saedsayad.com/data_mining_map.htm

| I SEM | Advanced Computer Architecture<br>**(Program Elective-I & Program Elective-II )** | Course Code:<br>V25CTT10 | **L**<br>3 | **T**<br>0 | **P**<br>0 | **C**<br>3 |
|---|---|---|---|---|---|---|

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain advanced concepts of computer architecture, including parallel and scalable systems. **(K2)**

**CO2:** Illustrate different parallel architectures. **(K2)**

**CO3:**Explain parallel programming concepts using appropriate models and languages. **(K3)**

**CO4:**Apply the design principles of pipelined and multiprocessor systems. **(K3)**

**CO5:** Demonstrate knowledge of modern computer architectures, I/O devices, and low-level programming concepts. **(K3)**

**UNIT-I: Theory of Parallelism:** Parallel Computer Models, The State of Computing, Multiprocessors and Multicomputer, Multi-vector and SIMD Computers, PRAM and VLSI Models, Program and Network Properties, Conditions of Parallelism, Program Partitioning and Scheduling, Program Flow Mechanisms, System Interconnect Architectures, Principles of Scalable Performance, Performance Metrics and Measures, Parallel Processing Applications, Speedup Performance Laws, Scalability Analysis and Approaches.

**UNIT-II: Hardware Technologies:** Processors and Memory Hierarchy, Advanced Processor Technology, Superscalar and Vector Processors, Memory Hierarchy Technology, Virtual Memory Technology.

**UNIT-III: Bus, Cache, and Shared Memory:** Bus Systems, Cache Memory Organizations, Shared Memory Organizations, Sequential and Weak Consistency Models, Pipelining and Superscalar Techniques, Linear Pipeline Processors, Nonlinear Pipeline Processors, Instruction Pipeline Design, Arithmetic Pipeline Design.

**UNIT-IV: Parallel and Scalable Architectures:** Multiprocessors and Multicomputers, Multiprocessor System Interconnects, Cache Coherence and Synchronization Mechanisms, Three Generations of Multicomputers, Message-Passing Mechanisms, Multi-vector and SIMD Computers, Vector Processing Principles, Multi-vector Multiprocessors, Compound Vector Processing, SIMD Computer Organizations (up to 8.4), Scalable, Multithreaded and Dataflow Architectures, Latency-Hiding Techniques, Principles of Multithreading, Fine-Grain Multicomputers, Scalable and Multithreaded Architectures, Dataflow and Hybrid Architectures.

**UNIT-V: Software for Parallel Programming:** Parallel Models, Languages, and Compilers, Parallel Programming Models, Parallel Languages and Compilers, Dependence Analysis of Data Arrays, Parallel Program Development and Environments, Synchronization and Multiprocessing Modes. Instruction and System Level Parallelism, Instruction Level Parallelism, Computer Architecture, Basic Design Issues, Problem Definition, Model of a Typical Processor, Compiler-detected Instruction Level Parallelism.

**Text Books:**


1. "Advanced Computer Architecture (SIE): Parallelism, Scalability, Programmability", Kai Hwang and Naresh Jotwani, McGraw Hill Education 3rd Edition. 2015
2. "Computer Architecture and Parallel Processing", Kai Hwang and Faye Briggs, Mc Graw-Hill International Edition, 2000


**Reference Books:**


1. John L. Hennessy and David A. Patterson, Computer Architecture:A quantitative approach, 5th edition, Morgan Kaufmann Elseveir, 2013
2. Introduction to High Performance Computing for Scientists and Engineers,G. Hager and G. Wellein, CRC Press, Taylor & Francis Group.
3. Sima D, Fountain T and Kacsuk P, "Advanced Computer Architectures: A Design Space Approach", Addison Wesley, 2000.

| I SEM | Artificial Neural Networks<br>**(Program Elective-I & Program Elective-II )** | Course Code:<br>V25CTT11 | **L**<br>3 | **T**<br>0 | **P**<br>0 | **C**<br>3 |
|---|---|---|---|---|---|---|

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain neural network architectures, including feed forward and feedback networks. **(K2)**

**CO2:** Apply learning rules to train neural networks. **(K3)**

**CO3:** Interpret testing methods and performance metrics for neural networks in pattern recognition. **(K2)**

**CO4:** Apply Self-Organizing Maps (SOM), feature mapping models, algorithms, and simulations for classification. **(K3)**

**CO5:** Explain neuro-dynamics concepts, including dynamical systems, stability, attractors, and Hopfield models. **(K2)**

**UNIT-I: Introduction:** A Neural Network, Human Brain, Models of a Neuron, Neural Networks viewed as Directed Graphs, Network Architectures, Knowledge Representation, Artificial Intelligence and Neural Networks.
**Learning Process:** Error Correction Learning, Memory Based Learning, Hebbian Learning, Competitive, Boltzmann Learning, Credit Assignment Problem, Memory, Adaptation, Statistical Nature of the Learning Process.

**UNIT-II: Single Layer Perceptrons:** Adaptive Filtering Problem, Unconstrained Organization Techniques, Linear Least Square Filters, Least Mean Square Algorithm, Learning Curves, Learning Rate Annealing Techniques, Perceptron – Convergence Theorem, Relation Between Perceptron and Bayes Classifier for a Gaussian Environment.
**Multilayer Perceptron:** Back Propagation Algorithm, XOR Problem, Heuristics, Output Representation and Decision Rule, Computer Experiment, Feature Detection.

**UNIT-III: Back Propagation:** Back Propagation and Differentiation, Hessian Matrix, Generalization, Cross Validation, Network Pruning Techniques, Virtues and Limitations of Back Propagation Learning, Accelerated Convergence, Supervised Learning.

**UNIT-IV: Self-Organization Maps (SOM):** Two Basic Feature Mapping Models, Self-Organization Map, SOM Algorithm, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, Adaptive Pattern Classification.

**UNIT-V:** Neuro Dynamics: Dynamical Systems, Stability of Equilibrium States, Attractors, Neuro Dynamical Models, Manipulation of Attractors as a Recurrent Network Paradigm.
Hopfield Models: Hopfield Models, Computer Experiment.

**Text Books**:

1. Neural Networks a Comprehensive Foundations, Simon Haykin, PHI edition.

**Reference Books**:

1. Artificial Neural Networks -B. Vegnanarayana Prentice Hall of India PLtd 2005
2. Neural Networks in Computer Intelligence, LiMinFu MCGRAW HILLE DUCATION 2003
3. Neural Networks-James A Freeman David M S Kapura Pearson Education2004.
4. Introduction to Artificial Neural Systems Jacek M. Zurada, JAICO Publishing House Ed.2006.

| I SEM | Data Structures and Algorithms Analysis Lab | Course Code: V25CTL01 | L | T | P | C |
|---|---|---|---|---|---|---|
| | | | 0 | 1 | 2 | 2 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Construct various data structures such as linked lists, stacks, queues, and trees using Java. **(K3)**

**CO2:** Apply sorting and searching algorithms, including Quick Sort, Merge Sort, and Heap Sort, to solve computational problems. **(K3)**

**CO3:** Examine advanced data structures such as graphs, AVL trees, and B-trees to solve complex problems. **(K3)**

**CO4:** Demonstrate the use of hashing and dictionary (ADT) operations to manage and retrieve data efficiently. **(K3)**

## List of Experiments

1. Construct a Java program to perform various operations on a single linked list.
2. Develop a Java program for the following:
   a) Reverse a linked list b) Sort the data in a linked list
   c) Remove duplicates d) Merge two linked lists

3. Develop a Java program to perform various operations on a doubly linked list.
4. Develop a Java program to perform various operations on a circular linked list.
5. Demonstrate a Java program for performing various operations on stack using linked list.
6. Demonstrate a Java program for performing various operations on queue using linked list.
7. Develop a Java program for the following using stack:
   a) Infix to postfix conversion b) Expression evaluation
   c) Obtain the binary number for a given decimal number

8. Develop a Java program to perform various operations on Binary Search Tree using recursive and non-recursive methods.

9. Construct a Java program to perform the following operations on a graph:
   a) BFS b) DFS

10. Demonstrate a Java program to perform Merge & Heap Sort on given elements.
11. Demonstrate a Java program to perform Quick Sort on given elements.
12. Demonstrate a Java program to perform various operations on AVL trees.
13. Construct a Java program to perform the following operations on B-tree:
    a) Insertion b) Searching
14. Demonstrate recursive and non-recursive functions for Binary Tree traversals.
15. Demonstrate all the functions of Dictionary (ADT) using Hashing.

| I SEM | Artificial Intelligence Lab | Course Code: V25CTL02 | L | T | P | C |
|---|---|---|---|---|---|---|
| | | | 0 | 1 | 2 | 2 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Develop Python programs to solve classical AI problems such as the Water Jug, Tic-Tac-Toe, Monkey Banana, and Puzzle Problems using search algorithms.**(K3)**

**CO2:** Apply heuristic and optimization techniques, including the Traveling Salesman Problem, Simulated Annealing, and Hill-Climbing algorithms, to solve complex problems. **(K3)**

**CO3:** Apply advanced search algorithms like A* and AO* in Python to solve path-finding and decision-making problems. **(K3)**

**CO4:** Demonstrate reasoning methods, including forward chaining, backward chaining, and the Min-Max algorithm, in expert systems and game playing applications. **(K3)**

**List of Experiments**

1. Develop a Python program to solve the Water Jug problem using DFS.
2. Develop a Python program to solve the Tic-Tac-Toe problem using BFS.
3. Apply a heuristic approach in Python to solve the Traveling Salesman Problem (TSP).
4. Apply the Simulated Annealing Algorithm in Python to solve optimization problems.
5. Apply the Hill-Climbing Algorithm in Python to solve a Puzzle Problem.
6. Develop a Python program to solve the Monkey Banana Problem.
7. Apply the A* Algorithm in Python for path-finding problems.
8. Apply the AO* Algorithm in Python to solve search problems.
9. Develop a Python program to implement the Min-Max Game Playing Algorithm.
10. Demonstrate forward chaining in an Expert System using Python.
11. Demonstrate backward chaining in an Expert System using Python.

**Text Books:**


1. Artificial intelligence, A modern Approach, 2nd ed,Stuart Russel, Peter Norvig, Prentice Hall
2. Artificial Intelligence, Saroj Kaushik,1st Edition, CENGAGE Learning,2011.


**Reference Books:**


1. Artificial intelligence, Structures and Strategies for Complex problem solving,5th Edition, George F Lugar, PEA.
2. Introduction to Artificial Intelligence, Ertel, Wolf Gang,Springer,2017.
3. Artificial Intelligence, A new Synthesis,1st Edition, Nils J Nilsson,Elsevier,1998.
4. Artificial Intelligence-3rd Edition, Rich, Kevin Knight, Shiv Shankar B Nair, TMH.
5. Introduction to Artificial Intelligence and Expert Systems,1st Edition, Patterson, Pearson India,2015.

| II SEM | Machine Learning | Course Code: V25CTL12 | L | T | P | C |
|---|---|---|---|---|---|---|
| | | | 3 | 1 | 0 | 4 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain the fundamental concepts of supervised and unsupervised learning. **(K2)**
**CO2:** Apply regression, classification, and clustering algorithms to solve real-world problems. **(K3)**
**CO3:** Apply machine learning models using ensemble methods and Python libraries. **(K3)**
**CO4:** Analyze time-series, sequence data, and deep learning models for feature representation. **(K3)**
**CO5:** Develop scalable machine learning solutions for advanced topics, including IoT applications. **(K3)**

**UNIT-I:** Supervised Learning (Regression/Classification) – Basic methods: Distance-based methods, Nearest-Neighbours, Decision Trees, Naive Bayes, Linear models: Linear Regression, Logistic Regression, Generalized Linear Models, Support Vector Machines, Nonlinearity and Kernel Methods, Beyond Binary Classification: Multi-class/Structured Outputs, Ranking.

**UNIT-II:** Unsupervised Learning – Clustering: K-means/Kernel K-means, Dimensionality Reduction: PCA and kernel PCA, Matrix Factorization and Matrix Completion, Generative Models (mixture models and latent factor models).

**UNIT-III:** Evaluating Machine Learning algorithms and Model Selection, Introduction to Statistical Learning Theory, Ensemble Methods (Boosting, Bagging, and Random Forests), Python Libraries – Introduction to Python Libraries: TensorFlow, PyTorch, Keras, Scikit-learn, NumPy, SciPy, Matplotlib, Pandas, Seaborn.

**UNIT-IV:** Sparse Modeling and Estimation, Modeling Sequence/Time-Series Data, Deep Learning and Feature Representation Learning.

**UNIT-V:** Scalable Machine Learning (Online and Distributed Learning) – A selection from some other advanced topics, e.g., Semi-supervised Learning, Active Learning, Reinforcement Learning, Inference in Graphical Models, Introduction to Bayesian Learning and Inference, Recent trends in various learning techniques of machine learning and classification methods for IoT applications, Various models for IoT applications.

**Text Books:**


1. Machine Learning: A Probabilistic Perspective, 1st edition, Kevin Murphy, MIT Press, 2012
2. The Elements of Statistical Learning, 2nd edition, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer 2009 (freely available online)
3. Pattern Recognition and Machine Learning, 2nd edition, Christopher Bishop, Springer, 2011
4. Programming Collective Intelligence: Building Smart Web 2.0 Applications, 1st edition, Toby Segaran, 2007
5. Building Machine Learning Systems with Python, 1st edition, Willi Richert, Luis Pedro Coelho, 2013
6. Applied Machine Learning, 1st edition, M. Gopal, McGraw Hill Education, 2019

| II SEM | OS and UNIX Programming | Course Code: V25CTL13 | L | T | P | C |
|---|---|---|---|---|---|---|
| | | | 3 | 1 | 0 | 4 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain the structure, functions, and services of operating systems. **(K2)**

**CO2:** Apply process scheduling, inter-process communication, and multithreading concepts to solve OS problems. **(K3)**

**CO3:** Explain memory management, virtual memory, and deadlock handling techniques in operating systems. **(K2)**

**CO4:** Demonstrate file system operations and Unix commands for process and directory management. **(K3)**

**CO5:** Develop programs to handle process control, signals, and file operations in Unix/Linux environments. **(K3)**

**UNIT-I: Operating Systems Overview** – Operating system functions, Operating system structure, Operating systems operations, Computing environments, Open-Source Operating Systems.
System Structures – Operating System Services, User and Operating-System Interface, system calls, Types of System Calls, system programs, operating system structure, operating system debugging, System Boot.

**UNIT-II: Process Concept** – Process scheduling, Operations on processes, Inter-process communication, Communication in client-server systems.
**Multithreaded Programming** – Multithreading models, Thread libraries, Threading issues.
Process Scheduling – Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling, Thread scheduling.
**Inter-process Communication** – Race conditions, Critical Regions, Mutual exclusion with busy waiting, Sleep and wakeup, Semaphores, Mutexes, Monitors, Message passing, Barriers, Classical IPC Problems: Dining philosophers problem, Readers and writers problem.

**UNIT-III: Memory-Management Strategies** – Introduction, Swapping, Contiguous memory allocation, Paging, Segmentation. **Virtual Memory Management** – Introduction, Demand paging, Copy on-write, Page replacement, Frame allocation, Thrashing, Memory-mapped files, Kernel memory allocation.
**Deadlocks** – Resources, Conditions for resource deadlocks, Ostrich algorithm, Deadlock detection and recovery, Deadlock avoidance, Deadlock prevention. **File Systems** – Files, Directories, File system implementation, management and optimization. **Secondary-Storage Structure** – Overview of disk structure and attachment, Disk scheduling, RAID structure, Stable storage implementation.

**UNIT-IV: Introduction to Unix** – Architecture of Unix, Responsibilities of shell, Unix file system, vi editor.
**Unix commands** – Some Basic Commands, file utilities, process utilities, text processing utilities, network utilities, disk utilities, backup utilities, Security by file permissions.

**UNIT-V: Files** – Introduction, file descriptors, open, creat, read, write, close, lseek, dup2, file status information (stat family), file and record locking (fcntl function), file permissions (chmod, fchmod), file ownership (chown, lchown), links (soft and hard links – symlink, link, unlink).
**Directories** – Creating, removing and changing Directories (mkdir, rmdir, chdir), obtaining current working directory (getcwd), Directory contents, Scanning Directories (opendir, readdir, closedir, rewinddir functions).
**Process Control** – process identifiers, fork function, vfork function, exit function, wait and waitpid functions, exec functions, user identification.
**Signals** – signal handling using signal function, kill and raise, alarm, pause, abort, and sleep functions.

**Text Books:**

1. Silberschatz A, Galvin P B, and Gagne G, Operating System Concepts, 9th edition, Wiley, 2013
2. Tanenbaum A S, Modern Operating Systems, 3rd edition, Pearson Education, 2008 (for Inter-process Communication and File Systems)
3. Unix the Ultimate Guide, 3rd edition, Sumitabha Das, TMH
4. Advanced Programming in the Unix Environment, W. Richard Stevens
5. Unix Network Programming, W. Richard Stevens

**Reference Books:**

1. Dhamdhere D M, Operating Systems: A Concept-Based Approach, 3rd edition, Tata McGraw-Hill,2012
2. Stallings W, Operating Systems - Internals and Design Principles, 6th edition, Pearson Education,2009
3. Venkateshmurthy, Introduction to Unix and Shell Programming
4. B.M. Harwani, Unix and Shell Programming, Oxford University Press

**e-Resource:**

1. NPTEL – Operating Systems (https://nptel.ac.in/courses/106/105/106105214/)

| II SEM | Cloud Computing | Course Code: V25CTL14 | L | T | P | C |
|---|---|---|---|---|---|---|
| | | | 3 | 1 | 0 | 4 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:**Explain cloud computing fundamentals, service models, deployment models, and major cloud providers. **(K2)**

**CO2:**Describe enabling technologies for cloud such as distributed computing, SOA, web services, and virtualization. **(K2)**

**CO3:**Apply virtualization and container technologies for cloud-based solutions. **(K3)**

**CO4:**Explain challenges in cloud computing related to scalability, interoperability, security, and energy efficiency. **(K3)**

**CO5:**Use advanced cloud concepts like serverless computing, IoT integration, edge/fog computing, and DevOps for modern applications. **(K3)**

**UNIT-I: Introduction to Cloud Computing Fundamentals:** Cloud computing at a glance, defining a cloud, cloud computing reference model, types of services (IaaS, PaaS, SaaS), cloud deployment models (public, private, hybrid), utility computing, cloud computing characteristics and benefits, cloud service providers (Amazon Web Services, Microsoft Azure, Google App Engine).

**UNIT-II: Cloud Enabling Technologies:** Ubiquitous Internet, parallel and distributed computing, elements of parallel computing, hardware architectures for parallel computing (SISD, SIMD, MISD, MIMD), elements of distributed computing, inter-process communication, technologies for distributed computing, remote procedure calls (RPC), service-oriented architecture (SOA), web services, virtualization.

**UNIT-III: Virtualization and Containers:** Characteristics of virtualized environments, taxonomy of virtualization techniques, virtualization and cloud computing, pros and cons of virtualization, technology examples (XEN, VMware), building blocks of containers, container platforms (LXC, Docker), container orchestration, Docker Swarm and Kubernetes, public cloud VM (e.g., Amazon EC2) and container (e.g., Amazon Elastic Container Service) offerings.

**UNIT-IV: Cloud Computing Challenges:** Economics of the cloud, cloud interoperability and standards, scalability and fault tolerance, energy efficiency in clouds, federated clouds, cloud computing security, fundamentals of computer security, cloud security architecture, cloud shared responsibility model, security in cloud deployment models.

**UNIT-V:Advanced Concepts in Cloud Computing:** Serverless computing, Function-as-a-Service, serverless computing architecture, public cloud (e.g., AWS Lambda) and open-source (e.g., OpenFaaS) serverless platforms, Internet of Things (IoT) applications, cloud-centric IoT and layers, edge and fog computing, DevOps, infrastructure-as-code, quantum cloud computing.

**Text Book:**

1. Learning Agile – Andrew Stellman, Jill Alison Hart, O'Reilly, 2015

**Reference Books**:

1. Head First Agile – Andrew Stellman, Jennifer Green, O'Reilly, 2017
2. Essential Scrum: A Practical Guide to the Most Popular Agile Process – Rubin K., Addison-Wesley, 2013

| II SEM | Data Science<br>**(Program Elective–III & Program Elective–IV)** | Course Code:<br>V25CTL15 | **L**<br>3 | **T**<br>0 | **P**<br>0 | **C**<br>3 |
|---|---|---|---|---|---|---|

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain Python programming concepts including data types, control structures, functions, modules, and OOP. **(K2)**

**CO2:** Develop GUI-based and networked Python applications using standard libraries**.(K3)**

**CO3:** Explain NumPy and Pandas for data processing, analysis, and handling missing or hierarchical data. **(K3)**

**CO4:** Illustrate datasets using data wrangling, aggregation, transformation and merging techniques**.(K3)**

**CO5:** Explain data visualizations using Matplotlib and Pandas for time series, financial, and economic datasets.**(K3)**

**UNIT-I:** PYTHON Basics and Programming Concepts – Introducing Python; Types and Operations: Numbers, Strings, Lists, Tuples, Dictionaries, Files, Numeric Types, Dynamic Typing; Statements and Syntax: Assignments, Expressions, Statements, Loops, Iterations, Comprehensions; Functions: Function Basics, Scopes, Arguments, Advanced Functions; Modules: Module Coding Basics, Module Packages, Advanced Module Topics; Classes and OOP: Class, Operator Overloading, Class Designing; Exceptions and Tools: Exception Basics, Exception Coding Details, Exception Objects, Designing with Exceptions, Parallel System Tools.

**UNIT-II:** GUI Programming – Graphical User Interface: Python GUI development options, Adding Widgets, GUI Coding Techniques, Customizing Widgets; Internet Programming: Network Scripting, Client-Side Scripting, Pymailgui client, Server-Side Scripting, Pymailcgi server; Tools and Techniques: Databases and persistence, Data structures, Text and Language, Python/C Integration.

**UNIT-III:** Pandas and NumPy – NumPy Basics: Fast Element-wise array functions, Multidimensional Array, Data Processing using arrays, File I/O with arrays; Pandas: Data Structures, Essential Functionality, Summarizing and Computing Descriptive Statistics, Handling Missing Data, Hierarchical Indexing.

**UNIT-IV:** Data Preprocessing – Data Loading, Storage, and File Formats: Reading and Writing data in text format, Binary data formats, Interacting with HTML and Web APIs, Interacting with Databases; Data Wrangling: Clean, Transform, Merge, Reshape – Combining and Merging Data Sets, Reshaping and Pivoting, Data Transformation, String Manipulation; Data Aggregation and Group Operations – Group by Mechanics, Data Aggregation, Groupby Operations and Transformations, Pivot Tables and Cross Tabulation.

**UNIT-V:** Data Visualization – A Brief Matplotlib API Primer, Plotting Functions in Pandas, Time Series, Financial and Economic Data Applications.

**Text Books:**


1. Learning Python, O'Reilly, Mark Lutz
2. Programming Python, O'Reilly, Mark Lutz
3. Python for Data Analysis, O'Reilly, Wes McKinney


**Reference Books:**


1. Python: The Complete Reference, Martin C. Brown, McGraw Hill Education
2. Head First Python, Paul Barry, O'Reilly

| II SEM | Quantum Computing<br>**(Program Elective–III & Program Elective–IV)** | Course Code:<br>V25CTL16 | **L**<br>3 | **T**<br>0 | **P**<br>0 | **C**<br>3 |
|---|---|---|---|---|---|---|

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain the history, principles, and foundational concepts of quantum computing. **(K2)**
**CO2:** Discuss linear algebra, quantum physics, and biological concepts to understand quantum systems. **(K2)**
**CO3:** Explain qubits and design quantum circuits including single and multiple qubit gates. **(K3)**
**CO4:** Illustrate quantum algorithms such as Deutsch, Deutsch–Jozsa, Shor, and Grover for problem-solving. **(K3)**
**CO5:** Explain quantum error correction techniques and apply quantum information and cryptography protocols. **(K3)**

**UNIT-I: History of Quantum Computing** – Importance of Mathematics, Physics, and Biology. Introduction to Quantum Computing – Bits vs Qubits, Classical vs Quantum logical operations.

**UNIT-II: Background Mathematics** – Basics of Linear Algebra, Hilbert Space, Probabilities, and Measurements.
**Background Physics** – Paul's Exclusion Principle, Superposition, Entanglement and Supersymmetry, Density Operators and Correlation, Basics of Quantum Mechanics, Measurements in bases other than computational basis.
**Background Biology** – Basic concepts of Genomics and Proteomics (Central Dogma).

**UNIT-III: Qubit** – Physical implementations of Qubit, Qubit as a quantum unit of information, The Bloch Sphere. **Quantum Circuits** – Single qubit gates, Multiple qubit gates, Designing quantum circuits, Bell states.

**UNIT-IV: Quantum Algorithms** – Classical computation on quantum computers, Relationship between quantum and classical complexity classes, Deutsch's Algorithm, Deutsch–Jozsa Algorithm, Shor's Factorization Algorithm, Grover's Search Algorithm.

**UNIT-V: Noise and Error Correction** – Graph states and codes, Quantum error correction, Fault-tolerant computation. **Quantum Information and Cryptography** – Comparison between classical and quantum information theory, Quantum Cryptography, Quantum Teleportation.

**Text Books:**
1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge

**Reference Books:**
1. Yanofsky N. S. and Mannucci M. A., Quantum Computing for Computer Scientists
2. Benenti G., Casati G., and Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol. II: Basic Tools and Special Topics, World Scientific
3. Pittenger A. O., An Introduction to Quantum Computing Algorithms

| II SEM | Object Oriented and Analysis Design **(Program Elective–III & Program Elective–IV)** | Course Code: V25CTL17 | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|---|
| | | | 3 | 0 | 0 | 3 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain the complexity of software systems and techniques to bring order in designing complex systems. **(K2 )**

**CO2:** Apply UML modeling principles and structural diagrams to represent software systems. **(K3 )**

**CO3:** Construct class and object diagrams using advanced modeling techniques. **(K3 )**

**CO4:** Analyze behavioral modeling diagrams such as use cases, interactions, and activities for system design. **(K3 )**

**CO5:** Develop architectural models including component and deployment diagrams for software applications. **(K3 )**

**UNIT- I: Introduction** – The Structure of Complex Systems, The Inherent Complexity of Software, Attributes of Complex Systems, Organized and Disorganized Complexity, Bringing Order to Chaos, Designing Complex Systems. **Case Study:** System Architecture – Satellite-Based Navigation.

**UNIT- II: Introduction to UML** – Importance of Modeling, Principles of Modeling, Object-Oriented Modeling, Conceptual Model of UML, Architecture, and Software Development Life Cycle. **Basic Structural Modeling** – Classes, Relationships, Common Mechanisms, and Diagrams. **Case Study:** Control System – Traffic Management.

**UNIT- III: Class & Object Diagrams** – Terms, Concepts, Modeling Techniques for Class & Object Diagrams. **Advanced Structural Modeling** – Advanced Classes, Advanced Relationships, Interfaces, Types and Roles, Packages. **Case Study:** AI – Cryptanalysis.

**UNIT- IV: Basic Behavioral Modeling-I** – Interactions, Interaction Diagrams, Use Cases, Use Case Diagrams, Activity Diagrams. **Case Study:** Web Application – Vacation Tracking System.

**UNIT-V: Advanced Behavioral Modeling** – Events and Signals, State Machines, Processes and Threads, Time and Space, State Chart Diagrams.
**Architectural Modeling** – Component, Deployment, Component Diagrams, and Deployment Diagrams. **Case Study:** Weather Forecasting.

**Text Books:**


1. Grady Booch, Robert A. Maksimchuk, Michael W. ENGLE, Bobbi J. Young, Jim Conallen, Kellia Houston , "Object- Oriented Analysis and Design with Applications",3rd edition, 2013, PEARSON.
2. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education.


**Reference Books:**


1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY- Dream tech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis &Design, The McGraw-Hill Companies.
4. Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.

| II SEM | Cryptography and Network Security<br>**(Program Elective–III & Program Elective–IV)** | Course Code:<br>V25CTL18 | **L**<br>3 | **T**<br>0 | **P**<br>0 | **C**<br>3 |
|---|---|---|---|---|---|---|

**Syllabus Details**

## Course Outcomes: After completion of course, students would be able to

**CO1:** Explain security concepts, attacks, services, and classical encryption methods. **(K2)**
**CO2:** Apply mathematical concepts to understand symmetric and asymmetric cryptography. **(K3)**
**CO3:** Demonstrate symmetric and asymmetric key algorithms. **(K3)**
**CO4:** Explain hash functions, MACs, and digital signatures in security. **(K2)**
**CO5:** Apply security protocols at the Application, Transport, and Network layers. **(K3)**

**UNIT - I: Security Concepts:** Introduction, need for security, security approaches, principles of security, types of security attacks, security services, security mechanisms, a model for network security.
**Cryptography – Classical Encryption Techniques:** Symmetric cipher model, substitution techniques, transposition techniques, rotor machines, steganography.

**UNIT - II: Introduction to Symmetric Cryptography:** Algebraic structures – groups, rings, fields, GF(2n) fields, polynomials.
**Mathematics of Asymmetric Cryptography:** Primes, checking for primeness, Euler's phi-functions, Fermat's Little Theorem, Euler's Theorem, generating primes, primality testing, factorization, Chinese Remainder Theorem, quadratic congruence, exponentiation and logarithm.

**UNIT - III: Symmetric Key Ciphers:** Block cipher principles, DES, AES, Blowfish, IDEA, block cipher operation. **Stream Ciphers:** RC4, RC5.
**Asymmetric Key Ciphers:** Principles of public key cryptosystems, RSA algorithm, Diffie-Hellman key exchange, Elgamal cryptographic system, elliptic curve arithmetic, elliptic curve cryptography.

**UNIT - IV: Cryptographic Hash Functions:** Applications of cryptographic hash functions, two simple hash functions, requirements and security, hash functions based on cipher block chaining, secure hash algorithms (SHA).
**Message Authentication Codes (MACs):** Message authentication requirements, message authentication functions, requirements for MACs, security of MACs, MACs based on hash functions (HMAC), MACs based on block ciphers (DAA, CMAC).
**Digital Signatures:** Digital signatures, Elgamal digital signature scheme, elliptic curve digital signature algorithm, RSA-PSS digital signature algorithm.

**UNIT - V: Network and Internet Security: Transport-Level Security:** Web security considerations, transport level security, HTTPS, SSH.
**IP Security:** IP security overview, IP security policy, encapsulating security payload, authentication header protocol.
**Electronic Mail Security:** Internet mail security, email format, email threats and comprehensive email security, S/MIME, PGP.

**Text Books:**
1. William Stallings, Cryptography and Network Security – Principles and Practice, 7th Edition, Pearson Education, 2017.
2. Behrouz A. Forouzan, Debdeep Mukhopadhyay, Cryptography and Network Security, 3rd Edition, McGraw Hill, 2015.

**Reference Books:**
1. Atul Kahate, Cryptography and Network Security, 3rd Edition, McGraw Hill.
2. Wade Trappe, Lawrence C. Washington, Introduction to Cryptography with Coding Theory, Pearson.
3. Wenbo Mao, Modern Cryptography: Theory and Practice, Pearson.

| II SEM | Secure coding<br>**(Program Elective–III & Program Elective–IV)** | Course Code:<br>V25CTL19 | **L**<br>3 | **T**<br>0 | **P**<br>0 | **C**<br>3 |
|---|---|---|---|---|---|---|

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain secure systems, principles, and common security attacks. **(K2)**
**CO2:** Demonstrate secure coding practices in C programs. **(K3)**
**CO3:** Apply secure coding techniques in C++ and Java. **(K3)**
**CO4:** Classify database and web-related vulnerabilities and remedies. **(K2)**
**CO5:** Explain requirements and practices for secure software engineering. **(K2)**

**UNIT - I: Introduction:** Need for secure systems, proactive security development process, security principles to live by, threat modelling.

**UNIT - II: Secure Coding in C:** Character strings – string manipulation errors, string vulnerabilities and exploits, mitigation strategies for strings. Pointers – mitigation strategies in pointer-based vulnerabilities. Buffer overflow–based vulnerabilities.

**UNIT - III: Secure Coding in C++ and Java:** Dynamic memory management, common errors in dynamic memory management, memory managers, double-free vulnerabilities, integer security, mitigation strategies.

**UNIT - IV: Database and Web-Specific Input Issues:** Quoting the input, use of stored procedures, building SQL statements securely, XSS-related attacks and remedies.

**UNIT - V: Software Security Engineering:** Requirements engineering for secure software, misuse and abuse cases, SQUARE process model, software security practices and knowledge for architecture and design.

**Text Book:**

**1.** Writing Secure Code, 2nd Edition, Michael Howard, David LeBlanc, Microsoft Press, 2003

**Reference Books:**

1. Secure Coding in C and C++, Robert C. Sea cord, 2nd edition, Pearson Education, 2013
2. Software Security Engineering: A guide for Project Managers, 1st ed, Julia H. Allen, Sean J.Barnum, Robert J. Ellison, Gary McGraw, Nancy R. Mead, Addison-Wesley Professional,2008

| II SEM | DevOps<br>**(Program Elective–III & Program Elective–IV)** | Course Code:<br>V25CTL20 | **L**<br>3 | **T**<br>0 | **P**<br>0 | **C**<br>3 |
|---|---|---|---|---|---|---|

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain the basic concepts and tools of DevOps and how it helps in software development.**(K2)**
**CO2:** Use Git for managing source code and apply tools like SonarQube to check code quality. **(K3)**
**CO3:** Operate build automation and continuous integration using Jenkins. **(K3)**
**CO4:** Experiment with continuous delivery and deployment using Docker and testing tools like Selenium.**(K3)**
**CO5:** Apply tools like Ansible, Kubernetes, Puppet, and Chef to manage and deploy applications easily. **(K3)**

**UNIT-I: Introduction to DevOps:** Introduction to SDLC, Agile Model, Introduction to DevOps, DevOps Features, DevOps Architecture, DevOps Lifecycle, Understanding Workflow and principles, Introduction to DevOps tools, Build Automation, Delivery Automation, Understanding Code Quality, Automation of CI/CD, Release management, Scrum, Kanban, delivery pipeline, bottlenecks, examples.

**UNIT-II: Source Code Management (GIT):** The need for source code control, the history of source code management, roles and code, source code management system and migrations, what is version control and GIT, GIT installation, GIT features, GIT workflow, working with remote repository, GIT commands, GIT branching, GIT staging and collaboration.
**Unit Testing – Code Coverage:** JUnit, nUnit & Code Coverage with SonarQube, SonarQube – Code Quality Analysis.

**UNIT-III: Build Automation** – Continuous Integration (CI): Build Automation, what is CI, why CI is required, CI tools, introduction to Jenkins (with architecture), Jenkins workflow, Jenkins master-slave architecture, Jenkins Pipelines, pipeline basics – Jenkins master, node, agent, and executor freestyle projects & pipelines, Jenkins for continuous integration, create and manage builds, user management in Jenkins, schedule builds, launch builds on slave nodes.

**UNIT-IV: Continuous Delivery:** Importance of continuous delivery, continuous deployment CD flow, containerization with Docker: introduction to Docker, Docker installation, Docker commands, images & containers, Docker file, running containers, working with containers and publish to Docker Hub. **Testing Tools:** Introduction to Selenium and its features, JavaScript testing.

**UNIT-V: Configuration Management – ANSIBLE:** Introduction to Ansible, Ansible tasks, roles, Jinja2 templating, vaults, deployments using Ansible.
**Containerization Using Kubernetes (OpenShift):** Introduction to Kubernetes namespace & resources, CI/CD on OpenShift Container Platform, BC, DC & config maps, deploying apps on OpenShift container pods, introduction to Puppet master and Chef.

**Text Books:**


1. Joyner, Joseph. DevOps for Beginners: DevOps Software Development Method Guide for Software Developers and IT Professionals, 1st Edition, Mihails Konoplows, 2015.
2. Alisson Machado de Menezes. Hands-on DevOps with Linux, 1st Edition, BPB Publications, India, 2021.


**Reference Books:**


1.Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective,Addison Wesley; ISBN-10.
2. Gene Kim, Jez Humble, Patrick Debois, John Willis. The DevOps Handbook, 1st Edition, IT Revolution Press, 2016.
3. Verona, Joakim. Practical DevOps, 1st Edition, Packt Publishing, 2016.
4. Joakim Verona. Practical DevOps, 2nd Edition, Ingram Short Title; 2nd Edition (2018), ISBN-10: 1788392574.
5. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint, Wiley Publications, ISBN: 9788126579952.


**Web Resources:**

1.https://archive.nptel.ac.in/courses/106/104/106104220/
2.https://www.tutorialspoint.com/blockchain/index.htm

| II SEM | Web Application Security<br>**(Program Elective–III & Program Elective–IV)** | Course Code:<br>V25CTL21 | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|---|
| | | | 3 | 0 | 0 | 3 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Explain basic web security concepts like HTTP, encoding, and authentication.**(K2)**
**CO2:** Apply session and access control methods to protect against security problems. **(K3)**
**CO3:** Use techniques to find and fix attacks on databases and backend systems. **(K3)**
**CO4:** Explain logic flaws and XSS attacks, and apply methods to prevent them. **(K3)**
**CO5:** Use tools and scripts to test for vulnerabilities and prevent data leaks. **(K3)**

**UNIT-I:** Web application security, handling user access, input, and attacks, HTTP protocol, web functionalities, encoding schemes, mapping the application.
**Client-Side Controls:** Transmitting data via the client, capturing user data, capturing user data, authentication technologies, design flaws in authentication, implementation flaws in authentication, securing authentication

**UNIT-II: Attacks on Session and Access Control**
The need for state, weaknesses in token generation handling, hijacking liberal cookie scope securing session management generate strong tokens protect tokens throughout their life cycle log, monitor, and alert, common vulnerabilities, testing with different user accounts, limited access testing direct access to methods, testing controls over static resources.

**UNIT-III: Attacking data-stores and back end components**
Injecting into Interpreted Contexts, injecting into SQL, injecting into NoSQL, injecting into XPath, injecting into LDAP, Injecting OS Commands, Manipulating File Paths, Injecting into XML Interpreters, Injecting into Back-end HTTP Requests, Injecting into Mail Services

**UNIT-IV: Attacking application logic and users**
The Nature of Logic Flaws, Real-World Logic Flaws, Avoiding Logic Flaws, Varieties of XSS, XSS Attacks in Action, Finding and Exploiting XSS Vulnerabilities, Preventing XSS Attacks, Inducing User Actions, Capturing Data Cross-Domain, The Same-Origin Policy Revisited, Other Client-Side Injection Attacks, Local Privacy Attacks, Attacking the Browser, Exploiting Non-HTTP Services, Exploiting Browser Bugs

**UNIT-V: Automating customized attacks and exploiting information disclosure**
Uses for Customized Automation, Enumerating Valid Identifiers, The Basic Approach, Detecting Hits, Scripting the Attack, JAttack, harvesting useful data, fuzzing for common vulnerabilities, burp intruder, barriers to automation, session-handling mechanisms, CAPTCHA controls, exploiting error messages, using public information, gathering published information, using inference, preventing information leakage.

**Text Book:**

1. Dafydd Stuttard, Marcus Pinto, The Web Application Hacker's Handbook, Wiley Publishing Inc.

**References:**

1. Improving Web Application Security: Threats and Countermeasures, Microsoft Corporation.
2. Bryan Sullivan, Vincent Liu, Web Application Security: A Beginner's Guide, McGraw Hill Publishers.
3. Ron Lepofsky, The Manager's Guide to Web Application Security, Apress Publishers.

| II SEM | Big Data Analytics<br>**(Program Elective–III & Program Elective–IV)** | Course Code:<br>V25CTL22 | **L**<br>3 | **T**<br>0 | **P**<br>0 | **C**<br>3 |
|---|---|---|---|---|---|---|

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Describe the concepts, trends, and industry uses of big data, including Hadoop, cloud computing, and analytics applications. **(K2)**

**CO2:** Apply data modeling, replication, and consistency techniques in NoSQL databases using tools like Cassandra. **(K3)**

**CO3:** Apply data processing methods in Hadoop and Hive for querying and optimization. **(K3)**

**CO4:** Apply data transformations, joins, and debugging techniques in Apache Spark for efficient processing. **(K3)**

**CO5:** Apply performance tuning and stream processing techniques in Spark to handle event-time data and structured streaming. **(K3)**

**UNIT-I:** What is big data, why big data, 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:** 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, Working with Cassandra ,Table creation, loading and reading data

**UNIT-III:** Data formats, analyzing data with Hadoop, scaling out, Architecture of Hadoop distributed file system (HDFS), fault tolerance ,with data replication, High availability, Data locality , Map Reduce Architecture, Process flow, Java interface, data flow, Hadoop I/O, data integrity, compression, serialization. Introduction to Hive, data types and file formats, HiveQL data definition, HiveQL data manipulation, Logical joins, Window functions, Optimization, Table partitioning, Bucketing, Indexing, Join strategies.

**UNIT-IV:** Apache spark- Advantages over Hadoop, lazy evaluation, In memory processing, DAG, Spark context, Spark Session, RDD, Transformations- Narrow and Wide, Actions, Data frames ,RDD to Data frames, Catalyst optimizer, Data Frame Transformations, Working with Dates and Timestamps, Working with Nulls in Data, Working with Complex Types, Working with JSON, Grouping, Window Functions, Joins, Data Sources, Broadcast Variables, Accumulators, Deploying Spark- On-Premises Cluster Deployments, Cluster Managers- Standalone Mode, Spark on YARN , Spark Logs, The Spark UI- Spark UI History Server, Debugging and Spark First Aid

**UNIT-V:** Spark-Performance Tuning, Stream Processing Fundamentals, Event-Time and State full Processing - Event Time, State full Processing, Windows on Event Time- Tumbling Windows, Handling Late Data with Watermarks, Dropping Duplicates in a Stream, Structured Streaming Basics - Core Concepts, Structured Streaming in Action, Transformations on Streams, Input and Output

**Text Books:**


1. Big Data, Big Analytics: Emerging, Michael Minnelli, Michelle Chambers, and Ambiga Dhiraj, 1st edition ,2013
2. SPARK: The Definitive Guide, Bill Chambers &MateiZaharia, O'Reilley, 2018-first Edition.
3. Business Intelligence and Analytic Trends for Today's Businesses", Wiley, First edition-2013.
4. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World Polyglot Persistence", Addison-Wesley Professional, 2012
5. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012


**Reference Books:**


1. "Hadoop Operations", O'Reilley, Eric Sammer, First Edition -2012.
2. "Programming Hive", O'Reilley, E. Capriolo, D. Wampler, and J. Rutherglen, 2012.
3. "HBase: The Definitive Guide", O'Reilley, Lars George, September 2011: First Edition..
4. "Cassandra: The Definitive Guide", O'Reilley, Eben Hewitt, 2010.
5. "Programming Pig", O'Reilley, Alan Gates, October 2011: First Edition.

| II SEM | Machine Learning Lab | Course Code: V25CTL03 | L | T | P | C |
|---|---|---|---|---|---|---|
| | | | 0 | 1 | 2 | 2 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:**Explain statistical measures and data preprocessing techniques.**(K2)**
**CO2:**Apply classification and regression algorithms on datasets. **(K3)**
**CO3:**Implement clustering algorithms and evaluate their results. **(K3)**
**CO4:**Analyze the performance of machine learning models using tuning and error measures. **(K4)**

**List of Experiments**

1. Compute Central Tendency Measures: Mean, Median, Mode. Measure of Dispersion: Variance, Standard Deviation.

2. Apply the following Pre-processing techniques for a given dataset.
   a) Attribute selection b) Handling Missing Values
   c) Discretization d) Elimination of Outliers

3. Apply KNN algorithm for classification and regression.

4. Demonstrate decision tree algorithm for a classification problem and perform parameter tuning for better results.

5. Demonstrate decision tree algorithm for a regression problem.

6. Apply Random Forest algorithm for classification and regression.

7. Demonstrate Naïve Bayes Classification algorithm.

8. Apply Support Vector algorithm for classification.

9. Demonstrate simple linear regression algorithm for a regression problem.

10. Apply Logistic regression algorithm for a classification problem.

11. Demonstrate Multi-layer Perceptron algorithm for a classification problem.

12. Develop the K-means algorithm and apply it to the data you selected. Evaluate performance by measuring the sum of the Euclidean distance of each example from its class center. Test the performance of the algorithm as a function of the parameters K.

13. Demonstrate the use of Fuzzy C-Means Clustering.

14. Demonstrate the use of Expectation Maximization based clustering algorithm.

| II SEM | OS and Unix Lab | Course Code: V25CTL04 | L | T | P | C |
|---|---|---|---|---|---|---|
| | | | 0 | 1 | 2 | 2 |

**Syllabus Details**

**Course Outcomes: After completion of course, students would be able to**

**CO1:** Apply Unix/Linux commands, file structures, and environment settings. **(K3)**

**CO2:** Apply system calls to write C programs for file and process management. **(K3)**

**CO3:** Analyze scheduling, memory management, and deadlock problems. **(K4)**

**CO4:** Analyze and build multi-threaded programs and memory solutions. **(K4)**

**List of Experiments**

1. a) Describe Unix/Linux general-purpose utility commands such as man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown.

   b) Describe the vi editor.
   c) Describe the Bash shell, Bourne shell, and C shell in the Unix/Linux operating system.
   d) Describe the Unix/Linux file system (tree structure).
   e) Describe .bashrc, /etc/bashrc, and Environment settings.

2. Develop a C program that copies a file using standard I/O and system calls.

3. Develop a C program that emulates the ls –l command.

4. Develop a C program to execute two commands concurrently using a command pipe (e.g. ls -l | sort).

5. Apply CPU scheduling algorithms such as Round Robin, SJF, FCFS, and Priority.

6. Operate memory management functions (`fork()`, `wait()`, `exec()`, and `exit()`) using system calls.

7. Construct simulations of multiprogramming with fixed (MFT) and variable (MVT) numbers of tasks.

8. Solve the Banker's algorithm for deadlock avoidance.

9. Demonstrate the Banker's algorithm for deadlock prevention.

10. Calculate file allocation strategies such as sequenced, indexed, and linked methods.

11. Employ shared memory communication between two processes using a C program.

12. Experiment with page replacement algorithms like FIFO, LRU, and LFU.

13. Demonstrate the producer-consumer problem using semaphores in C.

14. Construct a thread using the pthreads library and execute its function.

15. Operate concurrent execution of threads using the pthreads library.