# ACADEMIC REGULATIONS (BH25)
# COURSE STRUCTURE
# AND
# DETAILED SYLLABUS

# COMPUTER SCIENCE AND ENGINEERING

for

M.Tech. Two Year Degree Course

(Applicable for the batches admitted from A.Y. 2025-26 onwards)

**BVRIT HYDERABAD College of Engineering for Women**
(UGC Autonomous Institution | Approved by AICTE | Affiliated to JNTUH)
(NAAC Accredited – A Grade | NBA Accredited B.Tech. (EEE, ECE, CSE))
Bachupally, Hyderabad -500 090
**www.bvrithyderabad.edu.in**

# BVRIT HYDERABAD College of Engineering for Women

## VISION

To emerge as the best among the institutes of technology and research in the country dedicated to the cause of promoting quality technical education.

## MISSION

**At BVRITH, we strive to**

- Achieve academic excellence through innovative learning practices.
- Enhance intellectual ability and technical competency for a successful career.
- Encourage research and innovation.
- Nurture students towards holistic development with emphasis on leadership skills, life skills and human values.

## CORE VALUES

1. Holistic Development
2. Excellence in Education
3. Women Empowerment
4. Integrity
5. Social Responsibility
6. Accountability and Transparency
7. Freedom of Expression

## Program Educational Objectives

**PEO 1:** Exhibit technical excellence and innovation in the multidisciplinary realms of computer science through analysis, architectural design, critical thinking, and the implementation of scalable computing systems.

**PEO 2:** Apply advanced domain expertise and computational proficiency in computer science and engineering to build successful, enduring careers as Software Engineers, Technical Architects, System Analysts, or Researchers.

**PEO 3:** Develop entrepreneurial competence and technical leadership to create viable computational solutions and products for business and societal needs, fostering self-employment opportunities.

**PEO 4:** Instill a commitment to continuous, self-directed lifelong learning to stay abreast of emerging computing technologies, modern software frameworks, and evolving industry standards.

**PEO 5:** Inculcate high ethical standards, professional integrity, human values, and leadership skills for holistic personal development and meaningful social impact.

## Program Outcomes

**PO 1:** An ability to independently carry out research/investigation and development work to solve practical problems.

**PO 2:** An ability to write and present a substantial technical report/document.

**PO 3:** Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program.

**PO4:** An ability to Plan, design, and deploy state-of-the-art software systems and digital solutions to address industry, business, and societal needs.

**PO5:** Apply advanced computer science concepts, algorithms, software architecture, and modern computing tools to develop proficient computing professionals and technical leaders.

**PO6:** Demonstrate self-directed learning, adaptability, and continuous professional development to respond to emerging technologies and assume leadership roles.

## Academic Regulations of M. Tech (Regular/Full Time) Programmes, (BH25) (CBCS)

**(Effective for the students admitted into I Year from the AY 2025-26 and onwards)**

**1.0** Jawaharlal Nehru Technological University Hyderabad (JNTUH) offers **Two** Years (**Four** Semesters) full-time Master of Technology (M.Tech.) Degree programmes, under Choice Based Credit System (CBCS) at its non-autonomous affiliated colleges in different branches of Engineering and Technology with different specializations.

### 2.0 Eligibility for Admissions

**2.1** Admission to the M.Tech. programme shall be made subject to eligibility, qualification and specializations prescribed by the University from time to time, for each specialization under each M.Tech. programme.

**2.2** Admission to the post graduate programme shall be made on the basis of either the merit rank or Percentile obtained by the qualified student in the GATE Examination/ the merit rank obtained by the qualified student in an entrance test conducted by Telangana Government (PGECET) for M.Tech. programmes/ an entrance test conducted by JNTUH/ on the basis of any other exams approved by the University, subject to reservations as laid down by the Govt. from time to time.

**2.3** The medium of instructions for all PG Programmes will be **ENGLISH** only.

### 3.0 M.Tech. Programme Structure

**3.1** The M.Tech. Programs of JNTUH are of Semester pattern, consisting of **Two** academic years, each academic year having **Two** Semesters (Odd and Even Semesters).

**3.2** The two-year M. Tech. program consists of **68** credits and the student has to register for all **68** credits and earn all **68** credits for the award of M. Tech. degree.

**3.3** The student shall not take more than four academic years to fulfill all the academic requirements for the award of M. Tech. degree from the date of commencement of first year first semester, failing which the student shall forfeit the seat in M. Tech. programme.

**3.4** **UGC/AICTE** specified definitions/descriptions are adopted appropriately for various terms and abbreviations used in these PG academic regulations, as listed below:

#### 3.4.1 Semester Scheme

There shall be a minimum of 15 weeks of instruction, excluding the mid-term and semester-end exams. Around 15 instruction hours, 30 instruction hours and 45 hours of learning need to be followed per one credit of theory course, practical course and project/field-based learning respectively. In each semester, there shall be 'Continuous Internal Evaluation (CIE)' and 'Semester End Examination (SEE)' under Choice Based Credit System (CBCS). The curriculum/course structure suggested by AICTE is followed as a reference document.

#### 3.4.2 Credit Courses

All courses are to be registered by the student in a semester to earn credits which shall be assigned to each course in an L: T: P: $C^3$ (Lecture Periods: Tutorial Periods: Practical Periods: Credits) structure based on the following general pattern:

- One credit for one hour/week/semester for theory/lecture (L) courses or tutorials (T)
- One credit for two hours/ week/semester for laboratory/ practical (P) courses
- One credit is allocated for three hours per week in a semester for Project/Mini-Project session.

### 3.4.3 Course Classification

All courses offered for the Post-Graduate M.Tech. Degree program are broadly classified as follows. The University has followed in general the guidelines issued by AICTE/UGC.

| S. No. | Broad Course Classification | Course Group/ Category | Course Description |
|---|---|---|---|
| 1 | Core Courses (CoC) | PC- Professional Core | Includes courses related to the parent discipline/department/ branch of Engineering |
| | | Dissertation | M. Tech. Project or PG Project or Major Project |
| | | Mini Project with Seminar | Seminar based on core contents related to Parent Discipline/ Department/ Branch of Engineering |
| 2 | Elective Courses (EtE) | PE - Professional Electives | Includes elective courses related to the parent discipline/department/branch of Engineering |
| | | OE - Open Electives | Elective courses which include inter-disciplinary courses or courses in an area outside the parent discipline/department/ branch of Engineering |
| 3 | Audit Courses | -- | Non-Credit Audit Courses |

## 4.0 Course Registration

**4.1** A Faculty Advisor or Counselor shall be assigned to each specialization, who will advise on the Post Graduate Programme, its Course Structure and Curriculum, Choices/Options for Courses, based on his competence, progress, pre-requisites and interest.

**4.2** The on-line Registration Requests for any current semester shall be completed before the commencement of SEEs (Semester End Examinations) of the preceding semester.

**4.3** A Student can apply for on-line Registration, only after obtaining the written approval from his Faculty Advisor, which should be submitted to the College Academic Section through the Head of Department (a copy of it being retained with Head of Department, Faculty Advisor and the Student).

**4.4** If the Student submits ambiguous choices or multiple options or erroneous entries during on-line Registration for the Course(s) under a given/ specified Course Group/ Category

as listed in the Course Structure, only the first mentioned Course in that Category will be taken into consideration.

**4.5** Course Options exercised through on-line Registration are final and cannot be changed. further, alternate choices will not be considered. However, if the Course that has already been listed for Registration by the University in a Semester could not be offered due to unforeseen or unexpected reasons, then the Student will be allowed to have alternate choice either for a new Course, if it is offered, or for another existing Course (subject to availability of seats). Such alternate arrangements will be made by the Head of Department, with due notification and time-framed schedule, within the first week from the commencement of Class-work for that Semester.

**5.0 Attendance Requirements**

Attendance is calculated separately for each course.

**5.1** Attendance in all classes (Lectures/Laboratories) is compulsory. The minimum required attendance in each theory course (***also Audit Courses***) including the attendance of mid-term examination / Laboratory etc. is 75%. Two periods of attendance for each theory course shall be considered, if the student appears for the mid-term examination of that course. ***This attendance should also be included in the attendance uploaded every fortnight in the University Website. The attendance of Audit Courses should be uploaded separately to the University.*** A student shall not be permitted to appear for the Semester End Examinations (SEE), if his attendance is less than 75%.

**5.2** A student's Seminar report and presentation on Mini Project shall be eligible for evaluation, only if he ensures a minimum of 75% of his attendance in Seminar presentation classes on Mini Project during that Semester.

**5.3** **Condoning of shortage of attendance** up to a maximum of 10% (considering the days of attendance in sports, games, NCC, NSS activities and Medical grounds) in each course (Theory/Lab/Mini Project with Seminar) of a semester shall be granted by the College Academic Committee on genuine reasons.

**5.4** A prescribed fee per course shall be payable for condoning shortage of attendance after getting the approval of College Academic Committee for the same. The College Academic Committee shall maintain relevant documents along with the request from the student.

**5.5** Shortage of Attendance below 65% in any course shall in **no case be condoned.**

**5.6** A Student, whose shortage of attendance is not condoned in any course(s) (Theory/Lab/Mini Project with Seminar) in any Semester, is considered as 'Detained in that course(s), and is not eligible to write Semester End Examination(s) of such course(s), in that Semester; and he/she has to seek re-registration for those course(s) in subsequent Semesters, and attend the same as and when offered.

**5.7** A student fulfills the attendance requirement in the present semester, shall not be eligible for readmission into the same class.

**5.8** **a)** A student shall put in a minimum required attendance in at least three theory courses (excluding Audit course) in first Year I semester for promotion to first Year II Semester.

**b)** A student shall put in a minimum required attendance in at least three theory courses (excluding *Audit* course**)** in first Year II semester for promotion to second Year I Semester.

**6.0** **Academic Requirements**

The following academic requirements must be satisfied, in addition to the attendance requirements mentioned in clause no. 5. The performance of the candidate in each semester shall be evaluated course-wise, with a maximum of 100 marks per course (theory / practical), based on Continuous Internal Evaluation and Semester End Examination.

**6.1** A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each course, if he secures not less than:

- 40% of Marks (24 out of 60 marks) in the Semester End Examination;
- 40% of Marks in the internal examinations (16 out of 40 marks allotted for CIE); and
- A minimum of 50% of marks in the sum total of CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together; in terms of Letter Grades this implies securing **'B'** Grade or above in a course.

**6.2** A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to Mini Project with seminar, if student secures not less than 50% marks (i.e. 50 out of 100 allotted marks). The student would be treated as failed, if student (i) does not submit a seminar report on Mini Project or does not make a presentation of the same before the evaluation committee as per schedule or (ii) secures less than 50% marks in Mini Project with seminar evaluation. The failed student shall reappear for the above evaluation when the notification for supplementary examination is issued.

**6.3** A student shall register for all courses for total of **68** credits as specified and listed in the course structure for the chosen specialization, put in the required attendance and fulfill the academic requirements for securing **68** credits obtaining a minimum of **'B'** Grade or above in each course, and shall ***pass all the Audit Courses*** to complete the M.Tech. Programme successfully.

**Note: (1) The SGPA will be computed and printed on the marks memo only if the candidate passes in all the courses offered and gets minimum B grade in all the courses.**

**(2) CGPA is calculated only when the candidate passes in all the courses offered in all the semesters**

**6.4** Letter Grades obtained in all those courses covering the above specified **68** credits alone shall be considered for the calculation of final CGPA, which will be indicated in the consolidated grade memo.

**6.5** When a student is detained due to shortage of attendance in any course(s) in any semester, shall not be permitted to write the Semester End Examinations. However, he is eligible for re-registration of such course(s) in the subsequent semester(s), as and when next offered, with the academic regulations of the batch into which he is re-registered, by paying the prescribed fees per course. In all these re-registration cases, the student shall have to secure a fresh set of internal marks and Semester End Examination marks for performance evaluation in such course(s), and SGPA/CGPA calculations.

**6.6** A student eligible to appear for the Semester End Examination in any course, but absent from it or failed (failing to secure 'B' Grade or above), may reappear for that course at the supplementary examination as and when conducted. In such cases, his Internal Marks assessed earlier for that course will be carried over, and added to the marks secured in the supplementary semester end examination, for the purpose of evaluating his performance in that course.

**6.7** A Student who fails to earn **68** credits as per the specified course structure, and as indicated above, within **four** academic years from the date of commencement of his first year first semester, shall forfeit his seat in M. Tech. programme and his admission **shall stand cancelled.**

## 7.0 Evaluation - Distribution and Weightage of Marks

The performance of a student in each semester shall be evaluated course- wise (irrespective of credits assigned) for a maximum of 100 marks.

**7.1** The performance of a student in every course (including practicals and Project) will be evaluated for 100 marks each, with 40 marks allotted for CIE (Continuous Internal Evaluation) and 60 marks for SEE (Semester End-Examination). The Continuous Internal Evaluation for theory courses shall be made based on the average of the marks secured in the two Mid-Term Examinations conducted, first Mid-Term examinations in the middle of the Semester and second Mid-Term examinations during the last week of instruction.

**7.2** In CIE, for theory courses, during a semester, there shall be two mid-term examinations. Each Mid-Term examination consists of two parts i) **Part – A** for 10 marks, ii) **Part – B** for 20 marks with a total duration of two hours as follows:

1. Mid-Term Examination for 30 marks:
    a. Part - A: Objective/quiz paper for 10 marks.
    b. Part – B: Descriptive paper for 20 marks.

The objective/quiz paper is set with multiple choice, fill-in the blanks and match the following type of questions for a total of 10 marks. The descriptive paper shall contain 6 full questions out of which, the student has to answer 4 questions, each carrying 5 marks.

The average of the two Mid Term Examinations shall be taken as the final marks for Mid Term Examination for 30 marks.

The remaining 10 marks of Continuous Internal Assessment (out of 40) are distributed as:

2. 5 marks for Assignment. (Average of 2 Assignments each for 5 marks)
3. Course Viva-Voce/PPT/Poster Presentation/ Case Study on a topic in the concerned course for 5 marks.

While the first mid-term examination shall be conducted on 50% of the syllabus, the second mid-term examination shall be conducted on the remaining 50% of the syllabus.

Five (5) marks are allocated for assignments (as specified by the course teacher concerned). The first assignment should be submitted before the conduct of the first mid-term examination, and the second assignment should be submitted before the conduct of the second mid-term examination. The average of the two assignments shall be taken as the final marks for assignment.

Course Viva-Voce/PPT/Poster Presentation/ Case Study on a topic in the concerned course for five marks before II Mid-Term Examination.

- The Student, in each course, shall have to earn 40% of marks (i.e. 16 marks out of 40 marks) in CIE, 40% of marks (i.e. 24 marks out of 60) in SEE and Overall 50% of marks (i.e. 50 marks out of 100 marks) both CIE and SEE marks taking together.

*The student is eligible to write Semester End Examination of the concerned course, if the student scores ≥ 40% (16 marks) of 40 Continuous Internal Examination (CIE) marks.*

*In case, the student appears for Semester End Examination (SEE) of the concerned course but not scored minimum 40% of CIE marks (16 marks out of 40 internal marks), his performance in that course in SEE shall stand cancelled inspite of appearing the SEE.*

The details of the end semester question paper pattern are explained in the next clause:

**7.3** The Semester End Examinations (SEE), for theory courses, will be conducted for 60 marks consisting of two parts viz. i) **Part- A** for 10 marks, ii) **Part - B** for 50 marks.

- Part-A is a compulsory question which consists of ten sub-questions with uniform coverage from all units carrying equal marks.
- Part-B consists of five questions (numbered from 2 to 6) carrying 10 marks each. Each of these questions is from each unit and may contain sub-questions. For each question there will be an "either" "or" choice, which means that there will be two questions from each unit and the student should answer either of the two questions.
- The duration of Semester End Examination is three hours.

**7.4** For practical courses there shall be a Continuous Internal Evaluation (CIE) during the semester for 40 marks and 60 marks for semester end examination. Out of the 40 marks for internal evaluation:

1. A write-up on day-to-day experiment in the laboratory (in terms of aim, components/procedure, expected outcome) which shall be evaluated for 10 marks
2. 10 marks for viva-voce (or) tutorial (or) case study (or) application (or) poster presentation of the course concerned.
3. Internal practical examination conducted by the laboratory teacher concerned shall be evaluated for 10 marks.
4. The remaining 10 marks are for Laboratory Project, which consists of the Design (or) Software / Hardware Model Presentation (or) App Development (or) Prototype Presentation submission which shall be evaluated after completion of laboratory course and before semester end practical examination.

In the Semester End Examination, held for three hours, total 60 marks are divided and allocated as shown below:

1. 10 marks for write-up
2. 15 for experiment/program
3. 15 for evaluation of results
4. 10 marks for presentation on another experiment/program in the same laboratory course and
5. 10 marks for viva-voce on concerned laboratory course.

- The Student, in each course, shall have to earn 40% of marks (i.e. 16 marks out of 40 marks) in CIE, 40% of marks (i.e. 24 marks out of 60) in SEE and Overall 50% of marks (i.e. 50 marks out of 100 marks) both CIE and SEE marks taking together.

*The student is eligible to write Semester End Examination of the concerned course, if the student scores ≥ 40% (16 marks) of 40 Continuous Internal Examination (CIE) marks.*

*In case, the student appears for Semester End Examination (SEE) of the concerned course but not scored minimum 40% of CIE marks (16 marks out of 40 internal marks), his performance in that course in SEE shall stand cancelled inspite of appearing the SEE.*

**7.5** For conducting laboratory end examinations of all PG Programmes, one internal examiner and one external examiner are to be appointed by the Principal of the College and this is to be informed to the Director of University Examinations within two weeks, before commencement of the lab end examinations. The external examiner should be selected from outside the College concerned but within the cluster. No external examiner should be appointed from any other College in the same cluster/any other cluster which is run by the same Management.

**7.6** There shall be Mini Project with Seminar during I year II semester for internal evaluation of 100 marks. The Departmental Academic Committee (DAC) will review the progress of the mini project during the seminar presentations and evaluate the same for 50 marks. Mini Project Viva Voce will be evaluated by the DAC for another 50 marks before the semester end examinations. Student shall carryout the mini project in consultation with the mini project supervisor which may include critically reviewing the literature, project implementation and submit it to the department in the form of a report and shall make an oral presentation before the DAC consisting of Head of the Department, Mini Project supervisor and two other senior faculty members of the department. The student has to secure a minimum of 50% of marks in i) seminar presentation and ii) mini project viva voce, to be declared successful. If he fails to obtain the minimum marks, he has to reappear for the same as and when scheduled.

**7.7** Every candidate shall be required to submit a dissertation on a topic approved by the Dissertation Review Committee.

**7.8** The M.Tech. Dissertation shall be prepared in the structure prescribed by the University, adhering to the style files and formatting guidelines. To facilitate this process, each institution will organize a brief orientation session for the entire class/section at the beginning of the final semester, guiding the students on the required structure and formatting of the dissertation.

**7.9** A Dissertation Review Committee (DRC) shall be constituted with the Head of the Department as Chairperson, Dissertation Supervisor and one senior faculty member of the Department offering the M.Tech. programme.

**7.10** Registration of Dissertation Work: A candidate is permitted to register for the Dissertation Work after satisfying the attendance requirement in all the courses, both theory and laboratory.

**7.11** After satisfying the previous clause, a candidate must present in ***Dissertation Work Review - I***, in consultation with his Dissertation Supervisor, the title, objective and plan of action of his Dissertation work to the Dissertation Review Committee (DRC) for approval ***within four weeks*** from the commencement of **Second year First Semester**. Only after obtaining the approval of the DRC can the student initiate the Dissertation work.

**7.12** If a candidate wishes to change his supervisor or topic of the Dissertation, he can do so with the approval of the DRC. However, the DRC shall examine whether or not the change of topic/supervisor leads to a major change of his initial plans of Dissertation 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.

**7.13** A candidate shall submit his Dissertation progress report in two stages at least with a gap of **three** months between them.

**7.14** The work on the Dissertation shall be initiated at the beginning of the II year and the duration of the Dissertation is two semesters. A candidate is permitted to submit Dissertation Thesis only after successful completion of all theory and practical courses with the approval of DRC ***not earlier than 40 weeks*** from the date of approval of the Dissertation work. For the approval of DRC, the candidate shall submit the draft copy of thesis to the Head of the Department and make an oral presentation before the DRC.

**7.15** ***The Dissertation Work Review - II*** in II Year I Semester carries 100 internal marks. Evaluation should be done by the DRC for 50 marks and the Supervisor will evaluate the work for the other 50 marks. The Supervisor and DRC will examine the Problem Definition, Objectives, Scope of Work, Literature Survey in the same domain and progress of the Dissertation Work. A candidate has to secure a minimum of 50% of marks to be declared successful in Dissertation Work Review - II. If he fails to obtain the minimum required marks, he has to reappear for Dissertation Work Review - II as and when conducted.

**7.16** ***The Dissertation Work Review - III*** in II Year II Sem. carries 100 internal marks. Evaluation should be done by the DRC for 50 marks and the Supervisor will evaluate it for the other 50 marks. The DRC will examine the overall progress of the Dissertation Work and decide whether or not the Dissertation is eligible for final submission. A candidate has to secure a minimum of 50% of marks to be declared successful in Dissertation Work Review - III. If he fails to obtain the required minimum marks, he has to reappear for Dissertation Work Review - III as and when conducted. For Dissertation Evaluation (Viva Voce) in II Year II Semester there are external marks of 100 and it is evaluated by the external examiner. The candidate has to secure a minimum of 50% marks in Dissertation Evaluation (Viva-Voce) examination.

**7.17** Dissertation Work Reviews - II and III shall be conducted in phase I (Regular) and Phase II (Supplementary). Phase II will be conducted only for unsuccessful students in Phase I. The unsuccessful students in Dissertation Work Review - II (Phase II) shall reappear for it at the time of Dissertation Work Review - III (Phase I). These students shall reappear for Dissertation Work Review- III in the next academic year at the time of Dissertation Work Review - II only after completion of Dissertation Work Review - II, and then Dissertation Work Review - III follows. The unsuccessful students in Dissertation Work Review - III (Phase II) shall reappear for Dissertation Work Review - III in the next academic year only at the time of Dissertation Work Review - II (Phase I).

**7.18** After approval from the DRC, a soft copy of the thesis should be submitted for Anti-Plagiarism check and the plagiarism report should be submitted to the University and be included in the final thesis. The Thesis will be accepted for submission, if the similarity index is less than **30%**. If the similarity index is more than the required percentage,

the student is advised to revise the thesis and re-submit the soft copy of the thesis after one month. The maximum number of re-submissions of thesis after plagiarism check is limited to ***TWO***. The candidate has to register for the Dissertation work and work for two semesters. After three attempts, the admission is liable to be cancelled.

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

**7.20** The thesis shall be adjudicated by an external examiner selected by the University. For this, the Principal of the College/School/Institute shall submit a panel of **three** examiners from among the list of experts in the relevant specialization as submitted by the supervisor concerned and Head of the Department.

**7.21** If the report of the external examiner is unsatisfactory, the candidate shall revise and resubmit the Thesis. If the report of the examiner is unsatisfactory again, the thesis shall be summarily rejected. Subsequent actions for such dissertations may be considered, only on the specific recommendations of the external examiner and /or Dissertation Review Committee. No further correspondence in this matter will be entertained, if there is no specific recommendation for resubmission.

**7.22** If the report of the examiner is satisfactory, the Head of the Department shall coordinate and decide for the conduct of Dissertation Viva-Voce examination. The Dissertation Viva-Voce examination shall be conducted by a board consisting of the Supervisor, Head of the Department and the external examiner who adjudicated the Thesis. The candidate has to secure a minimum of 50% of marks in Dissertation Evaluation (Viva-Voce) examination.

**7.23** If he fails to fulfill the requirements as specified in the above clause, he will reappear for the Dissertation Viva-Voce examination ***only after three months***. In the reappeared examination also, if he fails to fulfill the requirements, he will not be eligible for award of the degree, unless he is asked to revise and resubmit his Dissertation Work by the board within a specified time period.

**7.24** The Dissertation Viva-Voce External examination marks must be submitted to the University on the same day of the examination.

**7.25** For Audit courses, a student has to secure 40 marks out of 100 marks (i.e. 40% of the marks allotted) in the continuous internal evaluation for passing the course. These marks should also be uploaded along with the internal marks of other courses.

**7.26** No marks or letter grades shall be allotted for Audit Courses. Only Pass/Fail shall be indicated in Grade Card.

## 8.0 Re-Admission/Re-Registration

### 8.1 Re-Admission for Discontinued Student

A student, who has discontinued the M. Tech. degree programme due to any reason whatsoever, may be considered for '**readmission**' into the same degree programme (with the same specialization) with the academic regulations of the batch into which he gets readmitted, with prior permission from the authorities concerned.

**8.2** If a student is detained in a course (s) due to shortage of attendance in any semester, he may be permitted to **re-register** for the same course(s) or the student may register in an equivalent course, If the same course is not available, as suggested by the Board of Studies of that department, in the subsequent semester(s), with the academic regulations of the batch into which he seeks re-registration, with prior permission from the authorities concerned.

**8.3** A candidate shall be given chance to re-register any number of courses, if the candidate failed in these courses due to securing less than 40% marks in CIE. A candidate must re-register for failed courses within four weeks of commencement of the class work, in the next academic year and secure the required minimum attendance. In the event of the student taking this chance, his Continuous Internal Evaluation (internal) marks and Semester End Examination marks obtained in the current semester only will be accepted, if he secures pass grade.

## 9.0 Examinations and Assessment - The Grading System

**9.1** Grades will be awarded to indicate the performance of each student in each Theory Course, or Lab/Practicals, or Mini Project with Seminar, Dissertation, etc., based on the percentage of marks obtained in CIE + SEE (Continuous Internal Evaluation + Semester End Examination, both taken together), and a corresponding Letter Grade shall be given.

**9.2** As a measure of the student's performance, a 10-point Absolute Grading System using the following Letter Grades (UGC Guidelines) and corresponding percentage of marks shall be followed:

| % of Marks Secured in a Course (Class Intervals) | Letter Grade (UGC Guidelines) | Grade Points |
|---|---|---|
| 90% and above (≥ 90%, ≤ 100%) | O (Outstanding) | 10 |
| Below 90% but not less than 80% (≥ 80%, <90%) | $A^+$ (Excellent) | 9 |
| Below 80% but not less than 70% (≥70%, <80%) | A (Very Good) | 8 |
| Below 70% but not less than 60% (≥60%, <70%) | $B^+$ (Good) | 7 |
| Below 60% but not less than 50% (≥50%, <60%) | B (above Average) | 6 |
| Below 50% (< 50%) | F (FAIL) | 0 |
| **Absent** | **Ab** | **0** |

**9.3** A student obtaining **'F'** Grade in any Course is deemed to have 'failed' and is required to reappear as 'Supplementary Candidate' for the Semester End Examination (SEE), as and when conducted. In such cases, his Internal Marks (CIE Marks) in those courses will remain as obtained earlier.

**9.4** If a student has not appeared for the examinations, '**Ab**' Grade will be allocated to him for any course and shall be considered 'failed' and will be required to reappear as 'Supplementary Candidate' for the Semester End Examination (SEE), as and when conducted.

**9.5** A Letter Grade does not imply any specific marks percentage; it is only the range of percentage of marks.

**9.6** In general, a student shall not be permitted to repeat any Course (s) only for the sake of 'Grade Improvement' or 'SGPA/ CGPA Improvement'.

**9.7** A student earns Grade Point (GP) in each Course, on the basis of the Letter Grade obtained by him in that Course. The corresponding 'Credit Points' (CP) are computed by multiplying the Grade Point with Credits for that particular Course.

**Credit Points (CP) = Grade Point (GP) x Credits …. For a Course**

**9.8** The student passes the Course only when he **gets GP □ 6 (B Grade or above).**

**9.9** The Semester Grade Point Average (SGPA) is calculated by dividing the Sum of Credit Points secured from all Courses registered in a Semester, by the total number of credits offered in that Semester. SGPA is rounded off to two decimal places. SGPA is thus computed as

$$\textbf{SGPA} = \{ \sum_{i=1}^{N} C_i G_i \} / \{ \sum_{i=1}^{N} C_i \} \textbf{ …. For each Semester}$$

where 'i' is the Course indicator index (taking into account all Courses in a Semester), 'N' is the no. of Courses offered in the Semester (as specifically required and listed under the Course Structure of the parent Department), $C_i$ is the no. of Credits allotted to the $i^{th}$ Course, and $G_i$ represents the Grade Points corresponding to the Letter Grade awarded for that $i^{th}$ Course.

**9.10** The Cumulative Grade Point Average (CGPA) is a measure of the overall cumulative performance of a student over all Semesters considered for registration. The CGPA is the ratio of the Total Credit Points secured by a student in all registered Courses in all Semesters, and the Total Number of Credits registered in all the Semesters. CGPA is rounded off to two decimal places. CGPA is thus computed from the I Year Second Semester onwards, at the end of each Semester, as per the formula

$$\textbf{CGPA} = \{ \sum_{j=1}^{M} C_j G_j \} / \{ \sum_{j=1}^{M} C_j \}$$

**(ie., up to and inclusive of S Semesters, S ≥ 2),**

where 'M' is the total no. of Courses (as specifically required and listed under the Course Structure of the parent Department) the Student has 'registered'. $C_j$ is the no. of Credits allotted to the $j^{th}$ Course, and $G_j$ represents the Grade Points (GP) corresponding to the Letter Grade awarded for that $j^{th}$ Course. After registration and completion of I Year I Semester however, the SGPA of that Semester itself may be taken as the CGPA, as there are no cumulative effects.

**Illustration of calculation of SGPA**

| Course | Credits | Letter Grade | Grade points | Credit Points |
|---|---|---|---|---|
| Course 1 | 4 | A | 8 | 4*8 = 32 |
| Course 2 | 4 | O | 10 | 4*10 = 40 |
| Course 3 | 4 | B | 6 | 4*6 = 24 |
| Course 4 | 3 | B | 6 | 3*6 = 18 |
| Course 5 | 3 | A+ | 9 | 3*9 = 27 |
| Course 6 | 3 | B | 6 | 3*6 = 18 |
| | 21 | | | 159 |

SGPA = 159/21 = 7.57

**Illustration of calculation of CGPA from SGPA**

| Semester | Credits | SGPA | Credits * SGPA |
|---|---|---|---|
| Semester I | 24 | 7 | 24*7 = 168 |
| Semester II | 24 | 6 | 24*6 = 144 |
| Semester III | 24 | 6.5 | 24*6.5 = 156 |
| Semester IV | 24 | 6 | 24*6 = 144 |
| | 96 | | 612 |

CGPA = 612/96 = 6.37

## 10.0 Award of Degree and Class

**10.1** If a student who registers for all the specified Courses as listed in the Course Structure, satisfies all the Course Requirements, and passes the examinations prescribed in the entire PG Programme, and secures the required number of **68** Credits (with CGPA ≥6.0), shall be declared to have 'qualified' for the award of the M.Tech. Degree in the chosen Branch of Engineering/Technology with the specialization that he was admitted into.

### 10.2 Award of Class

After a student has earned the requirements prescribed for the completion of the programme and is eligible for the award of M.Tech. Degree, he shall be placed in one of the following three classes based on the CGPA:

| Class Awarded | CGPA |
|---|---|
| First Class with Distinction | ≥ 7.50 |
| First Class | 6.50≤ CGPA < 7.50 |
| Second Class | 6.00≤ CGPA < 6.50 |

student with final CGPA (at the end of the **PGP) < 6.00** shall not be eligible for the Award of Degree.

### 11.0 Withholding of Results

If the student has not paid the dues, if any, to the University or if any case of indiscipline is pending against him, the result and degree of the student will be withheld and he will not be allowed into the next semester.

### 12.0 Conversion of CGPA into equivalent Percentage of Marks

The following formula shall be used for the conversion of CGPA into equivalent marks, whenever it is necessary

Percentage (%) of Marks = (Final CGPA – 0.5) x 10

### 13.0 Mapping with the Sustainable Development Goals

All the courses specified in the course structure of every programme are mapped with the one or more sustainable development goals.

### 14.0 General

**14.1** **Credit**: A unit by which the course work is measured. It determines the number of hours of instructions required per week. One credit is equivalent to one hour of teaching (lecture or tutorial) or two hours of practical work/field work per week.

**14.2** **Credit Point:** It is the product of grade point and number of credits for a course.

**14.3** Wherever the words “he”, “him”, “his”, occur in the regulations, they shall include “she”, “her”.

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

**14.5** In case of any doubt or ambiguity in the interpretation of the above rules, the decision of the University is final.

**14.6** 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.

**BVRIT HYDERABAD College of Engineering for Women**
(UGC Autonomous Institution | Approved by AICTE | Affiliated to JNTUH)
(NAAC Accredited – A Grade | NBA Accredited (EEE, ECE, CSE))
Bachupally, Hyderabad - 500090

## M. Tech. in COMPUTER SCIENCE AND ENGINEERING

## COURSE STRUCTURE (BH25 Regulations)
## Applicable from AY 2025-26 Batch

### I Year I Semester

| S No | Code | Title | L | T | P | Credits |
|---|---|---|---|---|---|---|
| **1** | **58101PC** | Mathematical Foundations of Computer Science | 3 | 0 | 0 | 3 |
| **2** | **58102PC** | Advanced Data Structures | 3 | 0 | 0 | 3 |
| **3** | | Professional Elective-I<br>1. Database Programming with PL/SQL (**58111PE**)<br>2. Deep Learning (**58112PE**)<br>3. Natural Language Processing (**58113PE**)<br>4. Advanced UNIX Programming (**58114PE**) | 3 | 0 | 0 | 3 |
| **4** | | Professional Elective-II<br>1. Applied Cryptography (**58121PE**)<br>2. Software Quality Engineering (**58122PE**)<br>3. Mining Massive Datasets (**58123PE**)<br>4. Agile Methodologies (**58124PE**) | 3 | 0 | 0 | 3 |
| **5** | **58103PC** | Advanced Data Structures Lab | 0 | 0 | 4 | 2 |
| **6** | | Professional Elective-I Lab<br>1. Database Programming with PL/SQL Lab (**58131PE**)<br>2. Deep Learning Lab (**58132PE**)<br>3. Natural Language Processing Lab (**58133PE**)<br>4. Advanced UNIX Programming Lab (**58134PE**) | 0 | 0 | 4 | 2 |
| **7** | **58104PC** | Research Methodology & IPR | 2 | 0 | 0 | 2 |
| **8** | | Audit Course- I | 2 | 0 | 0 | 0 |
| | | **Total** | **16** | **0** | **8** | **18** |

Professional Elective- I and Professional Elective- I Lab must be of same course.

**I Year II Semester**

| S No | Code | Title | L | T | P | Credits |
|---|---|---|---|---|---|---|
| **1** | **58201PC** | Advanced Algorithms | 3 | 0 | 0 | 3 |
| **2** | **58202PC** | Advanced Computer Architecture | 3 | 0 | 0 | 3 |
| **3** | | Professional Elective–III<br>1. Enterprise Cloud Concepts (**58211PE)**<br>2. Cyber Security (**58212PE)**<br>3. Parallel computing (**58213PE)**<br>4. Large Language Models (**58214PE)** | 3 | 0 | 0 | 3 |
| **4** | | Professional Elective–IV<br>1. Bioinformatics (**58221PE)**<br>2. Adhoc Sensor Networks (**58222PE)**<br>3. Robotic Process Automation (**58223PE)**<br>4. Generative AI (**58224PE)** | 3 | 0 | 0 | 3 |
| **5** | **58203PC** | Advanced Algorithms Lab | 0 | 0 | 4 | 2 |
| **6** | | Professional Elective-III Lab<br>1. Enterprise Cloud Concepts Lab **(58231PE)**<br>2. Cyber Security Lab **(58232PE)**<br>3. Parallel computing Lab **(58233PE)**<br>4. Large Language Models Lab **(58234PE)** | 0 | 0 | 4 | 2 |
| **7** | **58204CC** | Mini Project with Seminar | 0 | 0 | 4 | 2 |
| **8** | | Audit Course- II | 2 | 0 | 0 | 0 |
| | | **Total** | **14** | **0** | **12** | **18** |

Professional Elective- III and Professional Elective- III Lab must be of same course

## II Year I Semester

| S No | Code | Title | L | T | P | Credits |
|---|---|---|---|---|---|---|
| **1** | | Professional Elective-V<br>1. Digital Forensics (**58311PE)**<br>2. Advanced Operating Systems (**58312PE)**<br>3. Quantum Computing (**58313PE)**<br>4. Prompt Engineering (**58314PE)** | 3 | 0 | 0 | 3 |
| **2** | | Open Elective | 3 | 0 | 0 | 3 |
| **3** | **58301DS** | Dissertation Work Review - II | 0 | 0 | 12 | 6 |
| | | **Total Credits** | **6** | **0** | **12** | **12** |

## II Year II Semester

| S No | Code | Title | L | T | P | Credits |
|---|---|---|---|---|---|---|
| **1** | **58401DS** | Dissertation Work Review - III | 0 | 0 | 12 | 6 |
| **2** | **58402DS** | Dissertation Viva-Voce | 0 | 0 | 28 | 14 |
| | | **Total Credits** | **0** | **0** | **40** | **20** |

***For Dissertation Work Review - I, Please refer BH25 Academic Regulations.**

## OPEN ELECTIVES:

| | | |
|---|---|---|
| **Open Electives** | **58300OE** | Intellectual Property Rights |
| | **58301OE** | Generative AI |
| | **58302OE** | Intrusion Detection Systems |
| | **58303OE** | Digital Forensics |

**Audit Course I&II:**

| | | |
|---|---|---|
| **Audit Course** | **58501AC** | English for Research Paper Writing |
| | **58502AC** | Disaster Management |
| | **58503AC** | Sanskrit for Technical Knowledge |
| | **58504AC** | Value Education |
| | **58505AC** | Constitution of India |
| | **58506AC** | Pedagogy Studies |
| | **58507AC** | Stress Management by yoga |
| | **58508AC** | Personality Development Through Life Enlightenment Skills |

## I Year I Sem

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58101PC | Mathematical Foundations of Computer Science | 3 | 0 | 0 | 3 |

**Course Description:** This course introduces the elementary discrete mathematics for computer science and engineering such as formal logic notation, methods of proof, induction, sets, relations, graph theory, permutations and combinations, counting principles; recurrence relations and generating functions.

**Course Outcomes:** After completion of this course, the student will be able to

C101.1 Understand and construct precise mathematical proofs.
C101.2 Use logic and set theory to formulate precise statements.
C101.3 Analyze and solve counting problems on finite and discrete structures.
C101.4 Describe and manipulate sequences.
C101.5 Apply graph theory in solving computing problems.
C101.6 Construct Minimum Spanning Trees.

**Unit – I**

**The Foundations Logic and Proofs:** Propositional Logic, Applications of Propositional Logic, Propositional Equivalence, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy.

**Unit – II**

**Basic Structures, Sets, Functions, Sequences, Sums, Matrices and Relations:** Sets, Functions, Sequences & Summations, Cardinality of Sets and Matrices Relations, Relations and Their Properties, n-ary Relations and Their Applications, Representing Relations, Closures of Relations, Equivalence
Relations, Partial Orderings.

**Unit – III**

**Algorithms, Induction and Recursion:** Algorithms, The Growth of Functions, Complexity of Algorithms.
**Induction and Recursion:** Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Correctness.

**Unit – IV**

**Discrete Probability and Advanced Counting Techniques:** An Introduction to Discrete Probability. Probability Theory, Bayes' Theorem, Expected Value and Variance.
**Advanced Counting Techniques:** Recurrence Relations, Solving Linear Recurrence Relations, Divide-and-Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion-Exclusion, Applications of Inclusion-Exclusion.

**Unit – V**

**Graphs:** Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.
**Trees:** Introduction to Trees, Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees.

## Text Books:

1. Discrete Mathematical Structures with Applications to Computer Science: J.P. Tremblay, R. Manohar, McGraw-hill, 1st ed.
2. Discrete Mathematics for Computer Scientists & Mathematicians: Joe l. Mott, Abraham Kandel, Teodore P. Baker, Prentis Hall of India, 2nd ed.

## Reference Books:

1. Discrete and Combinatorial Mathematics - an applied introduction: Ralph.P. Grimald, Pearson education, 5th edition.
2. Discrete Mathematical Structures: Thomas Kosy, Tata McGraw Hill publishing co.

**I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58102PC** | Advanced Data Structures | 3 | 0 | 0 | 3 |
| **Pre - Requisite** | Data Structures | | | | |

**Course Description:** This course deals with the heap data structures such as leftist trees, binomial heaps, Fibonacci and minmax, Heaps and a variety of data structures such as disjoint sets, hash tables, search structures and digital search structures.

**Course Outcomes:** After completion of this course, the student will be able to

C102.1 Select the data structures that efficiently model the information in a problem.
C102.2 Understand the choice of data structures.
C102.3 Analyse the performance of different Hash functions.
C102.4 Design programs using a variety of data structures.
C102.5 Construct Digital search structures.
C102.6 Apply Pattern matching algorithms on various applications.

**Unit – I**
**Heap Structures:** Introduction, Min-Max Heaps, Leftist trees, Binomial Heaps, Fibonacci heaps

**Unit – II**
**Hashing and Collisions:** Introduction, Hash Tables, Hash Functions, different Hash Functions:- Division Method, Multiplication Method, Mid-Square Method, Folding Method, Collisions.

**Unit – III**
**Search Structures:** OBST, AVL trees, Red-Black trees, Splay trees, Multiway Search Trees: B-trees, 2-3 trees.

**Unit – IV**
**Digital Search Structures:** Digital Search trees, Binary tries and Patricia, Multiway Tries, Suffix trees, Standard Tries, Compressed Tries.

**Unit – V**
**Pattern matching:** Introduction, Brute force, the Boyer –Moore algorithm, Knuth-Morris-Pratt algorithm, Naïve String, Harspool, Rabin Karp

## Text Books:

1. Fundamentals of data structures in C++ Sahni, Horowitz, Mehatha, Universities Press.
2. Introduction to Algorithms, TH Cormen, PHI.

## Reference Books:

1. Design methods and analysis of Algorithms, SK Basu, PHI.
2. Data Structures & Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education.
3. Fundamentals of Computer Algorithms, 2nd Edition, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Universities Press.

**I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58111PE** | **Database Programming with PL/SQL** | 3 | 0 | 0 | 3 |

**Course Description:** This course deals with significance of SQL fundamentals, Evaluate functions & triggers of PL/SQL , packages in PL/SQL and its applications.

**Course Outcomes:** After completion of this course, the student will be able to

C111.1 Understand importance of PL/SQL basics.
C111.2 Identify various data Types and Cursor Structures.
C111.3 Apply Control Structures on databases.
C111.4 Implement functions and procedures using PL/SQL.
C111.5 Use various various packages.
C111.6 Analyse the importance of triggers in database.

**Unit – I**

**PL/SQL Basics:** Block Structure, Behavior of Variables in Blocks, Basic Scalar and Composite Data Types, Control Structures, Exceptions, Bulk Operations, Functions, Procedures, and Packages, Transaction Scope.

**Unit – II**

**Language Fundamentals & Control Structures:** Lexical Units, Variables and Data Types, Conditional Statements, Iterative Statements, Cursor Structures, Bulk Statements, Introduction to Collections, Object Types: Varray and Table Collections, Associative Arrays, Oracle Collection API.

**Unit – III**

**Functions and Procedures:** Function and Procedure Architecture, Transaction Scope, Calling Subroutines, Positional Notation, Named Notation, Mixed Notation, Exclusionary Notation, SQL Call Notation, Functions, Function Model Choices, Creation Options, Pass-by-Value Functions, Pass-by-Reference Functions, Procedures, Pass-by-Value Procedures, Pass-by-Reference Procedures, Supporting Scripts.

**Unit – IV**

**Packages:** Package Architecture, Package Specification, Prototype Features, Serially Reusable Precompiler Directive, Variables, Types, Components: Functions and Procedures, Package Body, Prototype Features, Variables, Types, Components: Functions and Procedures, Definer vs. Invoker Rights Mechanics, Managing Packages in the Database Catalog, Finding, Validating, and Describing Packages, Checking Dependencies, Comparing Validation Methods: Timestamp vs. Signature.

**Unit – V**

**Triggers:** Introduction to Triggers, Database Trigger Architecture, Data Definition Language Triggers, Event Attribute Functions, Building DDL Triggers, Data Manipulation Language Triggers, Statement-Level Triggers, Row-Level Triggers, Compound Triggers, INSTEAD OF Triggers, System and Database Event Triggers, Trigger Restrictions, Maximum Trigger Size, SQL Statements, LONG and LONG RAW Data Types.

## Text Book:

1. Oracle Database 12c PL/SQL Programming Michael McLaughlin, McGraw Hill Education.

## Reference Books:

1. N Benjamin Rosenzweig, Elena Silvestrova Rakhimov, Oracle PL/SQL by example Fifth Edition.
2. Dr. P. S. Deshpande, SQL & PL / SQL for Oracle 11g Black Book.

**I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58112PE | Deep Learning | 3 | 0 | 0 | 3 |

**Course Description:** This course deals with Deep Learning algorithms and their limitations using real-world data.

**Course Outcomes:** After completion of this course, the student will be able to

C112.1 Understand neural networks and traverse the layers of data.
C112.2 Learn training deep networks and high-level interfaces
C112.3 Understand applications of Deep Learning to Computer Vision
C112.4 Analyze Applications of Deep Learning to NLP
C112.5 Implement deep learning algorithms using Recurrent Neural Networks
C112.6 Apply Analogy reasoning using Convolutional Neural Networks.

**Unit – I**

**Introduction:** Feed forward Neural networks, Gradient descent and the back propagation algorithm, Unit saturation, the vanishing gradient problem, and ways to mitigate it. RelU Heuristics for avoiding bad local minima, Heuristics for faster training, Nestors accelerated gradient descent, Regularization, Dropout.

**Unit – II**

**Convolutional Neural Networks:** Architectures, convolution/pooling layers, Recurrent Neural Networks: LSTM, GRU, Encoder Decoder architectures. Deep Unsupervised Learning: Auto encoders, Variational Auto-encoders, Adversarial Generative Networks, Auto-encoder and DBM Attention and memory models, Dynamic Memory Models.

**Unit – III**

**Applications of Deep Learning to Computer Vision:** Image segmentation, object detection, automatic image captioning, Image generation with Generative adversarial networks, video to text with LSTM models, Attention Models for computer vision tasks.

**Unit – IV**

**Applications of Deep Learning to NLP:** Introduction to NLP and Vector Space Model of Semantics, Word Vector Representations: Continuous Skip-Gram Model, Continuous Bag-of-Words model (CBOW), Glove, Evaluations and Applications in word similarity.

**Unit – V**

**Analogy reasoning:** Named Entity Recognition, Opinion Mining using Recurrent Neural Networks: Parsing and Sentiment Analysis using Recursive Neural Networks: Sentence Classification using Convolutional Neural Networks, Dialogue Generation with LSTMs.

## Text Book:

1. Q. Li and C. Yao, Real-Time Concepts for Embedded Systems, Elsevier, 2011.

## Reference Books:

1. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press.
2. The Elements of Statistical Learning by T. Hastie, R. Tibshirani, and J. Friedman, Springer.
3. Probabilistic Graphical Models. Koller, and N. Friedman, MIT Press C. Hollabaugh, Embedded Linux: Hardware, Software, and Interfacing, Addison-Wesley, 2002.

**I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58113PE | Natural Language Processing | 3 | 0 | 0 | 3 |
| Pre - Requisite | 1. Data structures<br>2. finite automata and probability theory | | | | |

**Course Description:** This course deals with the problems and solutions of NLP and their relation to linguistics and statistics.

**Course Outcomes:** After completion of this course, the student will be able to

C113.1 Select linguistic phenomena in documents to Model with formal grammars.
C113.2 Understand proper experimental methodology for training NLP systems.
C113.3 Evaluate empirical NLP systems
C113.4 Manipulate probabilities, construct statistical models over strings and trees
C113.5 Analyze NLP algorithms to design different language modeling Techniques
C113.6 Design different language modeling Techniques.

**Unit – I**

**Finding the Structure of Words:** Words and Their Components, Issues and Challenges, Morphological Models Finding the Structure of Documents: Introduction, Methods, Complexity of the Approaches, Performances of the Approaches.

**Unit – II**

**Syntax Analysis:** Parsing Natural Language, Treebanks: A Data-Driven Approach to Syntax, Representation of Syntactic Structure, Parsing Algorithms, Models for Ambiguity Resolution in Parsing, Multilingual Issues.

**Unit – III**

**Semantic Parsing:** Introduction, Semantic Interpretation, System Paradigms, Word Sense Systems, Software.

**Unit – IV**

**Predicate**-Argument Structure, Meaning Representation Systems, Software.

**Unit – V**

**Discourse Processing:** Cohesion, Reference Resolution, Discourse Cohesion and Structure Language Modeling: Introduction, N-Gram Models, Language Model Evaluation, Parameter Estimation, Language Model Adaptation, Types of Language Models, Language-Specific Modeling Problems, Multilingual and Cross Lingual Language Modeling.

**Text Books:**

1. Multilingual natural Language Processing Applications: From Theory to Practice – Daniel M. Bikel and Imed Zitouni, Pearson Publication.
2. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S. Tiwary.

**Reference Book:**

1. Speech and Natural Language Processing - Daniel Jurafsky & James H Martin, Pearson Publications.

**I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58114PE | Advanced UNIX Programming | 3 | 0 | 0 | 3 |

**Course Description:** This course introduces Linux utilities and Shell scripting language (bash) to solve Problems.

**Course Outcomes:** After completion of this course, the student will be able to

- C114.1 Understand the Linux utilities and basics of Shell programming with Bourne again shell (bash).
- C114.2 Make use of Shell scripting language (bash) to solve Problems.
- C114.3 Implement some standard Linux utilities for processes.
- C114.4 Develop the skills necessary for systems programming
- C114.5 Interpret Interprocess Communication
- C114.6 Write network programs using Sockets.

**Unit – I**

**Linux Utilities** - File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities.

**Shell programming with Bourne again shell (bash)** - Introduction, shell responsibilities, pipes and Redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts

**Unit – II**

**Files and Directories** - File Concept, File types, File System Structure, file metadata-Inodes, kernel support for files, system calls for file I/O operations- 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, fchown, links-soft links and hard links – symlink, link, unlink. Directories - Creating, removing and changing Directories- mkdir, rmdir, chdir, obtaining current working directorygetcwd, Directory contents, Scanning Directories-opendir, readdir, closedir, rewinddir functions.

**Unit – III**

**Process** – Process concept, Layout of a C program image in main memory, Process environment - environment list, environment variables, getenv, setenv, Kernel support for process, process identification, process control - process creation, replacing a process image, waiting for a process, process termination, zombie process, orphan process, system call interface for process management fork, vfork, exit, wait, waitpid, exec family, Process Groups, Sessions and Controlling Terminal, Differences between threads and processes. Signals – Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise, alarm, pause, abort, sleep functions.

**Unit – IV**

**Interprocess Communication** - Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems, pipes-creation, IPC between related processes using unnamed pipes, FIFOs-creation, IPC between unrelated processes using FIFOs (Named pipes), differences between unnamed and named pipes, popen and pclose library functions. Message Queues - Kernel support for messages, APIs for message queues, client/server example. Semaphores - Kernel support for semaphores, APIs for semaphores, file locking with semaphores.

**Unit – V**

**Shared Memory** - Kernel support for shared memory, APIs for shared memory, shared memory example. Sockets - Introduction to Berkeley Sockets, IPC over a network, Client- Server model, Socket address structures (Unix domain and Internet domain),Socket system calls for connection oriented protocol and connectionless protocol, example-client/server programs-Single Server-Client connection, Multiple simultaneous clients, Socket options- setsockopt and fcntl system calls, Comparison of IPC mechanisms.

**Text Books:**

1. Unix System Programming using C++, T. Chan, PHI.
2. Advanced Programming in the Unix Environment, 2nd edition, W. R. Stevens and S. A. Rago, Pearson Education.
3. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.
4. Unix Network Programming, W. R. Stevens, PHI

**Reference Books:**

1. C Programming Language, Kernighan and Ritchie, PHI.
2. Beginning Linux Programming, 4th Edition, N. Matthew, R. Stones, Wrox, Wiley India Edition.
3. Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson.
4. System Programming with C and Unix, A. Hoover, Pearson.
5. Unix System Programming, Communication, Concurrency and Threads, K. A. Robbins and S. Robbins, Pearson Education.
6. Unix shell Programming, S. G. Kochan and P. Wood, 3rd edition, Pearson Education.
7. Shell Scripting, S. Parker, Wiley India Pvt. Ltd.
8. Unix and Shell programming, B. A. Forouzan and R. F. Gilberg, Cengage Learning.
9. Linux System Programming, Robert Love, O'Reilly, SPD.

**I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58121PE** | **Applied Cryptography** | 3 | 0 | 0 | 3 |

**Course Description:** This course introduces the significance of cryptographic protocols and symmetric and public key algorithms.

**Course Outcomes:** After completion of this course, the student will be able to

C121.1 Understand the various cryptographic protocols
C121.2 Analyze key length and algorithm types and modes
C121.3 Illustrate different public key algorithms in cryptosystems
C121.4 Understand special algorithms for protocols
C121.5 Undestand usage Secret key management protocol in the real world
C121.6 Develop Real World Approaches.

**Unit – I**

**Foundations:** Terminology, Steganography, Substitution Ciphers and Transposition Ciphers, Simple XOR, OneTime Pads, Computer Algorithms, Large Numbers, Cryptographic Protocols: Protocol Building Blocks Introduction to Protocols, Communications Using Symmetric Cryptography, One-Way Functions, One Way Hash Functions, Communications Using Public-Key Cryptography, Digital
Signatures, Digital Signatures with Encryption, Random and Pseudo-Random-Sequence Generation

**Unit – II**

**Cryptographic Techniques:** Key length: Symmetric Key length, Public key length, comparing symmetric and public key length. Algorithm types and modes: Electronic Codebook Mode, Block Replay, Cipher Block Chaining Mode, Stream Cipher, Self-Synchronizing Stream Ciphers, Cipher-Feedback Mode, Synchronous Stream Ciphers, Output-Feedback Mod, Counter Mode, Other Block-Cipher Modes.

**Unit – III**

**Public-Key Algorithms:** Background, Knapsack Algorithms, RSA, Pohlig-Hellman, Rabin,ElGamal, McEliece, Elliptic Curve Cryptosystems, LUC, Finite Automaton Public-Key Cryptosystems Public-Key Digital Signature Algorithms: Digital Signature Algorithm (DSA), DSA Variants, Gost Digital Signature Algorithm, Discrete Logarithm Signature Schemes, Ong-Schnorr-Shamir, ESIGN.

**Unit – IV**

**Special Algorithms for Protocols:** Multiple-Key Public-Key Cryptography, Secret-Sharing Algorithms, Subliminal Channel, Undeniable Digital Signatures, Designated Confirmer Signatures, Computing with Encrypted Data, Fair Coin Flips, One-Way Accumulators, All-or-Nothing Disclosure of Secrets, Fair and Failsafe Cryptosystems, Zero Knowledge Proofs of Knowledge, Blind Signatures, Oblivious Transfer, Secure Multiparty Computation, Probabilistic Encryption, Quantum Cryptography.

**Unit – V**

Real World Approaches IBM Secret key management protocol, ISDN, Kerberos, Krypto Knight, Privacy enhanced mail (PEM), Message security protocol (MSP), PGP, Public-Key Cryptography Standards (PKCS), Universal Electronic Payment System (UEPS).

## Text Book:

1. Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C (cloth).

**I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58122PE | Software Quality Engineering | 3 | 0 | 0 | 3 |

**Course Description:** This course deals with significance of Quality, quality assurance, quality engineering.

**Course Outcomes:** After completion of this course, the student will be able to

C122.1 Understand software quality and its perspectives.
C122.2 Understand Quality Assurance Classification.
C122.3 Implement Quality Engineering activities.
C122.4 Design Automation Test planning and preparation.
C122.5 Explore Coverage and Usage testing based on checklist.
C122.6 Prepare Case studies on cartridge Support Software.

**Unit – I**

**Software Quality:** Quality- perspectives and expectations, Quality frameworks and ISO-9126, correctness and defects: Definitions, properties and Measurements, A historical perspective of quality, software quality.

**Unit – II**

**Quality Assurance Classification:** QA as dealing with defects, Defect prevention- Education and training, Formal method, Other defect prevention techniques, Defect Reduction - Inspection: Direct fault detection and removal, Testing: Failure observation and fault removal, other techniques and risk identification, Defect Containment- software fault tolerance, safety assurance and failure Containment.

**Unit – III**

**Quality Engineering:** Activities and process, Quality planning: Goal setting and Strategy formation, Quality assessment and Improvement, Quality engineering in software process.

**Unit – IV**

Test Activities, Management and Automation Test planning and preparation, Test execution, Result checking and measurement, Analysis and follow- up, Activities People and Management, Test Automation.

**Unit – V**

Coverage and usage testing based on checklist and partitions Checklist based testing and its limitations, Testing for partition Coverage, Usage based Statistical testing with Musa's operational profiles, Constructing operational profiles Case Study: OP for the cartridge Support Software.

## Text Books:

1. Jeff Tian, Software Quality Engineering, Testing, Quality Assurance, and Quantifiable improvement.
2. Richard N. Taylor, Software Architecture: Foundations, Theory, and Practice.

**I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58123PE | Mining Massive Datasets | 3 | 0 | 0 | 3 |
| Pre - Requisite | 1. Data Mining<br>2. Discrete Mathematics. | | | | |

**Course Description:** This course will cover practical algorithms for solving key problems in mining of massive datasets and focuses on parallel algorithmic techniques that are used for large datasets.

**Course Outcomes:** After completion of this course, the student will be able to

C123.1 Handle massive data using MapReduce.
C123.2 Implement algorithms for massive data sets and methodologies in the context of data mining.
C123.3 Understand the algorithms for extracting models and information from large datasets
C123.4 Illustrate Clustering for Streams and Parallelism.
C123.5 Develop recommendation systems.
C123.6 Gain experience in matching various algorithms for particular classes of problems.

**Unit – I**

Data Mining-Introduction-Definition of Data Mining-Statistical Limits on Data Mining, MapReduce and the New Software Stack-Distributed File Systems, MapReduce, Algorithms Using MapReduce.

**Unit – II**

Similarity Search: Finding Similar Items-Applications of Near-Neighbor Search, Shingling of Documents, Similarity-Preserving Summaries of Sets, Distance Measures. Streaming Data: Mining Data Streams-The Stream Data Model , Sampling Data in a Stream, Filtering Streams.

**Unit – III**

Link Analysis-PageRank, Efficient Computation of PageRank, Link Spam Frequent Itemsets-Handling Larger Datasets in Main Memory, Limited-Pass Algorithms, Counting Frequent Items in a Stream. Clustering-The CURE Algorithm, Clustering in Non-Euclidean Spaces, Clustering for Streams and Parallelism.

**Unit – IV**

Advertising on the Web-Issues in On-Line Advertising, On-Line Algorithms, The Matching Problem, The Adwords Problem, Adwords Implementation. Recommendation Systems-A Model for Recommendation Systems, Content-Based Recommendations, Collaborative Filtering, Dimensionality Reduction, The NetFlix Challenge.

**Unit – V**

Mining Social-Network Graphs-Social Networks as Graphs, Clustering of Social-Network Graphs, Partitioning of Graphs, Simrank, Counting Triangles.

**Text Book:**

1. Jure Leskovec, Anand Rajaraman, Jeff Ullman, Mining of Massive Datasets, 3rd Edition

**Reference Books:**

1. Jiawei Han & Micheline Kamber, Data Mining – Concepts and Techniques 3rd Edition Elsevier.
2. Margaret H Dunham, Data Mining Introductory and Advanced topics, PEA.
3. Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann

**I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58124PE | Agile Methodologies | 3 | 0 | 0 | 3 |

**Pre - Requisite**
1. Data Structures
2. Finite automata and probability theory

**Course Description:** This course covers the concepts of Agile development, releasing, planning and developing.

**Course Outcomes:** After completion of this course, the student will be able to

C124.1 Understand basic concepts of agile methods.
C124.2 Relate extreme programming concepts.
C124.3 Analyze real customer involvement and ubiquitous language.
C124.4 Implement Bug free Release and version control.
C124.5 Discuss risk management and iteration planning
C124.6 Summarize incremental requirements, refactoring, incremental design and architecture

**Unit – I**

Introduction Extreme Programming (XP) - Agile Development Why Agile - Understanding Success, Beyond Deadlines, Importance of Organizational Success, Introduction to Agility How to Be Agile - Agile methods, Don't make your own method, Road to mastery Understanding XP (Extreme Programming) - XP life cycle, XP team, XP Concepts Adopting XP - Knowing whether XP is suitable, Implementing XP, assessing Agility Practicing XP - Thinking - Pair Programming, Energized work, Informative Workspace, Root cause Analysis, Retrospectives.

**Unit – II**

Collaborating Trust, Sit together, Real customer involvement, Ubiquitous language, meetings, coding standards, Iteration demo, Reporting.

**Unit – III**

Releasing: Bugfree Release, Version Control, fast build, continuous integration, Collective ownership, Documentation.

**Unit – IV**

Planning: Version, Release Plan, Risk Management, Iteration Planning, Slack, Stories, Estimating.

**Unit – V**

Developing: Incremental requirements, Customer tests, Test driven development, Refactoring, Incremental design and architecture, spike solutions, Performance optimization, Exploratory testing.

**Text Book:**

1. The art of Agile Development, James Shore and Shane Warden, 11th Indian Reprint, O'Reilly,2018

**Reference Books:**

1. Learning Agile, Andrew Stellman and Jennifer Greene, O'Reilly, 4th Indian Reprint, 2018.
2. Practices of an Agile Developer, Venkat Subramaniam and Andy Hunt, SPD, 5th Indian Reprint, 2015.
3. Agile Project Management - Jim Highsmith, Pearson Low price Edition 2004

**M. Tech. I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58103PC | Advanced Data Structures Lab | 0 | 0 | 4 | 2 |

**Course Description:** This Course emphasizes the implementation of efficient algorithms for key operations on advanced data structures.

**Course Outcomes:** After completion of this course, the students will be able to

C103.1 Implement insertion, deletion, and search operations on various tree structures.
C103.2 Analyze the performance of different sorting algorithms.
C103.3 Evaluate the efficiency of different heap structures for key operations.
C103.4 Apply advanced pattern matching algorithms to solve relevant problems.

**List of Programs:**

**Cycle 1:**

1. Write a program to perform the following operations on BST:
   a) Insert an element.
   b) Delete an element from a binary search tree.
   c) Search for a key element in a binary search tree.
2. Write a program for implementing the following sorting methods:
   a) Merge sort
   b) Heap sort
   c) Quick sort
3. Write a program to perform the following operations on B-Tree:
   a) Insert an element
   b) Delete an element
   c) Search for a key element
4. Write a program to perform the following operations on Min-Max heap:
   a) Insert an element
   b) Delete an element
   c) Search for a key element
5. Write a program to perform the following operations on Lefiist tree:
   a) Insert an element
   b) Delete an element
   c) Search for a key element

**Cycle 2:**

6. Write a program to perform the following operations on binomial heap:
   a) Insert an element
   b) Delete an element
   c) Search for a key element
7. Write a program to perform the following operations on AVL tree:
   a) Insert an element
   b) Delete an element
   c) Search for a key element

8. Write a program to perform the following operations on Red-Black tree:
    a) Insert an element
    b) Delete an element
    c) Search for a key element
9. Write a program to implement all the functions of a dictionary using hashing.
10. Write a program for implementing Knuth-Morris-Pratt pattern matching algorithm.
11. Write a program for implementing Brute Force pattern matching algorithm.
12. Write a program for implementing Boyer pattern matching algorithm.

**Text Books:**

1. F Fundamentals of Data structures in C, E. Horowitz, S. Sahni and Susan Anderson Freed, 2nd Edition, Universities Press
2. Data Structures Using C – A.S. Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson education.
3. Introduction to Data Structures in C, Ashok Kamthane, 1st Edition, Pearson.

**Reference Books:**

1. The C Programming Language, B.W. Kernighan, Dennis M.Ritchie, PHI/Pearson Education.
2. C Programming with problem solving, J.A. Jones & K. Harrow, Dreamtech Press.
3. Data structures: A Pseudocode Approach with C, R.F. Gilberg And B.A. Forouzan, 2nd Edition, Cengage Learning.

**M. Tech. I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58131PE** | **Database Programming with PL/SQL Lab** | 0 | 0 | 4 | 2 |

**Course Description:** This Course emphasizes the implementation of SQL fundamentals and also covers programs on control structures, packages in PL/SQL and its applications.

**Course Outcomes:** After completion of this course, the students will be able to

C131.1 Understand importance of PL/SQL basics
C131.2 Implement functions and procedures using PL/SQL.
C131.3 Implement procedures using PL/SQL.
C131.4 Understand the importance of triggers in database.

**List of Programs:**

**Cycle 1:**

1. Write a Pl/SQL program using FOR loop to insert ten rows into a database table.
2. Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID), write a cursor to select the five highest paid employees from the table.
3. Illustrate how you can embed PL/SQL in a high-level host language such as C/Java And demonstrates how a banking debit transaction might be done.
4. Given an integer i, write a PL/SQL procedure to insert the tuple (i, 'xxx') into a given relation.
5. Write a PL/SQL program to demonstrate Exceptions.

**Cycle 2:**

6. Write a PL/SQL Program to Demonstrate Cursors
7. Write a PL/SQL Program to Demonstrate Functions
8. Write a PL/SQL Program to Demonstrate Packages
9. Write a PL/SQL Program to Create Procedures
10. Write a PL/SQL Program to Create Triggers

## Text Book:

1. Oracle Database 12c PL/SQL Programming Michael McLaughlin, McGraw Hill Education.

## Reference Books:

1. N Benjamin Rosenzweig, Elena Silvestrova Rakhimov, Oracle PL/SQL by example Fifth Edition.
2. Dr. P. S. Deshpande, SQL & PL / SQL for Oracle 11g Black Book

## M. Tech. I Year I Sem

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58132PE | Deep Learning Lab | 0 | 0 | 4 | 2 |

**Course Description:** This Course emphasizes the implementation of successful machine learning concepts.

**Course Outcomes:** After completion of this course, the students will be able to

C132.1 Learn the Fundamental Principles of Deep Learning
C132.2 Identify the Deep Learning Algorithms for Various Types of Learning Tasks in various domains.
C132.3 Implement Deep Learning Algorithms.
C132.4 Implement various Solve Real-world problems.

**List of Programs:**

**Cycle 1:**

1. Setting up the Spyder IDE Environment and Executing a Python Program.
2. Installing Keras, Tensorflow and Pytorch libraries and making use of them.
3. Applying the Convolution Neural Network on computer vision problems.
4. Image classification on MNIST dataset (CNN model with Fully connected layer).

**Cycle 2:**

5. Applying the Deep Learning Models in the field of Natural Language Processing
6. Train a sentiment analysis model on IMDB dataset, use RNN layers with LSTM/GRU notes
7. Applying the Autoencoder algorithms for encoding the real-world data
8. Applying Generative Adversial Networks for image generation and unsupervised tasks.

**Text Books:**

1. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press.
2. The Elements of Statistical Learning by T. Hastie, R. Tibshirani, and J. Friedman, Springer.
3. Probabilistic Graphical Models. Koller, and N. Friedman, MIT Press.

**Reference Books:**

1. Bishop, C, M., Pattern Recognition and Machine Learning, Springer, 2006.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Golub, G.,H., and Van Loan, C.,F., Matrix Computations, JHU Press,2013.
4. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

**M. Tech. I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58133PE | Natural Language Processing Lab | 0 | 0 | 4 | 2 |

**Course Description:** This Course emphasizes the implementation of the problems and solutions of NLP.

**Course Outcomes:** After completion of this course, the students will be able to

C133.1 Show sensitivity to linguistic phenomena and to model them with formal grammars.
C133.2 Manipulate probabilities, construct statistical models over strings and trees.
C133.3 Estimate parameters using supervised and unsupervised training methods.
C133.4 Design, implement, and analyze NLP algorithms.

**List of Programs:**

Implement the following using Python

**Cycle 1:**

1. Tokenization
2. Stemming
3. Stop word removal (a, the, are,.)
4. Word Analysis
5. Word Generation

**Cycle 2:**

6. Pos tagging
7. Morphology
8. chunking
9. N-Grams
10. N-Grams Smoothing

**Text Books:**

1. Multilingual natural Language Processing Applications: From Theory to Practice – Daniel M. Bikel and Imed Zitouni, Pearson Publication.
2. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S. Tiwary

**Reference Book:**

1. Speech and Natural Language Processing - Daniel Jurafsky & James H Martin, Pearson Publications.

**M. Tech. I Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58134PE** | **Advanced Unix Programming Lab** | 0 | 0 | 4 | 2 |

**Course Description:** This Course provides hands-on experience in Linux Systems Programming. Students will master Bash shell scripting for task automation and use the C programming language to interface with the Linux kernel.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C134.1 | Utilize Linux utilities and Bash scripting to automate system tasks and solve computational problems. |
| C134.2 | Implement low-level File I/O and directory management programs using standard C system calls. |
| C134.3 | Develop applications that manage Process Lifecycles and concurrent execution using fork() and pipes |
| C134.4 | Construct networked and synchronized systems using Sockets, Semaphores, and Shared Memory for efficient IPC. |

**List of Programs:**

**Cycle 1:**

1. Working with basic Unix/Linux commands/utilities
2. Explore various Shells (Ex.: Bourne shell, C Shell, K shell, Bash shell) and editors Ex.: Vim, gedit, Sublime Text, Emacs, etc.,)
3. Write a shell that takes a valid directory name as an argument and recursively descend all the subdirectories, finds the maximum length of any file in that hierarchy and writes this maximum value to the standard output.
4. 
   i. Write a shell script that accepts two file names as arguments, checks if the permissions for these files are identical and if the permission are identical, output common permission and otherwise output each file name followed by its permissions.
   ii. Write a shell script which accepts valid log in names as arguments and prints their corresponding home directories, if no arguments are specified, print a suitable error message.
   iii. Write a shell script that gets executed displays the message either “Good Morning” or “Good Afternoon” or “Good Evening” depending upon time at which the user logs in.
5. Write a C program to implement the following Unix/Linux commands:

   i) ls ii) cp iii) cat

6. Write a program that takes one or more file/directory names as command line input and reports the following information on the file: File type, Number of links, Time of last access, Read, Write and Execute permissions.
7. Write a C program that illustrates the uses of the following directory management system calls: i) opendir( ) ii) readdir( ) iii) closedir( )

## Cycle 2:

8. Write a C program that counts the number of blanks in a text file using standard I/O
9. Write a C program that illustrates the creation of child process using fork( ) system call.
10. Write a C program that illustrates how to execute two commands concurrently using Pipe(), system call.
11. Write a C program to illustrate popen( ) and pclose( ) system calls.
12. Write a C program to implement the kill() and raise( )function.
13. Write a C program that illustrates file locking using semaphores.
14. Write a C program that illustrates various Message Queue APIs(msgget(), msgctl(), msgsnd(), msgrcv())
15. Write a C program that implements a producer-consumer system with two processes using semaphores
16. Write a C program that illustrates IPC mechanism using shared memories.

**Text Books:**

1. Unix System Programming using C++, T. Chan, PHI.
2. Advanced Programming in the Unix Environment, 2nd edition, W. R. Stevens and S.A. Rago, Pearson Education.
3. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.
4. Unix Network Programming, W. R. Stevens, PHI

## Reference Books:

1. C Programming Language, Kernighan and Ritchie, PHI.
2. Beginning Linux Programming, 4th Edition, N. Matthew, R. Stones, Wrox, Wiley India Edition.
3. Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson.
4. System Programming with C and Unix, A. Hoover, Pearson.
5. Unix System Programming, Communication, Concurrency and Threads, K. A. Robbins and S. Robbins, Pearson Education.
6. Unix shell Programming, S. G. Kochan and P. Wood, 3rd edition, Pearson Education.
7. Shell Scripting, S. Parker, Wiley India Pvt. Ltd.
8. Unix and Shell programming, B. A. Forouzan and R. F. Gilberg, Cengage Learning.
9. Linux System Programming, Robert Love, O'Reilly, SPD.

## M.Tech. I Year I Sem

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58104PC | Research Methodology & IPR | 2 | 0 | 0 | 2 |

**Course Description:** This course covers identifying research problems, conducting literature reviews, mastering technical writing and IPR components.

**Course Outcomes:** After completion of this course, the student will be able to

- C104.1 Identify research problems based on the scope and objectives.
- C104.2 Apply knowledge of literature studies, plagiarism, and ethics to ensure proper research practices and academic integrity.
- C104.3 Develop technical writing skills by creating well-structured research papers and reports.
- C104.4 Evaluate the nature of intellectual property rights and recent developments.
- C104.5 Interpret patent rights and their implications for protecting intellectual property in various fields.
- C104.6 Generalize the intellectual property rights for various fields.

**Unit – I**

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problems, data collection, analysis, interpretation, Necessary instrumentations.

**Unit – II**

Effective literature studies approaches, analysis, Plagiarism, Research ethics.

**Unit – III**

Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee.

**Unit – IV**

Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.

**Unit – V**

Patent Rights: Scope of Patent Rights. Licensing and transfer of technology. Patent information and databases. Geographical Indications. New Developments in IPR: Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc. Traditional knowledge Case Studies, IPR and IITs.

**Text Books:**

1. Stuart Melville and Wayne Goddard, Research methodology: an introduction for science & engineering students.
2. C.R. Kothari, Research Methodology, methods & techniques, 2nd edition, New age International publishers.

**Reference Books:**

1. Ranjit Kumar, 2nd Edition, “Research Methodology: A Step by Step Guide for beginners”
2. Halbert, “Resisting Intellectual Property”, Taylor & Francis Ltd ,2007.
3. Mayall, “Industrial Design”, McGraw Hill, 1992.
4. Niebel, “Product Design”, McGraw Hill, 1974.
5. Asimov, “Introduction to Design”, Prentice Hall, 1962.
6. Robert P. Merges, Peter S. Menell, Mark A. Lemley, “Intellectual Property in New Technological Age”, 2016.
7. T. Ramappa, “Intellectual Property Rights Under WTO”, S. Chand, 2008.

**M.Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58201PC | Advanced Algorithms | 3 | 0 | 0 | 3 |

**Pre - Requisite** Design and Analysis of Algorithms

**Course Description:** This course builds upon core concepts, exploring sophisticated techniques to address complex computational challenges.

**Course Outcomes:** After completion of this course, the student will be able to

C201.1 Evaluate the performance of various algorithms for specific scenarios.
C201.2 Apply the concepts of matroids to solve related problems.
C201.3 Design algorithms to address network-related problems.
C201.4 Make use of matrix computations for relevant problems
C201.5 Utilize manipulations on integers/polynomials with various representations
C201.6 Identify the classes of NP-Problems

**Unit – I**

Sorting: Review of various sorting algorithms, topological sorting Graph: Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkstra's), depth-first search and computation of strongly connected components, emphasis on correctness proof of the algorithm and time/space analysis, example of amortized analysis.

**Unit – II**

Matroids: Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set. Application to MST. Graph Matching: Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path.

**Unit – III**

Flow-Networks: Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm. Matrix Computations: Strassen's algorithm and introduction to divide and conquer paradigm, inverse of a triangular matrix, relation between the time complexities of basic matrix operations, LUP-decomposition.

**Unit – IV**

Shortest Path in Graphs: Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming. Modulo Representation of integers/polynomials: Chinese Remainder Theorem, Conversion between base-representation and modulo-representation. Extension to polynomials. Application: Interpolation problem. Discrete Fourier Transform (DFT): In complex field, DFT in modulo ring. Fast Fourier Transform algorithm. Schonhage-Strassen Integer Multiplication algorithm.

**Unit – V**

Linear Programming: Geometry of the feasibility region and Simplex algorithm NP-completeness: Examples, proof of NP-hardness and NP-completeness. Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures.

**Text Books:**

1. Cormen, Leiserson, Rivest, Stein, Introduction to Algorithms.
2. Aho, Hopcroft, Ullman, The Design and Analysis of Computer Algorithms.
3. Kleinberg and Tardos. Algorithm Design.

## Reference Books:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, SartajSahni and Sanguthevar Rajasekaran, Universities Press, 2011, Second Edition.
2. Algorithmic Graph Theory and Perfect Graphs, Martin Charles Golumbic, 2004, Elsevier, Second Edition.
3. Algorithms and Complexity, Herbert S. Wilf, AK Peters/CRC Press, 2002, Second Edition.
4. Parameterized Complexity, Rodney G. Downey and M. R. Fellows, Springer, 2012.
5. Approximation Algorithms, Vijay V. Vajirani, Springer, 2001.

**M.Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58202PC | Advanced Computer Architecture | 3 | 0 | 0 | 3 |

**Pre - Requisite** Computer Organization and Architecture

**Course Description:** This Course introduces scalable, parallel and vector processing architectures.

**Course Outcomes:** After completion of this course, the student will be able to

C202.1 Identify parallel computer models.
C202.2 Analyze the performance of scalable architectures.
C202.3 Design different kinds of pipeline architectures using shared-memory organization
C202.4 Compare parallel architectures involving multiprocessors and/or multicomputers
C202.5 Illustrate different types of scalable architectures.
C202.6 Implement multiprocessors using different computer organizations

**Unit – I**

Theory of Parallelism, Parallel computer models, The State of Computing, Multiprocessors and Multicomputers, Multivector and SIMD Computers, PRAM and VLSI models, Architectural development tracks, Program and network properties, Conditions of parallelism, Program partitioning and Scheduling, Program flow Mechanisms, System interconnect Architectures.

**Unit – II**

Principles of Scalable performance, Performance metrics and measures, Parallel Processing applications, Speed up performance laws, Scalability Analysis and Approaches, Hardware Technologies, Processes and Memory Hierarchy, Advanced Processor Technology, Superscalar and Vector Processors.

**Unit – III**

Shared-Memory Organizations, Sequential and weak consistency models, Pipelining and superscalar techniques, Linear Pipeline Processors, Non-Linear Pipeline Processors, Instruction Pipeline design, Arithmetic pipeline design, superscalar pipeline design.

**Unit – IV**

Parallel and Scalable Architectures, Multiprocessors and Multicomputers, Multiprocessor system interconnects, cache coherence and synchronization mechanism, Three Generations of Multicomputers, Message-passing Mechanisms, Multivetor and SIMD computers.

**Unit – V**

Vector Processing Principles, Multivector Multiprocessors, Compound Vector processing, SIMD computer Organizations, The connection machine CM-5.

## Text Books:

1. Advanced Computer Architecture, Kai Hwang, 2nd Edition, Tata McGraw Hill Publishers

## Reference Books:

1. Computer Architecture, J.L. Hennessy and D.A. Patterson, 4th Edition, ELSEVIER.
2. Advanced Computer Architectures, S.G.Shiva, Special Indian edition, CRC, Taylor &Francis.
3. Introduction to High Performance Computing for Scientists and Engineers, G. Hager and G. Wellein, CRC Press, Taylor & Francis Group.
4. Advanced Computer Architecture, D. Sima, T. Fountain, P. Kacsuk, Pearson education.

**M.Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58211PE | Enterprise Cloud Concepts | 3 | 0 | 0 | 3 |

**Course Description:** This course deals with understanding various cloud computing mechanisms and architectures.

**Course Outcomes:** After completion of this course, the student will be able to

- C211.1 Understand importance of cloud computing
- C211.2 Analyze the various cloud computing mechanisms
- C211.3 Evaluate various management mechanisms
- C211.4 Compare various cloud architectures
- C211.5 Assess the smart enterprises
- C211.6 Explores the methodologies

**Unit – I**

Understanding Cloud Computing: Origins and influences, Basic Concepts and Terminology, Goals and Benefits, Risks and Challenges. Fundamental Concepts and Models: Roles and Boundaries, Cloud Characteristics, Cloud Delivery Models, Cloud Deployment Models.

**Unit – II**

Cloud-Enabling Technology: Broadband Networks and Internet Architecture, Data Center Technology, Virtualization Technology CLOUD COMPUTING MECHANISMS: Cloud Infrastructure Mechanisms: Logical Network Perimeter, Virtual Server, Cloud Storage Device, Cloud Usage Monitor, Resource Replication.

**Unit – III**

Cloud Management Mechanisms: Remote Administration System, Resource Management System, SLA Management System, Billing Management System, Case Study Example Cloud Computing Architecture Fundamental Cloud Architectures: Workload Distribution Architecture, Resource Pooling Architecture, Dynamic Scalability Architecture, Elastic Resource Capacity Architecture, Service Load Balancing Architecture, Cloud Bursting Architecture, Elastic Disk Provisioning Architecture, Redundant Storage Architecture, Case Study Example.

**Unit – IV**

Cloud-Enabled Smart Enterprises Introduction, Revisiting the Enterprise Journey, Service-Oriented Enterprises, Cloud Enterprises, Smart Enterprises, The Enabling Mechanisms of Smart Enterprises Cloud-Inspired Enterprise Transformations Introduction, The Cloud Scheme for Enterprise Success, Elucidating the Evolving Cloud Idea, Implications of the Cloud on Enterprise Strategy, Establishing a Cloud-Incorporated Business Strategy.

**Unit – V**

Transitioning to Cloud-Centric Enterprises the Tuning Methodology, Contract Management in the Cloud Cloud-Instigated IT Transformations Introduction, Explaining Cloud Infrastructures, A Briefing on Next-Generation Services, Service Infrastructures, Cloud Infrastructures, Cloud Infrastructure Solutions, Clouds for Business Continuity, The Relevance of Private Clouds, The Emergence of Enterprise Clouds.

**Text Books:**

1. Erl Thomas, Puttini Ricardo, Mahmood Zaigham, Cloud Computing: Concepts, Technology & Architecture 1st Edition,
2. Pethuru Raj, Cloud Enterprise Architecture, CRC Press

**Reference Book:**

1. James Bond, The Enterprise Cloud, O'Reilly Media, Inc

## M.Tech. I Year II Sem

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58212PE | Cyber Security | 3 | 0 | 0 | 3 |

**Course Description:** This course deals essential Cyber Security concepts, focusing on threats, vulnerabilities, laws, digital forensics, mobile security, organizational impacts, and privacy challenges.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C212.1 | Explain fundamental Cyber Security concepts, including threats, vulnerabilities, and the CIA triad. |
| C212.2 | Analyze different types of cyber attacks and their motives. |
| C212.3 | Apply cyber laws and forensic methods to investigate digital crimes. |
| C212.4 | Evaluate security challenges in mobile and wireless computing environments. |
| C212.5 | Assess the impact of cyber threats on organizations and design appropriate security policies. |
| C212.6 | Identify privacy risks and implement strategies to protect sensitive data across different domains |

**Unit – I**

Introduction to Cyber Security: Basic Cyber Security Concepts, layers of security, Vulnerability, threat, Harmful acts, Internet Governance – Challenges and Constraints, Computer Criminals, CIA Triad, Assets and Threat, motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.

**Unit – II**

Cyberspace and the Law & Cyber Forensics: Introduction, Cyber Security Regulations, Roles of International Law. The INDIAN Cyberspace, National Cyber Security Policy. Introduction, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Forensics Investigation, Challenges in Computer Forensics.

**Unit – III**

Cybercrime: Mobile and Wireless Devices: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones, Organizational security Policies and Measures in Mobile Computing Era, Laptops.

**Unit – IV**

Cyber Security: Organizational Implications: Introduction, cost of cybercrimes and IPR issues, web threats for organizations, security and privacy implications, social media marketing: security risks and perils for organizations, social computing and the associated challenges for organizations.

**Unit – V**

Privacy Issues: Basic Data Privacy Concepts: Fundamental Concepts, Data Privacy Attacks, Data linking and profiling, privacy policies and their specifications, privacy policy languages, privacy in different domains- medical, financial, etc Cybercrime: Examples and Mini-Cases Examples: Official Website of Maharashtra Government Hacked, Indian Banks Lose Millions of Rupees, Parliament Attack, Pune City Police Bust Nigerian Racket, e-mail spoofing instances. Mini-Cases: The Indian Case of online Gambling, An Indian Case of Intellectual Property Crime, Financial Frauds in Cyber Domain.

**Text Books:**

1. Nina Godbole and Sunit Belpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley
2. B.B. Gupta, D.P. Agrawal, Haoxiang Wang, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press, ISBN 9780815371335,2018.

**Reference Books:**

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
2. Introduction to Cyber Security, Chwan-Hwa(john) Wu,J. David Irwin, CRC Press T&F Group

**I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58213PE | Parallel computing | 3 | 0 | 0 | 3 |
| Pre - Requisite | 1. Computer Organization & Architecture<br>2. Operating Systems<br>3. Programming for problem solving | | | | |

**Course Description:** This course introduces parallel computing concepts, covering parallel algorithm design, programming paradigms (MPI, PThreads), and fundamental algorithms for matrix operations, sorting, and graph processing.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C213.1 | Understand the motivation, scope, and basic communication operations in parallel computing. |
| C213.2 | Apply principles of parallel algorithm design and analytical modeling of parallel programs. |
| C213.3 | Develop parallel programs using Message Passing Interface (MPI) and Shared Memory (PThreads). |
| C213.4 | Implement parallel matrix-vector and matrix-matrix multiplication algorithms. |
| C213.5 | Design and analyze parallel sorting algorithms such as Bubble Sort, Quick Sort, and Radix Sort. |
| C213.6 | Solve graph problems using parallel algorithms for Minimum Spanning Tree, Single-Source Shortest Path, DFS, and BFS. |

**Unit – I**

Parallel Computing: Introduction, Motivation and scope - Parallel Programming Platforms – Basic Communication Operations.

**Unit – II**

Principles of Parallel Algorithm Design - Analytical Modelling of Parallel Programs.

**Unit – III**

Programming using Message Passing Paradigm (MPI) – Programming Shared Address Space Platforms (PThreads)

**Unit – IV**

Dense Matric Algorithms (Matrix-Vector Multiplication, Matrix-Matrix Multiplication) – Sorting Algorithms (Issues, Bubble Sort, Quick Sort, Bucket Sort, Enumeration Sort, Radix Sort)

**Unit – V**

Graph Algorithms (Minimum Spanning Tree: Prim's Algorithm - Single-Source Shortest Paths: Dijkstra's Algorithm) Search Algorithms (DFS, BFS).

**Text Book:**

1. Introduction to Parallel Computing, Second Edition, Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta, Addison-Wesley, 2003, ISBN: 0201648652

**Reference Books:**

1. Parallel Computing – Theory and Practice, Second Edition, Michaek J. Quinn, Tata McGrawHill Edition.
2. Parallel Computers – Architectures and Programming, V. Rajaraman, C. Siva Ram Murthy, PHI.

**M.Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58214PE** | **Large Language Models** | 3 | 0 | 0 | 3 |

**Course Description:** This course introduces Large Language Models (LLMs), covering their evolution, architecture, training techniques, prompt engineering, evaluation methods, deployment strategies, and ethical considerations.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C214.1 | Understand the architecture and inner workings of transformer-based Large Language Models (LLMs). |
| C214.2 | Apply prompt engineering and fine-tuning techniques for domain-specific applications. |
| C214.3 | Evaluate LLM performance using standard metrics and benchmark datasets. |
| C214.4 | Implement practical LLM-based applications such as text generation, summarization, and question answering using popular frameworks. |
| C214.5 | Identify key challenges in LLM training, deployment, and efficient scaling. |
| C214.6 | Analyze ethical, legal, and societal implications of LLM development and usage, including bias and hallucination mitigation. |

**Unit – I**

Foundations of Large Language Models Introduction to LLMs: Definition, scope, and historical evolution from statistical NLP to transformers. The Transformer architecture: Attention mechanisms, self-attention, multi-head attention. Pretraining objectives: Masked language modeling (MLM), Causal language modeling (CLM). Evolution of LLMs: BERT, GPT series, T5, LLaMA, Mistral.

**Unit – II**

Training and Fine-Tuning LLMs Pretraining datasets and tokenization: BPE, SentencePiece, WordPiece. Fine-tuning approaches: Full fine-tuning, LoRA, adapters, instruction tuning. Domain adaptation and few-shot/zero-shot learning. Data augmentation for LLMs and prompt-based tuning.

**Unit – III**

Prompt Engineering and Applications Principles of prompt design: Zero-shot, few-shot, and chain-of-thought prompting. System prompts, role prompting, and context length optimization. Use cases: Text generation, summarization, code generation, question answering, chatbots. Tools & frameworks: Lang Chain, Llama Index, Hugging Face Transformers.

**Unit – IV**

Evaluation and Deployment of LLMs Evaluation metrics: Perplexity, BLEU, ROUGE, METEOR, human evaluation. Benchmark datasets: GLUE, SuperGLUE, HELM, BIG-bench. Deployment strategies: API-based deployment, on-prem deployment, inference optimization. Scaling and latency considerations; quantization and pruning for LLMs.

**Unit – V**

Ethics, Safety, and Future Directions Bias, fairness, and toxicity in LLMs. Hallucination problem and mitigation techniques. Legal and regulatory issues: Copyright, data privacy, AI Act. Trends in LLM research: Multimodal LLMs, retrieval-augmented generation (RAG), open-source LLM ecosystems.

**Text Books:**

1. Vaswani, A. et al. (2017) Attention Is All You Need – NIPS Conference Paper.
2. Lewis, P. et al. (2021) Language Models are Few-Shot Learners – OpenAI Research Paper.
3. Tunstall, L., von Werra, L., & Wolf, T. (2022) Natural Language Processing with Transformers – O'Reilly Media.

**Reference Books:**

1. Bommasani, R. et al. (2021) On the Opportunities and Risks of Foundation Models – Stanford CRFM.
2. Jurafsky, D., & Martin, J. H. (2023) Speech and Language Processing (3rd Edition draft) – Pearson.
3. Mollick, E., & Mollick, L. (2024) Co-Intelligence: Living and Working with AI – Little, Brown Spark.
4. Hugging Face Documentation – https://huggingface.co/docs/

**M.Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58221PE** | **Bioinformatics** | 3 | 0 | 0 | 3 |

**Course Description:** This course deals with biological data analysis to explore biological databases, sequence alignment, phylogenetics, structural bioinformatics, genomics, proteomics, and their applications in life sciences.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C221.1 | Understand biological databases and tools for sequence and structural analysis. |
| C221.2 | Perform sequence alignment and interpret phylogenetic trees. |
| C221.3 | Construct and interpret phylogenetic trees to analyze evolutionary relationships among organisms. |
| C221.4 | Predict and visualize protein structures using computational methods and modeling tools. |
| C221.5 | Employ genomics and proteomics techniques for applications in drug discovery, agriculture, and personalized medicine. |
| C221.6 | Analyze biological data using computational approaches for insights into genomics, proteomics, and phylogenetics. |

**Unit – I**

The Central Dogma & XML (Bio XML) for Bioinformatics: Watson's definition, information flow, from data to knowledge, Convergence, the organization of DNA, the organization of Proteins, Introduction, Differences between HTML and XML, fundamentals of XML, fundamentals of XML namespaces. Introduction to DTDs, Document type Declarations, Declaring elements, declaring attributes, working with entities XML Schemas, Essential Concepts, working with simple types, working with complex types, Basic namespaces issues.

**Unit – II**

Perl (Bioperl) for Bioinformatics: Representing sequence data, program to store a DNA sequence, concatenating DNA fragments, Transcription, Calculating the reverse complement in Perl, Proteins, files, reading proteins in files, Arrays, Flow control, finding motifs, counting Nucleotides, exploding strings into arrays, operating on strings, writing to files, subroutines and bugs.

**Unit – III**

Databases: Flat file, Relational, object oriented databases, object Relational and Hypertext, Data life cycle, Database Technology, Database Architecture, Database Management Systems and Interfaces.

**Unit – IV**

Sequence Alignment Algorithms: Biological motivations of sequence analysis, the models for sequence analysis and their biological motivation, global alignment, local alignment, End free-space alignment and gap penalty, Sequence Analysis tools and techniques.

**Unit – V**

Phylogenetic Analysis: Introduction, methods of Phylogenetic analysis, distance methods, the neighbor- Joining (NJ) method, The Fitch/ Margoliash method, character-based methods, Other methods, Tree evaluation and problems in phylogenetic analysis, Clustering, Protein structure visualization and Protein structure prediction.

**Text Books:**

1. S.C. Rastogi, N. Mendiratta, "Bioinformatics Methods and Applications", CBS publications, 2004
2. James D. Tisdall, "Beginning Perl for Bioinformatics" O'Reilly media, 1st Edition, 2001

**Reference Books:**

1. D.R. Westhead, J.H. Parish, "Bioinformatics" Viva books private limited, New Delhi ,2003.
2. Att Wood, "Bioinformatics" Pearson Education, 2004.
3. Bryan Bergeron, M.D, "Bioinformatics Computing" Pearson Education, 2003.

.

**M.Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58222PE** | **Adhoc Sensor Networks** | 3 | 0 | 0 | 3 |
| **Pre - Requisite** | Computer Networks | | | | |

**Course Description:** This course introduces the principles of Mobile Ad Hoc Networks (MANETs) and Wireless Sensor Networks (WSNs), focusing on routing protocols, data transmission techniques, geocasting, and layer-specific challenges.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C222.1 | Understand the characteristics, applications, and challenges of MANETs and sensor networks. |
| C222.2 | Analyze different MANET routing protocols including proactive, reactive, hybrid, and position-based approaches. |
| C222.3 | Design efficient data transmission and rebroadcasting schemes to address the broadcast storm problem in MANETs. |
| C222.4 | Implement geocasting techniques and analyze TCP performance in ad hoc environments. |
| C222.5 | Understand the architecture and lower-layer issues of Wireless Sensor Networks (physical, MAC, link, and routing layers). |
| C222.6 | Evaluate upper-layer challenges in WSNs, including transport and application layer adaptations for dynamic network behavior. |

**Unit – I**

Introduction to Ad Hoc Networks Characteristics of MANETs, Applications of MANETs and Challenges of MANETs. Routing in MANETs Criteria for classification, Taxonomy of MANET routing algorithms, Topology-based routing algorithms-Proactive: DSDV, WRP; Reactive: DSR, AODV, TORA; Hybrid: ZRP; Position- based routing algorithms-Location Services-DREAM, Quorum-based, GLS; Forwarding Strategies Greedy Packet, Restricted Directional Flooding- DREAM, LAR; Other routing algorithms-QoS Routing, CEDAR.

**Unit – II**

Data Transmission Broadcast Storm Problem, Rebroadcasting Schemes-Simple-flooding, Probability-based Methods, Area-based Methods, Neighbour Knowledge-based: SBA, Multipoint Relaying, AHBP. Multicasting: Tree-based: AMRIS, MAODV; Mesh-based: ODMRP, CAMP; Hybrid: AMRoute, MCEDAR.

**Unit – III**

Geocasting Data-transmission Oriented-LBM; Route Creation Oriented-GeoTORA, MGR. TCP over Ad Hoc TCP protocol overview, TCP and MANETs, Solutions for TCP over Ad hoc.

**Unit – IV**

Basics of Wireless Sensors and Lower Layer Issues Applications, Classification of sensor networks, Architecture of sensor network, Physical layer, MAC layer, Link layer, Routing Layer.

**Unit – V**

Upper Layer Issues of WSN Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs.

**Text Books:**

1. Ad Hoc and Sensor Networks – Theory and Applications, Carlos Corderio Dharma P.Aggarwal, World Scientific Publications, March 2006, ISBN – 981-256-681-3
2. Wireless Sensor Networks: An Information Processing Approach, Feng Zhao, Leonidas Guibas, Elsevier Science, ISBN – 978-1-55860-914-3 (Morgan Kauffman)

**Reference Books:**

1. C. Siva Ram Murthy, B.S. ManojAd Hoc Wireless Networks: Architectures and Protocols
2. Taieb Znati Kazem Sohraby, Daniel Minoli, Wireless Sensor Networks: Technology, Protocols and Applications, Wiley.

**M.Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58223PE** | **Robotic Process Automation** | 3 | 0 | 0 | 3 |

**Course Description:** This course deals with Robotic Process Automation (RPA) concepts, focusing on automating repetitive tasks using RPA tools like Automation Anywhere.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C223.1 | Understand the fundamentals of RPA, its use cases, and the Automation Anywhere platform. |
| C223.2 | Navigate the Web Control Room and Client interfaces to manage bots, devices, and audit logs. |
| C223.3 | Configure settings, manage users, roles, and licenses, and interact with exposed APIs. |
| C223.4 | Create bots using various recorders (Smart, Web, Screen) and manage tasks with task editor and variables. |
| C223.5 | Utilize advanced commands such as Excel, Database, String Operations, XML, Terminal Emulator, and FTP for robust automation. |
| C223.6 | Design complex workflows using workflow and report designers, implement error handling, and integrate PDF and Object Cloning commands |

**Unit – I**

Introduction to Robotic Process Automation & Bot Creation Introduction to RPA and Use cases – Automation Anywhere Enterprise Platform – Advanced features and capabilities – Ways to create Bots.

**Unit – II**

Web Control Room and Client Introduction - Features Panel - Dashboard (Home, Bots, Devices, Audit, Workload, Insights) - Features Panel – Activity (View Tasks in Progress and Scheduled Tasks) - Bots (View Bots Uploaded and Credentials).

**Unit – III**

Devices (View Development and Runtime Clients and Device Pools) - Workload (Queues and SLA Calculator) - Audit Log (View Activities Logged which are associated with Web CR) - Administration (Configure Settings, Users, Roles, License and Migration) - Demo of Exposed API's – Conclusion – Client introduction and Conclusion.

**Unit – IV**

Bot Creator Introduction – Recorders – Smart Recorders – Web Recorders – Screen Recorders - Task Editor – Variables - Command Library – Loop Command – Excel Command – Database Command - String Operation Command - XML Command.

**Unit – V**

Terminal Emulator Command - PDF Integration Command - FTP Command - PGP Command - Object Cloning Command - Error Handling Command - Manage Windows Control Command - Workflow Designer - Report Designer.

**Text Book:**

1. Learning Robotic Process Automation: Create Software robots and automate business processes with the leading RPA tool - UiPath: Create Software robots. with the leading RPA tool – UiPath Kindle Edition

**Reference Book:**

1. Robotic Process Automation A Complete Guide - 2020 Edition Kindle Edition

**M.Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58224PE | Generative AI | 3 | 0 | 0 | 3 |

**Course Description:** This course explores the foundations, architectures, and applications of Generative AI, focusing on advanced neural models like GANs, VAEs, and Large Language Models (LLMs).

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C224.1 | Understand the foundations of AI, deep learning, and the role of generative models in modern applications. |
| C224.2 | Analyze advanced neural architectures such as VAEs, GANs, Transformers, and their role in generative tasks. |
| C224.3 | Apply Large Language Models and prompt engineering techniques to build applications like chatbots and creative writing tools. |
| C224.4 | Design and develop multi-agent systems that collaborate for generative AI tasks in areas like research, creativity, and distributed problem solving. |
| C224.5 | Implement multimodal generative solutions involving text, code, image, and video using frameworks like LangChain and RAG. |
| C224.6 | Evaluate ethical, legal, and social implications of generative AI systems, and apply responsible AI practices in real-world deployments. |

**Unit – I**

Foundations of AI and Generative Models Introduction and historical evolution to Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP)and Deep Learning (DL), Structure of Artificial Neural Networks (ANNs), Mathematical and computational foundations of generative modeling, Overview of generative models and their applications across various domains; Importance of Generative AI in modern applications, Transfer learning and in advancing Generative AI.

**Unit – II**

Advanced Neural Architectures for Generative AI Variational Autoencoders (VAEs): principles and applications, Generative Adversarial Networks (GANs): architecture and working principles; Transformer architecture and attention mechanisms (in detail); Long Short-Term Memory Networks (LSTMs) and the limitations of traditional RNNs/LSTMs, Advanced Transformer architectures and techniques, Pre-training and transfer learning strategies for generative models.

**Unit – III**

Large Language Models and Prompt Engineering Overview of Large Language Models (LLMs), GPT architecture, variants, and working principles, Prétraining and fine-tuning GPT models for applications (e.g., chatbots, text generation), Case study: GPT-based customer support chatbot, BERT architecture, pre-training objectives, and fine-tuning, Prompt Engineering: Designing effective prompts, controlling model behavior, and improving output quality, Fine-tuning language models for creative writing and chatbot development.

**Unit – IV**

Multi-Agent Systems and Generative AI Applications Introduction to Multi-Agent Systems (MAS),Types of agents: reactive, deliberative, hybrid, and learning agents, Multi-agent collaboration

and orchestration for generative tasks, Use cases: autonomous research assistants, cooperative creative generation, distributed problem-solving, Frameworks and tools: AutoGen, CrewAI, Hugging GPT for LLM-powered multi-agent systems, Generative AI applications: Art, Creativity, Image/Video generation, Music composition, Healthcare, Finance, Real-world case studies and deployment
challenges.

**Unit – V**

Frameworks, Multimodal Applications, and Ethics LangChain framework: components and LLM application development, Retrieval-Augmented Generation (RAG), Embeddings, Indexing networks, and Vector databases, Generative AI across modalities: Text, Code, Image, and Video generation, Image and Video generation using GANs and VAEs, Multimodal Generative AI: integration and training strategies, Ethical considerations: bias, fairness, trust, and responsible AI deployment, Social
and legal implications of Generative AI, Risk mitigation strategies and real-world ethical case studies.

**Text Books:**

1. Altaf Rehmani, Generative AI for Everyone: Understanding the Essentials and Applications of This Breakthrough Technology.
2. Charu C. Aggarwal, Neural Networks and Deep Learning: A Textbook. Joseph Babcock, Raghav Bali, Generative AI with Python and TensorFlow 2, 2024.

**Reference Books:**

1. Josh Kalin, Generative Adversarial Networks Cookbook.
2. Jesse Sprinter, Generative AI in Software Development: Beyond the Limitations of Traditional Coding, 2024.

**M. Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58203PC | Advanced Algorithms Lab | 0 | 0 | 4 | 2 |

**Course Description:** This course offers hands-on experience in implementing and analyzing algorithms for optimization, mathematical computations, string matching, and graph theory.

**Course Outcomes:** After completion of this course, the students will be able to

C203.1 Solve complex optimization problems using algorithmic techniques.
C203.2 Implement efficient mathematical algorithms for long integers and Gaussian elimination.
C203.3 Analyze and apply string matching algorithms effectively.
C203.4 Utilize graph algorithms to determine maximum flow in networks.

**List of Programs:**

**Cycle 1:**

1. Implement assignment problem using Brute Force method
2. Perform multiplication of long integers using divide and conquer method.
3. Implement a solution for the knapsack problem using the Greedy method.
4. Implement Gaussian elimination method.
5. Implement LU decomposition

**Cycle 2:**

Implement

6. Warshall algorithm
7. Rabin Karp algorithm.
8. KMP algorithm.
9. Harspool algorithm
10. Max-flow problem.

**Text Book:**

1. Design and Analysis of Algorithms, S.Sridhar, OXFORD University Press.

**Reference Books:**

1. Introduction to Algorithms, second edition, T.H. Cormen, C.E. Leiserson, R.L. Rivest and C.Stein,PHI Pvt. Ltd./ Pearson Education.
2. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharam, Universities Press.
3. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson education

**M. Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58231PE** | **Enterprise Cloud Concepts Lab** | 0 | 0 | 4 | 2 |

**Course Description:** This Course emphasizes the implementation of cloud computing and its fundamental concepts and models,

**Course Outcomes:** After completion of this course, the students will be able to

C231.1 Understand importance of cloud architecture
C231.2 Illustrating the fundamental concepts of cloud security
C231.3 Analyze various cloud computing mechanisms
C231.4 Understanding the architecture and working of cloud computing.

**List of Programs:**

**Cycle 1:**

1. Install Virtualbox/VMware Workstation with different flavors of linux or windows OS on top of windows7 or 8.
2. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs
3. Install Google App Engine. Create a hello world app and other simple web applications using python/java

**Cycle 2:**

4. Find a procedure to transfer the files from one virtual machine to another virtual machine.
5. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
6. Install Hadoop single node cluster and run simple applications like word count.

## E- Resources

1. https://www.iitk.ac.in/nt/faq/vbox.htm
2. https://edwardsamuel.wordpress.com/2014/10/25/tutorial-creating-openstack-instance-in-trystack/
3. https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster

**M. Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58232PE | Cyber Security Lab | 0 | 0 | 4 | 2 |

**Course Description:** This course provides hands-on experience with essential cybersecurity tools and techniques, including network scanning, honeypots, cryptographic algorithms, foot printing, packet sniffing, traffic analysis, digital forensics, and memory analysis.

**Course Outcomes:** After completion of this course, the students will be able to

| | |
|---|---|
| C232.1 | Understand the fundamental principles and architectures of Deep Learning. |
| C232.2 | Analyze and select appropriate deep learning algorithms for different learning tasks across various domains. |
| C232.3 | Design and implement deep learning solutions to address real-world problems effectively. |
| C232.4 | Evaluate and optimize deep learning models using performance metrics to enhance accuracy and efficiency. |

**List of Programs:**

**Cycle 1:**

1. Perform an Experiment for port scanning with NMAP.
2. Setup a honeypot and monitor the honeypot on the network
3. Install Jcrpt /Cryptool tool (or any other equivalent) and demonstrate Asymmetric, Symmetric crypto algorithm, Hash and Digital/PKI signatures.
4. Generate minimum 10 passwords of length 12 characters using open SSL command
5. Perform practical approach to implement Foot printing-Gathering target information using Dmitry-Dmagic, UAtester.
6. Working with sniffers for monitoring network communication (Wireshark).

**Cycle 2:**

7. Use Snort to perform real time traffic analysis and packet logging.
8. Perform email analysis using Autopsy tool.
9. Perform Registry analysis and get boot time logging using process monitor tool
10. Perform File type detection using Autopsy tool
11. Perform Memory capture and analysis using FTK imager tool
12. Perform Network analysis using the Network Miner tool

**Text Books:**

1. Real Digital Forensics for Handheld Devices, E. P. Dorothy, Auerback Publications, 2013.
2. Handbook of Digital Forensics and Investigation, E. Casey, Academic Press, 2010

**Reference Books:**

1. The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics, J. Sammons, Syngress Publishing, 2012.
2. Malware Forensics Field Guide for Windows Systems: Digital Forensics Field Guides, C. H. Malin, E. Casey and J. M. Aquilina, Syngress, 2012
3. The Best Damn Cybercrime and Digital Forensics Book Period, J. Wiles and A.Reyes, Syngress, 2007.

**M. Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58233PE | Parallel Computing Lab | 0 | 0 | 4 | 2 |

**Course Description:** This course provides practical experience in designing and implementing parallel algorithms using MPI, OpenMP, and Pthread programming constructs.

**Course Outcomes:** After completion of this course, the students will be able to

| | |
|---|---|
| C233.1 | Develop parallel programs for matrix-vector and matrix-matrix multiplication using MPI for improved computational performance. |
| C233.2 | Implement parallel sorting algorithms (Bubble Sort, Quick Sort, Bucket Sort) using OpenMP and Pthread constructs. |
| C233.3 | Design and implement parallel graph algorithms such as Prim's, DFS, BFS, and Dijkstra's using MPI, OpenMP, and Pthread programming techniques. |
| C233.4 | Analyze the performance improvements of parallel solutions compared to sequential implementations in real-time computing scenarios. |

**List of Programs:**

Implement the following using Python

**Cycle 1:**

1. Design a parallel program to implement Matrix-Vector and Matrix-Matrix Multiplication using MPI library.
2. Design a parallel program to implement Bubble Sort using OpenMP and Pthread Programming Constructs.
3. Design a parallel program to implement Quick Sort using OpenMP and Pthread Programming Constructs.
4. Design a parallel program to implement Bucket Sort using OpenMP and Pthread Programming Constructs.

**Cycle 2:**

5. Design a parallel program to implement Prim's Algorithm using OpenMP and Pthread Programming Constructs.
6. Design a parallel program to implement DFS Algorithm using OpenMP and Pthread Programming Constructs.
7. Design a parallel program to implement BFS Algorithm using OpenMP and Pthread Programming Constructs.
8. Design a parallel program to implement Dijkstra's Algorithm using MPI library

**Text Book:**

1. Introduction to Parallel Computing, Second Edition, Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta, Addison-Wesley, 2003, ISBN: 0201648652

**Reference Books:**

1. Parallel Computing – Theory and Practice, Second Edition, Michaek J. Quinn, Tata McGrawHill Edition.
2. Parallel Computers – Architectures and Programming, V. Rajaraman, C. Siva Ram Murthy, PHI.

**M. Tech. I Year II Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58234PE** | **Large Language Models Lab** | 0 | 0 | 4 | 2 |

**Course Description:** This Course provides LLM technology, including the principles of text generation, summarization, and classification by effectively utilizing key Python libraries such as transformers and LangChain to build and interact with LLM-based systems.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C234.1 | Describe the core concepts of Large Language Models and their different applications. |
| C234.2 | Implement Python code to perform fundamental LLM tasks, including text generation, summarization, and classification. |
| C234.3 | Analyze the effect of various parameters and prompts on LLM outputs and optimize the model's performance for specific use cases. |
| C234.4 | Construct a functional LLM application, such as a RAG system or a simple agent, to solve a defined problem. |

**List of Programs:**

**Cycle 1:**

1. Write a Python code to generate text using a pre-trained Large Language Model (LLM) based on a given prompt.
2. Write a python codefor a pre-trained Large Language Model (LLM) to classify text into categories?
3. Write a python code using a pre-trained Large Language Model (LLM)for Text translation from one language to other language.
4. Write a Python code to summarize a long text using a pre-trained LLM.
5. Write a Python code to simulate a chatbot conversation using a conversational LLM.
6. Write a python code to detect sentiment of a given text using LLM

**Cycle 2:**

7. Write a Python code to demonstrate Retrieval-Augmented Generation (RAG) using LangChain, allowing an LLM to answer questions based on a custom knowledge base.
8. Write a Python code to predict the missing word in a sentence using a pre-trained Large Language Model.
9. Understand Text Processing by LLMs
   a) Load the tokenizer for the GPT-2 model.
   b) Tokenize a sample sentence and print the resulting input_ids and attention_mask.
   c) Decode the input_ids back to text to verify the process.
   d) Compare the output of this tokenizer with one from a different model family (e.g., Google's T5 or Meta's Llama 2).

10. Building a Simple Application with LangChain
    a) Use LangChain's LLMChain and PromptTemplate to create a structured text summarization tool.
    b) Implement semantic search over a document collection using LLM-based embeddings.

11. Evaluating Model Performance
    a) Use the evaluate library from Hugging Face to calculate BLEU and ROUGE for generated outputs. Compare LLM predictions before and after fine-tuning.
    b) Use Hugging Face Transformers to perform extractive vs abstractive summarization. Evaluate results using ROUGE scores.

**Textbooks:**

1. Vaswani, A. et al. Attention Is All You Need –2017, NIPS Conference Paper.
2. Lewis, P. et al., Language Models are Few-Shot Learners 2021, OpenAI Research Paper.
3. Tunstall, L., von Werra, L., & Wolf, T., Natural Language Processing with Transformers –2022, O'Reilly Media.

**Reference Books:**

1. Bommasani, R. et al., On the Opportunities and Risks of Foundation Models –2021, Stanford CRFM.
2. Jurafsky, D., & Martin, J. H., Speech and Language Processing (3rd Edition draft) 2023, Pearson.
3. Mollick, E., & Mollick, L. Co-Intelligence: Living and Working with AI –2024, Little, Brown Spark.
4. Hugging Face Documentation – https://huggingface.co/docs/

**M. Tech. II Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58311PE** | **Digital Forensics (Professional Elective-V)** | 3 | 0 | 0 | 3 |
| **Pre – Requisite** | Computer Networks | | | | |

**Course Description:** This course introduces the principles, tools, and investigative techniques used in digital and computer forensics for identifying, collecting, and analyzing electronic evidence. It also covers cybercrime investigation processes, forensic imaging, memory and network forensics, mobile forensics, and relevant legal frameworks such as the IT Act and DPDP Act.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C311.1 | Understand relevant legislation and ethical codes to ensure legal compliance in digital investigations. |
| C311.2 | Illustrate the holistic investigative process and detective procedures for various categories of computer crime. |
| C311.3 | Execute standard protocols for the search, seizure, and documentation of electronic evidence for court proceedings. |
| C311.4 | Implement evidence management strategies and forensic mindsets to maintain the integrity of the chain of custody. |
| C311.5 | Summarize specialized computer and network forensic investigations using industry-standard workstations and open-source tools. |
| C311.6 | Discover mobile forensic techniques and recent trends to retrieve data from contemporary digital devices. |

**Unit – I**

**Digital Forensics Science:** Forensics science, computer forensics, and digital forensics.
**Computer Crime:** Criminalistics as it relates to the investigative process, analysis of cyber criminalistics area, holistic approach to cyber-forensics, NIST Incident Response Lifecycle, NIST Digital Forensics**,** Framework (NIST SP 800-86).

**Unit – II**

**Cyber Crime Scene Analysis:** Discuss the various court orders etc., methods to search and seizure electronic evidence, retrieved and un-retrieved communications, Discuss the importance of understanding what court documents would be required for a criminal investigation, Forensic Imaging Techniques, FTK Imager and Autopsy.

**Unit – III**

**Evidence Management & Presentation:** Create and manage shared folders using operating system, importance of the forensic mindset, define the workload of law enforcement, Explain what the normal case would look like, Define who should be notified of a crime, parts of gathering evidence, Define and apply probable cause, Forensic Report Writing standards.

**Unit – IV**

**Computer Forensics:** Prepare a case, Begin an investigation, understand computer forensics workstations and software, Conduct an investigation, Complete a case, Critique a case.
**Memory Forensics:** Volatility Framework, Ransomware Investigations.
**Network Forensics:** Open-source security tools for network forensic analysis, requirements for preservation of network data.

**Unit – V**

**Mobile Forensics:** mobile forensics techniques, mobile forensics tools, Recent trends in mobile forensic technique and methods to search and seizure electronic evidence.

**Legal Aspects of Digital Forensics**: IT Act 2000, amendment of IT Act 2008, Introduction to Digital Personal Data Protection (DPDP) Act.

**Text Books:**

1. John Sammons, The Basics of Digital Forensics, Elsevier.
2. John Vacca, Computer Forensics: Computer Crime Scene Investigation, Laxmi Publications.

**Reference Books:**

1. William Oettinger, Learn Computer Forensics: A beginner's guide to searching, analyzing, and securing digital evidence, 1st Edition, 2020, Packt Publishing.
2. Thomas J. Holt , Adam M. Bossler, Kathryn C. Seigfried-Spellar , Cybercrime and Digital Forensics: An Introduction, Routledge
3. Kenneth C.Brancik , Insider Computer Fraud, 2008, Auerbach Publications Taylor & Francis Group.
4. Bill Nelson, Amelia Phillips, Christopher Steuart, Guide to Computer Forensics and Investigations, 4th E, Cengage Learning.
5. NIST SP 800-86 Guide to Integrating Forensic Techniques into Incident Response, 2006.
6. Michael Ligh, Andrew Case, Jamie, Walters, The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory, 2014, Wiley.
7. Gerard Johansen, Digital Forensics and Incident Response - 3rd Edition,2022 O'Reilly.
8. https://dpdpa.com/index.html

**M. Tech. II Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58312PE | Advanced Operating Systems (Professional Elective-V) | 3 | 0 | 0 | 3 |

**Pre - Requisite** Operating Systems

**Course Description:** This course deals with the main concepts of advanced operating systems (parallel processing systems, distributed systems, real time systems, network operating systems, and open source operating systems), Hardware and software features that support these systems.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C312.1 | Examine the architectures and logical clock synchronization mechanisms of advanced distributed systems. |
| C312.2 | Analyze token-based and non-token-based algorithms for mutual exclusion in distributed environments. |
| C312.3 | Appraise deadlock handling strategies and detection algorithms across various control organizations. |
| C312.4 | Illustrate the design issues of multiprocessor operating systems, including threads and processor scheduling. |
| C312.5 | Determine the requirements and architectural mechanisms for building Distributed File Systems. |
| C312.6 | Implement load distributing algorithms and coherence protocols for Distributed Shared Memory. |

**Unit – I**

Architectures of Distributed Systems: System Architecture Types, Distributed Operating Systems, Issues in Distributed Operating Systems, Communication Primitives.
Theoretical Foundations: Inherent Limitations of a Distributed System, Lamport's Logical Clocks, Vector Clocks, Causal Ordering of Messages, Termination Detection.

**Unit – II**

Distributed Mutual Exclusion: The Classification of Mutual Exclusion Algorithms, Non-Token – Based Algorithms: Lamport's Algorithm, The Ricart-Agrawala Algorithm, Maekawa's Algorithm, Token-Based Algorithms: Suzuki-Kasami's Broadcast Algorithm, Singhal's Heuristic Algorithm, Raymond's Heuristic Algorithm, Leader Election algorithms (Bully, Ring), Fault Tolerance.

**Unit – III**

Distributed Deadlock Detection: Preliminaries, Deadlock Handling Strategies in Distributed Systems,, Time out based and Lease based Approach**,** Issues in Deadlock Detection and Resolution, Control Organizations for Distributed Deadlock Detection, Centralized- Deadlock – Detection Algorithms,
Distributed Deadlock Detection Algorithms, Hierarchical Deadlock Detection Algorithms**.**

**Unit – IV**

Multiprocessor System Architectures: Introduction, Motivation for multiprocessor Systems, Basic Multiprocessor System Architectures, Multi Processor Operating Systems: Introduction, Structures of Multiprocessor Operating Systems, Operating Design Issues, Threads, Process Synchronization, Processor Scheduling, Containerization. Distributed File Systems: Architecture, Mechanisms for Building Distributed File Systems, Design Issues.

**Unit – V**

Distributed Scheduling: Issues in Load Distributing, Components of a Load Distributed Algorithm, Stability, Load Distributing Algorithms, Requirements for Load Distributing, Task Migration, Issues in task Migration Distributed Shared Memory: Architecture and Motivation, Algorithms for Implementing DSM, Memory Coherence, Coherence Protocols, Design Issues. Cloud Resource Management: Autoscaling Mechanisms, Serverless computing overview.

**Text Books:**

1. Mukesh Singhal, Niranjan G. Shivaratri, Advanced Concepts in Operating Systems, 2001, Tata McGraw-Hill Edition.
2. Sam NewMan, Building Microservices: Designing Fine-Grained Systems, 2021, O'Reilly.
3. George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, Distributed Systems: Concepts and Design, 5th Edition, Addison Wesley.

**Reference Books:**

1. Andrew S. Tanenbaum, Maarten Van Steen, Distributed Systems, 2nd Edition, 2007, Pearson Prentice Hall .
2. Nigel Poulton, Docker Deep Dive, 2025, Shroff/Poulton.
3. https://docs.aws.amazon.com/

**M. Tech. II Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58313PE | Quantum Computing (Professional Elective-V) | 3 | 0 | 0 | 3 |

**Course Description:** This course introduces the fundamentals of quantum computing, covering linear algebra, quantum mechanics principles, qubits, quantum gates, circuits, and key quantum algorithms. It also examines the impact of quantum computing on modern cryptography and explores emerging post-quantum cryptographic techniques.

**Course Outcomes:** After completion of this course, the student will be able to

- C313.1 Explain the fundamentals of quantum computing and its mathematical foundations.
- C313.2 Describe the physical principles and implementation of qubits.
- C313.3 Apply quantum gates and circuits to implement basic quantum algorithms.
- C313.4 Analyze standard quantum algorithms for computational problem solving.
- C313.5 Evaluate the impact of quantum computing on asymmetric cryptographic algorithms.
- C313.6 Assess security challenges and future directions in post-quantum cryptography.

**Unit – I**

**Introduction to Essential Linear Algebra:** Some Basic Algebra, Matrix Math, Vectors and Vector Spaces, Set Theory, Unitary matrices and Hermitian operators**.**

**Complex Numbers:** Definition of Complex Numbers, Algebra of Complex Numbers, Complex Numbers Graphically, Vector Representations of Complex Numbers, Pauli Matrice, Transcendental Numbers.

**Unit – II**

**Basic Physics for Quantum Computing:** The Journey to Quantum, Quantum Physics Essentials, Basic Atomic Structure, Hilbert Spaces, Uncertainty, Quantum States, Entanglement.

**Basic Quantum Theory:** Further with Quantum Mechanics, Quantum Decoherence, Quantum Electrodynamics, Quantum Chromodynamics, Feynman Diagram Quantum Entanglement and QKD, Quantum Entanglement, Interpretation, QKE, Quantum Noise models (Bit-Flip, Phase-Flip), Shift from theoretical physics-heavy content to computational quantum mechanics focus.

**Unit – III**

**Quantum Architecture:** Further with Qubits, Quantum Gates, More with Gates, Quantum Circuits, The D-Wave Quantum Architecture , IBM Quantum architecture.

**Quantum Hardware:** Qubits, How Many Qubits Are Needed? Addressing Decoherence, Topological Quantum Computing, Quantum Essentials, Introduction to Qiskit.

**Unit – IV**

**Quantum Algorithms** What Is an Algorithm? Deutsch's Algorithm, Deutsch-Jozsa Algorithm, Bernstein-Vazirani Algorithm, Simon's Algorithm, Quantum Fourier Transform (QFT), Shor's Algorithm, Grover's Algorithm, Modern NISQ Algorithms: Variational Quantum Algorithms (VQE), Quantum Approximate Optimization Algorithm( QAOA).

**Unit – V**

**Current Asymmetric Algorithms:** RSA, Diffie-Hellman, Elliptic Curve.

**The Impact of Quantum Computing on Cryptography:** Asymmetric Cryptography, Specific Algorithms, Specific Applications, Post-Quantum Cryptography (PQC) & Lattice-based cryptography.

**Text Books:**

1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge University Press
2. Dr. Chuck Easttom, Quantum Computing Fundamentals, Pearson

**Reference Books:**

1. Noson S. Yanofsky and Mirco A. Mannucci, Quantum Computing for Computer Scientists by Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Basic Concepts.
2. Pittenger, A. O., An Introduction to Quantum Computing Algorithms, Basic Tools and Special Topics, World Scientific Publishing.
3. Johannes Buchmann, Erik Dahmen, Post-Quantum Cryptography, Springer.
4. https://quantum.cloud.ibm.com/docs

**M. Tech. II Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58314PE** | **Prompt Engineering (Professional Elective-V)** | 3 | 0 | 0 | 3 |

**Course Description:** This course introduces the principles and techniques of prompt engineering for generative AI and large language models (LLMs), including prompt design, optimization, and structured output generation. It also covers LLM architectures, tokenization, text chunking, vector databases, and Retrieval-Augmented Generation (RAG) for building and evaluating real-world AI applications.

**Course Outcomes:** After completion of this course, the student will be able to

- C314.1 Apply core principles of prompt engineering to guide generative AI outputs effectively.
- C314.2 Explain the architecture and working of modern large language models (LLMs).
- C314.3 Generate structured outputs (JSON, YAML, CSV) using advanced prompting techniques.
- C314.4 Implement text chunking, tokenization, and format control using Python-based tools.
- C314.5 Build Retrieval-Augmented Generation (RAG) pipelines using vector databases such as FAISS and Pinecone.
- C314.6 Evaluate LLM outputs for quality, efficiency, and real-world deployment scenarios.

**Unit – I**

**Fundamentals and Principles of Prompting** Overview of the Five Principles of Prompting: Give Direction, Specify Format, Provide Examples, Evaluate Quality, Divide Labor.
**Prompt Techniques:** zero shot, few shot, chain of thoughts and Role based prompting.
**Prompt Optimization:** Prompt iteration and refinement**.**

**Unit – II**

**Introduction to Large Language Models for Text Generation** What Are Text Generation Models, Vector Representations: The Numerical Essence of Language, Transformer Architecture: Orchestrating Contextual Relationships, Probabilistic Text Generation: The Decision Mechanism, Historical Underpinnings: The Rise of Transformer Architectures, Attention Mechanism**,** OpenAI's Generative Pretrained Transformers, GPT-3.5-turbo and ChatGPT, GPT-4, Google's Gemini, Meta's Llama and Open Source**,** Hallucination Causes.

**Unit – III**

**Standard Practices for Text Generation with LLM- Part-A** Generating Lists, Hierarchical List Generation, When to Avoid Using Regular Expressions, Generating JSON, YAML Filtering YAML Payloads, Handling Invalid Payloads in YAML, Diverse Format Generation with ChatGPT, Mock CSV Data, Universal Translation Through LLMs, Ask for Context, Text Style Unbundling, Identifying the Desired Textual Features, Generating New Content with the Extracted Features, Extracting Specific Textual Features with LLMs.

**Unit – IV**

**Standard Practices for Text Generation with LLM- Part-B** Chunking Text, Benefits of Chunking Text, Scenarios for Chunking Text, Poor Chunking Example, Chunking Strategies, Sentence Detection Using SpaCy, building a Simple Chunking Algorithm in Python, Sliding Window Chunking, Text Chunking Packages, Text Chunking with Tiktoken, Encodings, Understanding the Tokenization of Strings, Hallucinations in LLM, Context window.

**Unit – V**

**Vector Databases with FAISS and Pinecone:** Retrieval Augmented Generation (RAG), Introducing Embeddings, Document Loading, Memory Retrieval with FAISS, RAG with Lang Chain, Hosted Vector Databases with Pinecone, SelfQuerying, Alternative Retrieval Mechanisms, Evaluating AI Applications.

**Text Books:**

1. Phoenix J, Taylor M. Prompt Engineering for Generative AI. 2024, O'Reilly Media, Inc.
2. Chip Huyen, AI Engineering, 2025, O'Reilly Media, Inc.

**Reference Books:**

1. Tunstall L, Von Werra L, Wolf T. Natural language processing with transformers. 2022, O'Reilly Media, Inc.
2. Foster D. Generative deep learning. 2022, O'Reilly Media, Inc.

**M. Tech. II Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58300OE | Intellectual Property Rights (Open Elective) | 3 | 0 | 0 | 3 |

**Course Description:** This course deals with the fundamentals, legal frameworks, and protection mechanisms of Intellectual Property Rights, including trademarks, copyrights, patents, and trade secrets, along with recent national and international developments.

**Course Outcomes:** After completion of this course, the student will be able to

- C300.1 Distinguish and explain various forms of Intellectual Property Rights.
- C300.2 Identify appropriate forms of IPR protection for different types of intellectual works.
- C300.3 Apply statutory provisions to protect intellectual property under relevant IPR laws.
- C300.4 Analyze legal frameworks governing trademarks, copyrights, patents, and trade secrets.
- C300.5 Evaluate issues related to infringement, unfair competition, and enforcement of IPRs.
- C300.6 Assess recent developments and international perspectives in intellectual property law.

**Unit – I**

**Introduction to Intellectual property:** Introduction, types of intellectual property, international organizations, agencies and treaties, importance of intellectual property rights.

**Unit – II**

**Trade Marks:** Purpose and function of trademarks, acquisition of trade mark rights, protectable matter, selecting, and evaluating trade mark, trade mark registration processes.

**Unit – III**

**Law of copyrights:** Fundamental of copyright law, originality of material, rights of reproduction, rights to perform the work publicly, copyright ownership issues, copyright registration, notice of copyright, international copyright law.

**Law of patents:** Foundation of patent law, patent searching process, ownership rights and transfer.

**Unit – IV**

**Trade Secrets:** Trade secret law, determination of trade secret status, liability for misappropriations oftrade secrets, protection for submission, trade secret litigation. Unfair competition: Misappropriation right of publicity, false advertising.

**Unit – V**

**New development of intellectual property:** new developments in trade mark law; copyright law, patent law, intellectual property audits. International overview on intellectual property, international – trade mark law, copyright law, international patent law, and international development in trade secrets law.

**Text Books:**

1. Intellectual property right, Deborah. E. Bouchoux, Cengage learning.
2. Intellectual property right – Unleashing the knowledge economy, prabuddha ganguli, Tata McGraw Hill Publishing company Ltd

**M. Tech. II Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58301OE** | **Generative AI (Open Elective)** | 3 | 0 | 0 | 3 |

**Course Description:** This course provides an overview of Generative AI, covering the foundations of AI, ML, DL, NLP, and advanced neural architectures such as GANs, Transformers, and Large Language Models. It also explores prompt engineering, multi-agent systems, multimodal generative AI applications, and ethical considerations for responsible AI deployment.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C301.1 | Explain the foundations of AI, generative models, and advanced neural architectures. |
| C301.2 | Apply generative AI techniques for text, image, video, and multimodal applications. |
| C301.3 | Design and fine-tune large language models for domain-specific use cases. |
| C301.4 | Analyze ethical, social, and legal implications of generative AI systems. |
| C301.5 | Develop multi-agent and framework-based generative AI applications using LLM ecosystems. |
| C301.6 | Evaluate multimodal generative AI systems and propose responsible AI deployment strategies. |

**Unit – I**

**Foundations of AI and Generative Models** Introduction and historical evolution to Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP) and Deep Learning (DL)**,** Structure of Artificial Neural Networks (ANNs), Mathematical and computational foundations of generative modeling, Overview of generative models and their applications across various domains; Importance of Generative AI in modern applications, Transfer learning and in advancing Generative AI.

**Unit – II**

**Advanced Neural Architectures for Generative AI:** principles and applications, Generative Adversarial Networks (GANs): architecture and working principles; Transformer architecture and attention mechanisms (in detail); Long Short-Term Memory Networks (LSTMs) and the limitations of traditional RNNs/LSTMs, Advanced Transformer architectures and techniques, Pre-training and transfer learning strategies for generative models, Parameter-Efficient Fine-Tuning (PEFT), Low Rank Adaptation (LoRA).

**Unit – III**

**Large Language Models and Prompt Engineering:** Overview of Large Language Models (LLMs), GPT architecture, variants, and working principles, Prétraining and fine-tuning GPT models for applications (e.g., chatbots, text generation), Case study: GPT-based customer support chatbot, BERT architecture, pre-training objectives, and fine-tuning, RAG vs Fine-tuning comparison, Prompt Engineering: Designing effective prompts, controlling model behavior, and improving output quality, Fine-tuning language models for creative writing and chatbot development, Concept of Function Calling.

**Unit – IV**

**Multi-Agent Systems and Generative AI Applications** Introduction to Multi-Agent Systems (MAS),Types of agents: reactive, deliberative, hybrid, and learning agents, Multi-agent collaboration and orchestration for generative tasks, Planning and Task Decomposition, Reasoning in Agentic Systems, Use cases: autonomous research assistants, cooperative creative generation, Frameworks and tools: AutoGen, CrewAI, Hugging GPT for LLM-powered multi-agent systems, Generative AI applications: Art, Creativity, Image/Video generation, Music composition, Healthcare, Finance, Real-world case studies and deployment challenges.

**Unit – V**

**GEN AI Ethics and LangChain Framework:** components and LLM application development, Retrieval-Augmented Generation (RAG), Embeddings, Indexing networks, and Vector databases, Generative AI across modalities: Text, Code, Image, and Video generation, Image and Video generation using GANs and VAEs, Multimodal Transformers, Multimodal Generative AI: integration and training strategies, Ethical considerations: bias, fairness, trust, and responsible AI deployment, Social and legal implications of Generative AI, Risk mitigation strategies and real-world ethical case studies.

**Text Books:**

1. Altaf Rehmani, Generative AI for Everyone: Understanding the Essentials and Applications of This Breakthrough Technology.
2. Charu C. Aggarwal, Neural Networks and Deep Learning: A Textbook, Springer.
3. Joseph Babcock, Raghav Bali, Generative AI with Python and TensorFlow 2, 2024, Packt Publishing Limited.
4. David Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose and Play, 2019, O'Reilly Media, Inc.

**Reference Books:**

1. Josh Kalin, Generative Adversarial Networks Cookbook, 2021, Packt Publishing Limited.
2. Jesse Sprinter, Generative AI in Software Development: Beyond the Limitations of Traditional Coding, 2024
3. Richard Steeve, Efficient Fine-Tuning of LLMs: Practical Techniques Using LoRA, QLoRA, and PEFT for Scalable and Cost-Effective NLP Model, 2025.
4. Maarten Balliauw, Daniel Cazzulino, Programming Large Language Models with Azure OpenAI, 2024, Pearson.
5. Sinan Ozdemir, Building Agentic AI Systems, 2024, Pearson.
6. Akansha Singh, Krishna Kant Singh, Multimodal Generative AI, 2024, Springer Nature.

**M. Tech. II Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| 58302OE | Intrusion Detection Systems (Open Elective) | 3 | 0 | 0 | 3 |

**Course Description:** This course compares alternative tools and approaches for Intrusion Detection through quantitative analysis to determine the best tool or approach to reduce risk from intrusion.

**Course Outcomes:** After completion of this course, the student will be able to

- C302.1 Explain the fundamentals of intrusion detection and intrusion prevention systems.
- C302.2 Identify network, application, and human-layer attacks and attacker classifications.
- C302.3 Analyze signature-based and anomaly-based intrusion detection algorithms
- C302.4 Apply IDS tools and rule-based techniques for detecting malicious activities.
- C302.5 Evaluate malware, botnet, and insider threat detection mechanisms.
- C302.6 Assess alert correlation, attack modelling, and future trends in collaborative security.

**Unit – I**

The state of threats against computers, and networked systems-Overview of computer security solutions and why they fail-Vulnerability assessment, firewalls, VPN's -Overview of Intrusion Detection and Intrusion Prevention, Network and Host-based IDS, EDR/XDR (Endpoint & Extended Detection and Response), Ransomware basics.

**Unit – II**

Classes of attacks - Network layer: scans, denial of service, penetration Application layer: software exploits, code injection-Human layer: identity theft, root access-Classes of attackers-Kids/hackers/sop Hesitated groups-Automated: Drones, Worms, Viruses, Ransomware Attack Vectors and the Threat Landscape**.**

**Unit – III**

A General IDS model and taxonomy, Signature-based Solutions, Suricata, Suricata rules, Evaluation of IDS, Cost sensitive IDS, Understanding SIEM, Splunk Integration with Other Security Tools, Security Intelligence.

**Unit – IV**

Anomaly Detection Systems and Algorithms-Network Behavior Based Anomaly Detectors (rate based)-Host-based Anomaly Detectors-Software Vulnerabilities-State transition, Immunology, Payload Anomaly Detection, User and Entity Behavior Analytics (UEBA).

**Unit – V**

Attack trees and Correlation of alerts- Autopsy of Worms and Botnets-Malware detection - Obfuscation, polymorphism- Document vectors. Email/IM security issues-Viruses/Spam-From signatures to thumbprints to zero day detection-Insider Threat issues-Taxonomy-Masquerade and Impersonation Traitors, Decoys and Deception-Future: Collaborative Security, SOAR (Security Orchestration, Automation, and Response) framework, Behavioral analytics.

**Text Books:**

1. Peter Szor, The Art of Computer Virus Research and Defense, Symantec Press.
2. Markus Jakobsson and Zulfikar Ramzan, Crimeware, Understanding New Attacks and Defenses, Addison-Wesley Professional.

**Reference Books:**

1. Saiful Hasan, Intrusion Detection System, Kindle Edition.
2. Ankit Fadia, Intrusion Alert: An Ethical Hacking Guide to Intrusion Detection, Chaukhamba Auriyantaliya.
3. Guven Boyraz, Endpoint Detection and Response Essentials, 2024, Packt Publishing.
4. Roger A. Grimes, Windows Ransomware Detection and Protection, 2021, Packt Publishing.
5. Jit Sinha, Ultimate Splunk for Cybersecurity, 2024, Orange Education Pvt Ltd.
6. Gerardus Blokdyk, UEBA – A Complete Guide, 2020, The Art of Service.
7. Jason Edwards, Adversarial Machine Learning: Mechanisms, Vulnerabilities, and Strategies for Trustworthy AI, 2026, Wiley.
8. Benjamin Kovacevic, Security Orchestration, Automation, and Response for Security Analysts, 2023, Packt Publishing.

**M. Tech. II Year I Sem**

| Course Code | Course Title | L | T | P | Credits |
|---|---|---|---|---|---|
| **58303OE** | **Digital Forensics (Open Elective)** | 3 | 0 | 0 | 3 |

**Pre - Requisite** Computer Networks

**Course Description:** This course introduces the principles, tools, and investigative techniques used in digital and computer forensics for identifying, collecting, and analyzing electronic evidence. It also covers cybercrime investigation processes, forensic imaging, memory and network forensics, mobile forensics, and relevant legal frameworks such as the IT Act and DPDP Act.

**Course Outcomes:** After completion of this course, the student will be able to

| | |
|---|---|
| C303.1 | Understand relevant legislation and ethical codes to ensure legal compliance in digital investigations. |
| C303.2 | Illustrate the holistic investigative process and detective procedures for various categories of computer crime |
| C303.3 | Execute standard protocols for the search, seizure, and documentation of electronic evidence for court proceedings. |
| C303.4 | Implement evidence management strategies and forensic mindsets to maintain the integrity of the chain of custody. |
| C303.5 | Summarize specialized computer and network forensic investigations using industry-standard workstations and open-source tools. |
| C303.6 | Discover mobile forensic techniques and recent trends to retrieve data from contemporary digital devices. |

**Unit – I**

**Digital Forensics Science:** Forensics science, computer forensics, and digital forensics.

**Computer Crime:** Criminalistics as it relates to the investigative process, analysis of cyber criminalistics area, holistic approach to cyber-forensics, NIST Incident Response Lifecycle, NIST Digital Forensics**,** Framework (NIST SP 800-86).

**Unit – II**

**Cyber Crime Scene Analysis:** Discuss the various court orders etc., methods to search and seizure electronic evidence, retrieved and un-retrieved communications, Discuss the importance of understanding what court documents would be required for a criminal investigation, Forensic Imaging Techniques, FTK Imager and Autopsy.

**Unit – III**

**Evidence Management & Presentation:** Create and manage shared folders using operating system, importance of the forensic mindset, define the workload of law enforcement, Explain what the normal case would look like, Define who should be notified of a crime, parts of gathering evidence, Define and apply probable cause, Forensic Report Writing standards.

**Unit – IV**

**Computer Forensics:** Prepare a case, Begin an investigation, understand computer forensics workstations and software, Conduct an investigation, Complete a case, Critique a case.
**Memory Forensics:** Volatility Framework, Ransomware Investigations.
**Network Forensics:** open-source security tools for network forensic analysis, requirements for preservation of network data.

**Unit – V**

**Mobile Forensics:** mobile forensics techniques, mobile forensics tools, Recent trends in mobile forensic technique and methods to search and seizure electronic evidence.
**Legal Aspects of Digital Forensics**: IT Act 2000, amendment of IT Act 2008, Introduction to Digital Personal Data Protection (DPDP) Act.

**Text Books:**

1. John Sammons, The Basics of Digital Forensics, Elsevier.
2. John Vacca, Computer Forensics: Computer Crime Scene Investigation, Laxmi Publications.

**Reference Books:**

1. William Oettinger, Learn Computer Forensics: A beginner's guide to searching, analyzing, and securing digital evidence, Packt Publishing; 1st edition pril 2020), ISBN : 1838648178.
2. Thomas J. Holt , Adam M. Bossler, Kathryn C. Seigfried-Spellar , Cybercrime and Digital Forensics: An Introduction, Routledge
3. Kenneth C.Brancik , Insider Computer Fraud, 2008,Auerbach Publications Taylor & Francis Group.
4. Bill Nelson, Amelia Phillips, Christopher Steuart, Guide to Computer Forensics and Investigations, 4e, Cengage Learning.
5. NIST SP 800-86 Guide to Integrating Forensic Techniques into Incident Response, 2006.
6. Michael Ligh, Andrew Case, Jamie, Walters, The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory, 2014, Wiley.
7. Gerard Johansen, Digital Forensics and Incident Response - Third Edition,2022 O'Reilly.
8. https://dpdpa.com/index.html