**NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR**
**(An Autonomous Institute)**

**Affiliated to**

**DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, UTTAR PRADESH LUCKNOW**

# Evaluation Scheme & Syllabus

For

## B.Tech in Computer Science & Engineering (Collaboration &Twinning)

## Second Year

**(Effective from the Session: 2026-27)**

# NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY, GREATER NOIDA

## (An Autonomous Institute)

## B. TECH (CSE-C&T)

## Evaluation Scheme

### SEMESTER III

| Sl. No. | Subject Codes | Subject Name | Type of Subject | Periods | | | Evaluation Schemes | | | | End Semester | | Total | Credit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | L | T | P | CT | TA | TOTAL | PS | TE | PE | | |
| 1 | CCSCC0301 | Employability Skill Development - I | Mandatory | 2 | 0 | 0 | 60 | 40 | 100 | | | | 100 | 2 |
| 2 | CASL0301 | Technical Communication | Mandatory | 2 | 0 | 0 | 30 | 20 | 50 | | 50 | | 100 | 2 |
| 3 | CCSE0303A | Operating Systems | Mandatory | 2 | 0 | 0 | 30 | 20 | 50 | | 50 | | 100 | 2 |
| 4 | CCSE0301 | Data Structures and Algorithm-II | Mandatory | 3 | 0 | 0 | 30 | 20 | 50 | | 100 | | 150 | 3 |
| 5 | CCSAI0301 | Artificial Intelligence | Mandatory | 2 | 0 | 0 | 30 | 20 | 50 | | 50 | | 100 | 2 |
| 6 | CCSE0304 | Computer Architecture & Parallel Processing | Mandatory | 3 | 0 | 0 | 30 | 20 | 50 | | 100 | | 150 | 3 |
| 7 | CCSE0353A | Operating Systems Lab | Mandatory | 0 | 0 | 4 | | | | 50 | | 50 | 100 | 2 |
| 8 | CCSE0351 | Data Structures and Algorithm-II Lab | Mandatory | 0 | 0 | 4 | | | | 50 | | 50 | 100 | 2 |
| 9 | CCSAI0351 | Artificial Intelligence lab | Mandatory | 0 | 0 | 2 | | | | 25 | | 25 | 50 | 1 |
| 10 | CCSE0355 | Object Oriented Techniques using Java | Mandatory | 0 | 0 | 6 | | | | 50 | | 100 | 150 | 3 |
| 11 | CCSE359 | Internship-I | Mandatory | 0 | 0 | 2 | | | | 50 | | | 50 | 1 |
| 12 | CNC0301/ CNC0302 | Artificial Intelligence and Cyber Ethics / Environmental Science | Compulsory Audit | 2 | 0 | 0 | 30 | 20 | 50 | | | | 50 | NA |
| 13 | | MOOCs (For B.Tech. Hons. Degree) | *MOOCs | | | | | | | | | | | |
| | | **TOTAL** | | **16** | **0** | **18** | **240** | **160** | **350** | **225** | **350** | **225** | **1150** | **23** |

*** List of MOOCs Based Recommended Courses for Second year B. Tech Students**

| S. No. | Subject Code | Course Name | University/ Industry Partner Name | N. of Hours | Credits |
|---|---|---|---|---|---|
| **1.** | CMC0035 | Data Structures and Algorithms using Python - Part 2 | Infosys Wingspan (Infosys Springboard) | 34h 41m | 2.5 |
| **2.** | CMC0027 | Object Oriented Programming Using Python | Infosys Wingspan (Infosys Springboard) | 46h 13m | 3.5 |
| **3.** | CMC0036 | RH294 - Red Hat Enterprise Linux Automation with Ansible | RedHat | 96 h | 4 |
| **4.** | CMC0037 | Operating Systems & Shell Scripting | NASSCOM | 6h | 0.5 |

**PLEASE NOTE:**

A 3-4 weeks Internship shall be conducted during summer break after semester-II and will be assessed during semester-III

Compulsory Audit (CA) Courses (Non-Credit - CNC0301/CNC0302)

- ➢ All Compulsory Audit Courses (a qualifying exam) do not require any credit.
- ➢ The total and obtained marks are not added in the grand total.

**Abbreviation Used:**

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam. CE: Core Elective, OE: Open Elective, DE: Departmental Elective, PE: Practical End Semester Exam, CA: Compulsory Audit, MOOCs: Massive Open Online Courses.

# NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY, GREATER NOIDA

## (An Autonomous Institute)

## B. TECH (CSE-C&T)

## Evaluation Scheme

### SEMESTER IV

| Sl. No. | Subject Codes | Subject Name | Types of Subject | Periods | | | Evaluation Schemes | | | | End Semester | | Total | Credit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | L | T | P | CT | TA | TOTAL | PS | TE | PE | | |
| 1 | CASCC0401 | Employability Skill Development - II | Mandatory | 2 | 0 | 0 | 60 | 40 | 100 | | | | 100 | 2 |
| 2 | CCSE0402 | Software Engineering and Design | Mandatory | 3 | 0 | 0 | 30 | 20 | 50 | | 100 | | 150 | 3 |
| 3 | CCSE0401 | Database Management Systems | Mandatory | 2 | 0 | 0 | 30 | 20 | 50 | | 50 | | 100 | 2 |
| 4 | CCSE0403 | Theory of Computation | Mandatory | 3 | 0 | 0 | 30 | 20 | 50 | | 100 | | 150 | 3 |
| 5 | CCSE0404 | Machine Learning Techniques | Mandatory | 3 | 0 | 0 | 30 | 20 | 50 | | 100 | | 150 | 3 |
| 6 | CAS0401 | Statistics and Probability | Mandatory | 3 | 1 | 0 | 30 | 20 | 50 | | 100 | | 150 | 4 |
| 7 | CCSE0452 | Software Engineering & Design lab | Mandatory | 0 | 0 | 2 | | | | 25 | | 25 | 50 | 1 |
| 8 | CCSE0451 | Database Management Systems Lab | Mandatory | 0 | 0 | 4 | | | | 50 | | 50 | 100 | 2 |
| 9 | CCSE0455 | Web Technology | Mandatory | 0 | 0 | 6 | | | | 50 | | 100 | 150 | 3 |
| 10 | CCSE0459 | Mini Project | Mandatory | 0 | 0 | 2 | | | | 50 | | | 50 | 1 |
| 11 | CCSCC0452 | Problem Solving Approaches | Mandatory | 0 | 0 | 2 | | | | 50 | | | 50 | 1 |
| 12 | CNC0402/ CNC0401 | Environmental Science / Artificial Intelligence and Cyber Ethics | Compulsory Audit | 2 | 0 | 0 | 30 | 20 | 50 | | | | 50 | NA |
| | | MOOCs (For B.Tech. Hons. Degree) | *MOOCs | | | | | | | | | | | |
| | | **TOTAL** | | **18** | **1** | **16** | **240** | **160** | **350** | **225** | **450** | **175** | **1200** | **25** |

*** List of MOOCs Based Recommended Courses for Second year B. Tech Students**

| S. No. | Subject Code | Course Name | University/ Industry Partner Name | N. of Hours | Credits |
|---|---|---|---|---|---|
| **1.** | CMC0058 | Agile Scrum Certification | Infosys Wingspan (Infosys Springboard) | 32h 45 m | 2.5 |
| **2.** | CMC0059 | Database Management System – Part 1 | Infosys Wingspan (Infosys Springboard) | 63h 50m | 4 |
| **3.** | CMC0060 | Twitter Bootstrap | Infosys Wingspan (Infosys Springboard) | 23h | 1.5 |
| **4.** | CMC0057 | SQL Fundamentals | NASSCOM | 8h | 0.5 |

**PLEASE NOTE:**
A 3-4 weeks Internship shall be conducted during summer break after semester-II and will be assessed during semester-III
Compulsory Audit (CA) Courses (Non-Credit - CNC0402/CNC0401)

- ➢ All Compulsory Audit Courses (a qualifying exam) do not require any credit.
- ➢ The total and obtained marks are not added in the grand total.

**Abbreviation Used:**

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam.
CE: Core Elective, OE: Open Elective, DE: Departmental Elective, PE: Practical End Semester Exam, CA: Compulsory Audit,
MOOCs: Massive Open Online Courses.

# NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY, GREATER NOIDA
## (An Autonomous Institute)

**AICTE Guidelines in Model Curriculum:**

A student will be eligible to get Under Graduate degree with Honours only, if he/she completes the additional MOOCs courses such as Infosys Springboard/NASSCOM/CISCO (as applicable) certifications, or any other online courses recommended by the Institute (Equivalent to 20 credits). During Complete B.Tech. Program Guidelines for credit calculations are as follows.

1. For 6 to 12 Hours =0.5 Credit
2. For 13 to18 =1 Credit
3. For 19 to 24 =1.5 Credit
4. For 25 to 30 =2 Credit
5. For 31 to 35 =2.5 Credit
6. For 36 to 41 =3 Credit
7. For 42 to 47 =3.5 Credit
8. For 48 and above =4 Credit

For registration to MOOCs Courses, the students shall follow Infosys Springboard/NASSCOM/CISCO (as applicable) registration details as per the assigned login and password by the Institute these courses may be cleared during the B. Tech degree program (as per the list provided). After successful completion of these MOOCs courses, the students shall provide their successful completion status/certificates to the Controller of Examination (COE) of the Institute through their coordinators/Mentors only.

The students shall be awarded Honours Degree as per following criterion.

i. If he / she secures 7.50 as above CGPA.
ii. Passed each subject of that degree program in the single attempt without any grace.
iii. Successful completion of MOOCs based 20 credits

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY
## GREATER NOIDA-201306
### (An Autonomous Institute)
### School of Computer Science and Information Technology

| Course Code: CCSCC0301 | Course Name: Employability Skill Development - I | L | T | P | C |
|---|---|---|---|---|---|
| **Course Offered in:** B.Tech III Semester<br>**Note:** Section C - offered in B.Tech. III/IV SEM | | 2 | 0 | 0 | 2 |

**Pre-requisite: Students should have successfully completed communication skills courses in the first year.**

**Course Objectives:**

**Section A:** Develop understanding of computer systems, mathematical foundations, and programming concepts while applying structured problem-solving techniques to design logical solutions effectively.

**Section B:** Enhance numerical ability and logical reasoning skills through speed mathematics, number systems, and analytical problem-solving techniques for competitive aptitude development.

**Section C:** To develop the ability to communicate effectively and empathetically in workplace situations while strengthening active listening, adaptability, and cross-cultural awareness for navigating complex professional environments.

| Course Outcome | | Bloom's Knowledge Level (KL) |
|---|---|---|
| **CO1** | Apply fundamental concepts of computer systems, mathematical foundations, and programming constructs to analyze and solve basic computational problems. | **K3** |
| **CO2** | Design and develop structured solutions for problem statements by applying SDLC principles, logical thinking, and flowchart/pseudocode techniques. | **K6** |
| **CO3** | Apply number theory and logical reasoning concepts to solve quantitative and analytical problems efficiently. | **K3** |
| **CO4** | Apply spoken English skills to participate effectively in standard workplace routines, including daily stand-ups, reporting, and collaborative problem-solving. | **K3** |
| **CO5** | Evaluate and adapt communication styles to suit dynamic workplace communication situations and deliver context-sensitive, and logically structured verbal responses under time constraints. | **K6** |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| CO-PO Mapping | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 | PSO4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | - | 2 | 3 | 1 | - | 3 | 2 | 2 | 2 | 1 | 1 | 2 | 1 |
| **CO2** | 3 | 3 | 3 | 3 | 3 | 2 | 1 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 3 |
| **CO3** | 3 | 3 | 2 | 1 | - | - | - | 1 | 2 | 2 | 2 | 3 | 2 | 2 | 3 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO4** | - | - | - | - | 1 | 2 | - | 3 | 2 | 2 | - | - | - | - | 1 |
| **CO5** | - | 2 | 1 | - | 1 | 2 | - | 3 | 2 | 2 | 1 | - | - | - | 1 |

**Course Contents / Syllabus**

**Section-A**

| **Module 1** | **Foundations of Computer Systems and Mathematical Concepts** | **8 hours** |
|---|---|---|

**Computer System Fundamentals:** Introduction to Assembler, Compiler, Interpreter, Role of Loader and Linker in program execution.
**Mathematical Foundations for Computing:** Sets, Relations, and Functions: definitions and applications, Principle of Mathematical Induction and its use in proofs.
**Basic Concepts of Programming Language:** Memory model of C/C++ Program, Activation Record of a function, L-Value/ R-Value concept. Operators, Precedence, Order of Evaluation and Associativity. Pointers and Dynamics Memory Concept, Functions (argument passing, concept of scope), Understanding Recursion, Bitwise Operations.

| **Module 2** | **Design and develop structured solutions for problem statements** | **8 hours** |
|---|---|---|

Introduction to Software Development Life Cycle, Step-by-step solution to simple problems, Developing logic/flowchart/pseudocode. Stepwise refinement and Procedural Abstraction.
**Case Study*:** Understand the problem statement, identify a feasible solution, and develop the logic along with a flowchart for the proposed solution.

**Section-B**

| **Module 3:** | **Speed Math and Number System** | **16 hours** |
|---|---|---|

Classification of number, Divisibility Rule, Factorization, HCF & LCM, It's Application, Unit digit (Cyclicity), Last two digit, Remainder theorem, Factorial and Number of zeroes, Highest power.
**Analytical and Logical Reasoning**: Direction and Sense, Blood Relation, Number Series and Letter Series, Coding Decoding

**Section-C**

| **Module-4** | **Workplace Interaction Essentials** | **12 hours** |
|---|---|---|

- The Professional Shift: Navigating the transition from academic to professional tone. Using appropriate register when addressing peers vs. supervisors.
- Active Listening & The Feedback Loop: Moving beyond passive hearing. Practicing the SBI (Situation-Behaviour-Impact) model for peer feedback and clarifying multi-step technical instructions.
- Structured Micro-Speaking: The art of the concise update. Delivering 1-2 minute "Stand-up" reports (What I did, What I will do, Blockers) and explaining basic technical processes to laymen.
- Guided Interpersonal Simulations: Role-playing everyday scenarios: politely interrupting a meeting, seeking assistance on a delayed task, and confirming understanding of complex briefs using paraphrasing techniques.
- Digital & Hybrid Nuances: Verbal signalling in virtual meetings, managing "overtalk," and translating concise written chat messages into spoken professional updates.

| **Module-5** | **Workplace Communication in Action** | **12 hours** |
|---|---|---|

- Dynamic Problem-Based Communication: Simulating the "War Room." Explaining sudden technical failures, proposing immediate solutions, and negotiating revised deadlines with stakeholders.
- Listening Under Pressure & Global Adaptability: Interpreting fast, accented, or incomplete instructions (incorporating CEFR-level listening benchmarks). Decoding tone, urgency, and indirect cultural communication styles.
- Collaborative Conflict Resolution: Managing disagreements in team settings using "I-statements" and the "Yes, and…" technique to build consensus without escalating tension.

- The Persuasive Professional: Delivering evidence-based arguments using the PREP (Point, Reason, Example, Point) framework to justify a technical choice or resource allocation.
- Real-Time "Crisis" Simulation (Capstone): A comprehensive role-play where student groups are given a sudden project change or client issue. They must conduct an emergency meeting, delegate tasks verbally, and present a united response to the "client."

**Total: 56 Hours**

**A list of problem statements will be provided by the mentor; you may choose anyone.*
***Note:** Students may choose an alternative problem statement with prior approval from the mentor.*

**Reference Books:**

| S. No. | Book Title | Author |
|---|---|---|
| 1. | A Project-Based Introduction to Programming | Access Point Publishing |
| 2. | Programming: Principles and Practice Using C++ | Bjarne Stroustrup |
| 3. | Effective Modern C++ | Scott Meyers |
| 4. | Quantitative Aptitude | R S Aggarwal |
| 5. | Verbal & Non-Verbal Reasoning | R S Aggarwal |
| 6. | *Crucial Conversations Tools* | Kerry Patterson et al. |
| 7. | *HBR Guide to Persuasive Presentations* | Nancy Duarte |
| **Self-Study Content Links:** | | **Duration (in Hrs)** |
| **MOOCs** | https://swayam-plus.swayam2.ac.in/courses/course-details?id=F_WADHW_042 | **105 hours** |
| **Video Lectures** | Employability Skills Training Session - YouTube | **1 hour** |
| **Web References** | https://dde.manuu.edu.in/sites/default/files/DDE/DDE-Assignments/Diploma-Sem1-2024/Employability-Skills-1.pdf | |

**Mode of Evaluation**

| CIE | | | | ESE | Total |
|---|---|---|---|---|---|
| **Section-A (ST)** | **Section-B (ST)** | **Section-C (ST)** | **Attendance 10** | | |
| **30** | **30** | **30** | **10** | | **100** |

**NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**GREATER NOIDA-201306**

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| **Course Code: CASL0301** | **Course Name: Technical Communication** | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|
| **Course Offered in:** B. Tech. All branches (except CSBS) | | **2** | **0** | **0** | **2** |

**Pre-requisite: Intermediate level (CEFR) and above**

**Course Objectives:**

**Demonstrate effective verbal and non-verbal communication skills** in diverse professional settings, including meetings, presentations, and interpersonal interactions.

**Develop and apply clear, concise, and audience-appropriate written communication**, such as emails, letters, memos, resume', using correct grammar, tone, and format.

**Adapt communication style based on cultural, organizational, and situational contexts** to foster inclusive and respectful professional relationships.

**Employ digital communication tools and platforms** (e.g., video conferencing, business messaging apps) responsibly and effectively in remote or hybrid work environments.

| **Course Outcome:** After completion of the course, the student will be able to | | **Bloom's Knowledge Level (KL)** |
|---|---|---|
| **CO1** | Comprehend the principles and functions of technical communication. | K2 |
| **CO2** | Write for specific audience and purpose to fulfil the provided brief | K3 |
| **CO3** | Recognize and produce different kinds of technical documents. | K3 |
| **CO4** | Apply effective speaking skills to efficiently carry out official discourses. | K3 |
| **CO5** | Demonstrate their understanding of communication through digital media. | K3 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| **CO-PO Mapping** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 3 | 1 | 1 | - | - | - | - |
| **CO2** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | - | 1 | 1 |
| **CO3** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 2 | 1 | 2 | 2 |
| **CO4** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 2 | 2 |
| **CO5** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 2 |

| **Course Contents / Syllabus** | | |
|---|---|---|
| **Module 1** | **Introduction to Technical Communication** | **4 hours** |
| **Technical Communication**: Definition, Process, Types, Levels, and Flow; **Barriers to Technical Communication:** emphasis on gender neutral language and cultural sensitivity; **Significance of audience in technical communication** | | |
| **Module 2** | **Technical Writing 1** | **5 hours** |
| **Technical writing skill**: characteristics, examples; **Business letters/emails**: Content organization, Tone and intent; **Agenda & Minutes of Meetings** | | |
| **Module 3** | **Technical Writing 2** | **5 hours** |
| **Job application**, **Resume'**; **Report, proposal**; **Technical paper**: Abstract; **Ethical Writing:** Copy Editing, Referencing and Plagiarism | | |
| **Module 4** | **Public Speaking** | **6 hours** |
| **Components of effective speaking**: Simplicity, order, balance in arranging ideas. Importance of *KOPPACT;* **Appearing for a job interview:** FAQs; **Telephonic & Online Interviews** | | |
| **Module 5** | **Virtual/Remote Communication** | **4 hours** |
| **Remote work**: online platforms; **Video conferencing**; **Virtual etiquette**: email ids, usernames; **Writing Blogs & creating Vlogs** | | |
| | | **Total: 24 hours** |

**Textbook:**

| **SNO** | **Books Title** | **Author** |
|---|---|---|
| **1** | Technical Communication – Principles and Practices, 4th Edition , Oxford Univ. Press, 2022, New Delhi. | Meenakshi Raman & Sangeeta Sharma |

**Reference Books:**

| **S.NO** | **Books Title** | **Author** |
|---|---|---|
| **1** | Technical Communication, 15th Edition , Pearson, 2021 | John M. Lannon & Laura J. Gurak |
| **2** | Spoken English- A Manual of Speech and Phonetics (5th Edition) by , 2024, New Delhi. | R K Bansal & J B Harrison, Orient Blackswan |
| **3** | Business Correspondence and Report Writing (6 Edition), Tata McGraw Hill & Co. Ltd., 2020, New Delhi. | Prof. R C Sharma, Krishna Mohan, and Virendra Singh Nirban |
| **4** | Intercultural Communication in Virtual Exchange , Cambridge Univ. Press, 2024. | Francesca Helm |

| **Self-Study Content Links:** | | **Duration (in Hrs)** |
|---|---|---|
| **MOOCs** | https://onlinecourses.nptel.ac.in/noc24_ge37/preview<br>https://archive.nptel.ac.in/courses/109/106/109106094/ | 4 weeks |
| **Video Lectures** | https://www.youtube.com/watch?v=kOJlwMJxEG0&t=8s<br>https://www.youtube.com/watch?v=Sg7Q_dC_fWU&list=PLPuC5CMHiqmuzq_KQ4aw0V9Q7xJY6aezb<br>https://www.youtube.com/watch?v=ymLFJDpjgCk&list=PLPuC5CMHiqmuzq_KQ4aw0V9Q7xJY6aezb&index=6 | 2.5 hours |

| | | |
|---|---|---|
| **Web References** | Technical Communication And Soft Skills For Engineers | |

**Mode of Evaluation**

| CIE | | | | | | ESE | Total |
|---|---|---|---|---|---|---|---|
| **ST1** | **ST2** | **ST3** | **TA1 5** | **TA2 5** | **Attendance 10** | | |
| **30** | | | **20** | | | **50** | **100** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| Course Code: CCSE0303A | Course Name: Operating Systems | L | T | P | C |
|---|---|---|---|---|---|
| **Course Offered in: CSE/CSE-R/IT/CS/CSE(AI)/CSE(AIML)/ IOT/DS/CYS/M.Tech.Int./ CSE-TWIN/IT-TWIN** | | **2** | **0** | **0** | **2** |

**Pre-requisite:** Basic knowledge of computer fundamentals, C programming, Data structures and Computer organization**.**

**Course Objectives:** The objective of the course is to provide a foundational understanding of operating system concepts, including system architecture, process and thread management, concurrency, deadlock, resource management, memory and file systems, Linux shell scripting, and an introduction to virtualization and distributed systems.

| **Course Outcome:** After completion of the course, the student will be able to | | **Bloom's Knowledge Level (KL)** |
|---|---|---|
| **CO1** | Understand operating system architecture and types, and use the Linux CLI for basic Operations. | K2 |
| **CO2** | Implement the CPU scheduling algorithms including uses of multithreading models. | K3 |
| **CO3** | Implement concurrency control, process synchronization techniques, and deadlock handling techniques. | K3 |
| **CO4** | Implement memory management strategies to optimize system performance and analyze file systems and configure virtual machines in modern operating systems. | K3 |
| **CO5** | Analyze file systems and configure distributed systems and virtual machines in modern operating systems. | K4 |

| CO - PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 | PSO4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | 2 | 1 | 2 | - | - | - | - | 1 | 1 | 2 | 1 | 2 | 2 |
| **CO2** | 3 | 3 | 3 | 2 | 2 | - | - | - | - | 1 | 1 | 2 | 2 | 1 | 2 |
| **CO3** | 3 | 3 | 3 | 2 | 2 | - | - | - | - | 1 | 1 | 2 | 3 | 3 | 2 |
| **CO4** | 3 | 3 | 3 | 2 | 2 | - | - | - | - | 1 | 2 | 2 | 2 | 2 | 2 |
| **CO5** | 3 | 3 | 2 | 3 | 3 | 2 | 1 | - | 2 | 2 | 2 | 3 | 3 | 3 | 2 |

**Course Contents / Syllabus**

| **Unit 1** | **Fundamentals & Shell scripting** | **7 hours** |
|---|---|---|

**Fundamentals of Operating Systems:** Overview of Operating Systems, Generations of OS, Operating system architecture, Interrupt handling, System call and kernel, Types of Operating System: Batch OS, Multi-programming OS, Multitasking OS, Multiprocessor OS, Real time OS.

**Shell Scripting in Linux:** Introduction to Linux Operating System & Architecture, Basic Command Line Interface (CLI) Operations in Linux, Shell Scripting Basics: Variables, Control Structures, Functions.

**Case study:** Automating system administration tasks using shell scripts in Linux.

| **Unit 2** | **Process & Thread Management** | **7 hours** |
|---|---|---|

**Process Management**: Process, Transition Diagram, Process Control Block (PCB), Types of Schedulers: Long Term, Mid Term, Short Term Scheduler,

**CPU Scheduling:** Pre-emptive and Non-Pre-emptive Algorithm (FCFS, SJF, SRTF, Non-Pre-emptive Priority, Pre-emptive Priority, Round Robin, Multilevel Queue Scheduling and Multilevel Feedback Queue Scheduling)

**Thread:** Processes Vs Threads, Thread states, Benefits of threads, Types of threads, Multithreading Models, Concept of Hyper-Threading

**Case study:** Analyse and implement CPU Scheduling in Real-Time Embedded Systems and RTOS.

| **Unit 3** | **Concurrency and Deadlock Management** | **8 hours** |
|---|---|---|

**Concurrency:** Introduction of Concurrency, Types of Process, Race Condition, Critical Section, Inter Process Communication, Producer consumer problem.

**Process Synchronization**: Lock variable, Peterson's Solution, Strict alternation, Lamport Bakery Solution, Test and set lock, Semaphore- counting, binary and monitor,

**Classical Problem of Synchronization:** Bound Buffer, Dinning Philosopher, Reader writer, Sleeping barber.

**Deadlock:** Deadlock, Deadlock characterization, Deadlock Prevention, Deadlock Avoidance: Bankers Algorithms, Deadlock Detection, Recovery from Deadlock.

**Case study:** Deadlock avoidance in database transaction management systems like Oracle or MySQL.

| **Unit 4** | **Memory Management** | **8 hours** |
|---|---|---|

**Memory Management:** Memory Management function, Loading and linking Address Binding, Memory management techniques, Contiguous technique- Fixed Partitions, variable partitions, Memory Allocation: Allocation Strategies (First Fit, Best Fit, and Worst Fit), Non-contiguous, Paging, Segmentation, Segmented paging,
**Virtual Memory:** Virtual Memory Concepts, Demand Paging, Performance of Demand Paging, Page Replacement Algorithms: FIFO, LRU, Optimal and LFU, Belady's Anomaly, Thrashing

**File and Disk Management:** File Access Mechanism, File Allocation Method, Disk Scheduling Algorithms

**Modern Operating System:** Virtual machines – hypervisor, Introduction to GPU. Introduction of CUDA Programming.
**Case study:** Virtual memory management in modern OS like Windows 10 and how paging impacts performance.

| **Total Lecture Hours** | **30 hours** |
|---|---|

**Textbook:**

| **S.No** | **Book Title** | **Author** |
|---|---|---|
| **1.** | " Operating System Concepts Essentials" , Willey Publication,10th Edition,2018. | Abraham Silberschatz, Peter Baer Galvin and Greg Gagne |

| | | |
|---|---|---|
| **2.** | "A practical guide to Linux: Commands, Editors and Shell Programming", CreateSpace Independent Publishing Platform, 4th Edition,2017. | Marks G. Sobell |
| **3.** | "LINUX for beginners", 1stEdtion,2014. | Jason Cannon |
| **Reference Books:** | | |
| **S.No** | **Book Title** | **Author** |
| **1.** | "Operating Systems: Internals and Design Principles", Pearson Education , 9th Edition, 2019. | William Stallings |
| **2.** | "Operating System: A Design-oriented Approach" , McGraw Hill Education,2017. | Charles Patrick Crowley |
| **3.** | "Learning Linux Shell Scripting", Packt Publishing ,2nd Edition 2018. | Ganesh Naik |

| **Self-Study Contents Links:** | | **Duration (in hours)** |
|---|---|---|
| **MOOCs** | https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013843258047987712344 43_shared/overview | **29 hours** |
| **Video Lectures** | https://www.youtube.com/watch?v=dTl9QkH4j8o&list=PL6CdojO56mZ3SeRfpzMBMObSnTziA0gfE<br>https://www.youtube.com/watch?v=Q35sMYHaIvQ<br>https://www.youtube.com/watch?v=rnGVincwk30<br>https://www.youtube.com/watch?v=akU1Ji8Vzdk&list=PLbMVogVj5nJRa3VKt_eyZdJ_DitCz1cvQ<br>https://www.youtube.com/watch?v=pPAKs7tT8sw<br>https://www.youtube.com/watch?v=hBPYP0ZEvS8<br>https://www.youtube.com/watch?v=UeVQxzlgd10<br>https://www.youtube.com/watch?v=9cuwgiHQyoI<br>https://www.youtube.com/watch?v=-NONm-Jq34Y&t=25s<br>https://www.youtube.com/watch?v=5uyGn147rr0<br>https://www.youtube.com/watch?v=utYC2tKDdoQ<br>https://www.youtube.com/watch?v=Cuo8GTklgoE<br>https://www.youtube.com/watch?v=YHQxp-XduS0<br>https://www.youtube.com/watch?v=yB8sm-pwvZo<br>https://www.youtube.com/watch?v=Y14b7_T3AEw<br>https://www.youtube.com/watch?v=bK-VhQA512c<br>https://www.youtube.com/watch?v=o2_iCzS9-ZQ<br>https://www.youtube.com/watch?v=SnKgEuUfV4k<br>https://www.youtube.com/watch?v=86FAWCzIe_4 | **30 hours** |
| **Web References** | https://www.geeksforgeeks.org/operating-systems/operating-systems/<br>https://studyhubzone.com/operating-systems-notes/ | **4 hours** |

**Mode of Evaluation**

| CIE | | | | | | | ESE | Total |
|---|---|---|---|---|---|---|---|---|
| **ST1** | **ST2** | **ST3** | **TA1**<br>**10** | **TA2**<br>10 | **TA3**<br>**10** | **Attendance**<br>**10** | | |
| **30** | | | **20** | | | | 50 | **100** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| **Course Code:** CCSE0301 | **Course Name: DATA STRUCTURES AND ALGORITHMS-II** | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|
| **Course Offered in: CSE/CSE-R/CS/IT/CSE(AI)/CSE(AIML)/CSE(DS)/CS/CSE(IOT)/M.tech.Int/ CSE-TWIN/IT-TWIN/CYS** | | **3** | **0** | **0** | **3** |

**Pre-requisite:** C, Python

**Course Objectives:** The objective of the course is to learn the basic concepts of algorithm analysis, along with the implementation of non-linear data structures.

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

| **S.No** | **Course Outcome** | **Bloom's Level** |
|---|---|---|
| CO 1 | Apply tree structures to solve specific problems using various tree operations and algorithms. | K3 |
| CO 2 | Analyse the graph data structure and evaluate the efficiency of its operations for problem solving. | K4 |
| CO 3 | Evaluate dynamic programming solutions for efficient problem-solving across diverse contexts. | K4 |
| CO 4 | Apply efficient backtracking and branch –and –bound techniques across diverse problem-solving scenarios. | K3 |
| CO 5 | Understand principles of advanced data structures, including their implementation and applications. | K2 |

**CO-PO Mapping (Scale: 1: Low, 2: Medium, 3: High)**

| **CO-PO Mapping** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | - | 2 | 1 | 2 | 1 |
| **CO2** | 3 | 3 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | - | 2 | 1 | 2 | 1 |
| **CO3** | 3 | 3 | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | - | 2 | 1 | 2 | 1 |
| **CO4** | 3 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | - | 2 | 1 | 2 | 1 |
| **CO5** | 3 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | - | 2 | 1 | 2 | 1 |

**Course Contents / Syllabus**

| **Unit 1** | **Design and Analysis of Algorithms: Tree** | **9 hours** |
|---|---|---|

**Trees:** Terminology used with Trees, Binary Tree, Memory representation of Tree, Traversal Algorithms: In-order, Pre-order, and post-order. Constructing Binary Tree from given Tree Traversal, Operation of Insertion, Deletion, Searching & Modification of data in Binary Search tree, Binary Heaps, Threaded Binary trees, Traversing Threaded Binary trees, AVL Tree.

**Application of Trees:** Priority Queue, Heap Sort

| **Unit 2** | **Design and Analysis of Algorithms: Graphs** | **9 hours** |
|---|---|---|

**Graphs:** Terminology used with Graph, Data Structure for Graph Representations: Adjacency matrices, Adjacency List.
**Graph Traversal:** Depth First Search and Breadth First Search. Connected Component, Spanning Trees.
**Algorithms on Graphs:** Minimum Cost Spanning
**Trees:** Prim' s and Kruskal's algorithm , Transitive Closure and Shortest Path algorithms: Dijkstra Algorithm, Bellman Ford Algorithm, Floyd Warshall's Algorithm.

| **Unit 3** | **Dynamic Programming** | **9 hours** |
|---|---|---|

**Dynamic Programming:** Dynamic Programming concepts 0/1 Knapsack, Longest Common Sub Sequence, Matrix Chain Multiplication, Resource Allocation Problem.

| **Unit 4** | **Backtracking, Branch and Bound** | **9 hours** |
|---|---|---|

**Backtracking:** Backtracking, Branch, and Bound with Examples Such as Travelling Salesman Problem, Graph Colouring, n-Queen Problem, Hamiltonian Cycles, and Sum of Subsets.

| **Unit 5** | **Advanced- Data Structures** | **9 hours** |
|---|---|---|

Red-Black Trees, B – Trees, B+ Trees, Binomial Heaps, Fibonacci Heaps

**Total Lecture Hours: 45 hours**

**Textbook:**

| **S.No.** | **Book Details** | **Author** |
|---|---|---|
| **1** | "Data Structures and Algorithms in Python: An Indian Adaptation", 1st Edition, 2021 | Michael T. Goodrich, Roberto Tamassia |
| **2** | "Data Structures" Schaum's Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd, 2nd Edition, 2017. | Lipschutz |
| **3** | "Introduction to Algorithms", Printice Hall of India, 4th Edition, 2022 | Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, |

**Reference Books:**

| **S.No.** | **Book Details** | **Author** |
|---|---|---|
| **1** | "Data Structure Using C", Oxford University Press, 2nd Edition, 2014. | Reema Thareja |
| **2** | "Data Structure Using C", Pearson Education India, 2nd Edition,2011. | AK Sharma |
| **3** | "C and Data structure", Wiley Dreamtech Publication, 1st Edition, 2004. | P. S. Deshpandey |

| **Self-Study Content Links:** | | **Duration (in Hrs)** |
|---|---|---|
| **MOOCs** | https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013842032404848640104 70_shared/overview | **18 hours** |

| | | |
|---|---|---|
| | https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013094439914094592258 4_shared/overview | **3 hours** |
| **Video Links** | https://www.youtube.com/watch?v=zWg7U0OEAoE&list=PLBF3763AF2E1C572F<br>https://www.youtube.com/watch?v=LQx9E2--p5c&pp=ygUMYXJyYXlzIG5wdGVs<br>https://www.youtube.com/watch?v=K7VIKlUdo20&pp=ygUPbGluayBsaXN0IG5wdGVs<br>https://www.youtube.com/watch?v=g1USSZVWDsY&list=PLB3CD0BBB95C1BF09&index=2&pp=iAQB<br><br>https://www.youtube.com/watch?v=THMyk2_p530&pp=ygUccXVldWUgZGF0YSBzdHJ1Y3R1cmUgICBucHRlbA%3D%3D<br>https://www.youtube.com/watch?v=_VV9v41FIq0&pp=ygUZZGl2aWRlIGFuZCBjb25xdWVyICBucHRlbA%3D%3D<br><br>https://www.youtube.com/watch?v=ARvQcqJ_-NY&list=PLfFeAJ-vQopt_S5XlayyvDFL_mi2pGJE3 | **40 Hours** |
| **Web References** | https://www.msajce-edu.in/academics/it/LectureNote/CD3291-LN.pdf | - |

**Mode of Evaluation**

| CIE | | | | | | | ESE | Total |
|---|---|---|---|---|---|---|---|---|
| **ST1** | **ST2** | **ST3** | **TA1 10** | **TA2 10** | **TA3 10** | **Attendance 10** | | |
| **30** | | | **20** | | | | **100** | **150** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

### (An Autonomous Institute)

### School of Computer Science and Information Technology

| **Course Code:** CCSAI0301 | **Course Name: Artificial Intelligence** | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|
| **Course Offered in: CSE/IT/CSE-TWIN/IT-TWIN/AI/AIML/CSE-R/M.Tech.Int** | | **2** | **0** | **0** | **2** |

**Pre-requisite: Fundamentals of AI, Basic Python, Problem Solving Approach**

**Course Objectives:** The objective of this course is to equip students with a foundational understanding of Artificial Intelligence. The course emphasizes intelligent agent design, search strategies, knowledge representation, planning, and expert systems, fostering analytical thinking and enabling students to model and solve real-world AI problems effectively.

| **Course Outcome:** After completion of the course, the student will be able to | | **Bloom's Knowledge Level (KL)** |
|---|---|---|
| **CO1** | Apply uninformed and informed search techniques to solve real world problems. | K3 |
| **CO2** | Analyze the performance of adversarial search algorithms in solving competitive problems. | K4 |
| **CO3** | Demonstrate knowledge representation techniques. | K3 |
| **CO4** | Model statistical reasoning to create solutions. | K4 |
| **CO5** | Analyze and evaluate search strategies, knowledge representation models, and expert system approaches for solving complex AI problems | K4 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| **CO-PO Mapping** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 2 | 2 | 3 | - | - | 1 | - | - | - | 3 | 1 | 2 | 2 |
| **CO2** | 3 | 3 | 2 | 2 | 3 | - | - | 1 | - | - | - | 3 | 1 | 2 | 1 |
| **CO3** | 2 | 2 | - | 1 | 2 | - | - | - | - | - | - | 3 | 1 | 3 | 2 |
| **CO4** | 3 | 3 | - | 2 | 3 | - | - | - | - | - | - | 3 | 1 | 3 | 2 |
| **CO5** | 3 | 3 | - | 2 | 2 | 3 | - | 2 | 2 | - | - | 2 | 1 | 2 | 2 |

**Course Contents / Syllabus**

| **Module 1** | **Problem Solving Methodologies** | **8 hours** |
|---|---|---|

Solving Problems by Searching, Uninformed search: BFS, DFS, Iterative deepening, Bi-directional search, Informed search techniques: heuristic, Greedy Best First Search, A* search, AO*

| | | | |
|---|---|---|---|
| search, Constraint satisfaction problems | | | |
| **Module 2** | **Adversarial Search** | | **8 hours** |
| Game Playing: minimax, alpha-beta pruning | | | |
| Solving Problem: Water-Jug problem, Queens Problem, Travelling Salesperson Problem, Missionaries Cannibals problem, tiles problem. | | | |
| **Module 3** | **Knowledge Representation and Reasoning** | | **8 hours** |
| Building a Knowledge Base: Propositional logic, first order logic, Semantic Net, Frame.<br><br>Expert System: Expert System, Architecture of Expert System, | | | |
| **Module 4** | **Statistical Reasoning** | | **6 hours** |
| Probability and Bayes Theorem, Certainty factors and Rule Based systems, Bayesian Networks, Dempster-Shafer Theory, Fuzzy Logic. | | | |
| **Total Lecture Hours** | | | **30 hours** |
| **Textbook:** | | | |
| **S.No** | **Book Title** | **Author** | |
| **1** | *Artificial Intelligence: A Modern Approach* , Pearson Education, 4th Edition, 2020 | Stuart Russell & Peter Norvig | |
| **2** | *Artificial Intelligence* , McGraw-Hill Education, 3rd Edition, 2009 | Elaine Rich, Kevin Knight, Shivashankar B. Nair | |
| **Reference Books:** | | | |
| **S.No** | **Book Title** | **Author** | |
| **1** | *Artificial Intelligence and Machine Learning* , Dreamtech Press, 1st Edition, 2020 | P. S. Deshpande | |
| **2** | *Introduction to Artificial Intelligence* , Springer, 2nd Revised Edition (English Translation), 2017 | Wolfgang Ertel, | |
| **Self-Study Contents Links:** | | **Duration (in Hrs)** | |
| **MOOCs** | https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013842796631269376235 75_shared/overview<br><br>https://infyspringboard.onwingspan.com/web/en/app/toc/lex_8840337130015322000_shared/overview | 7 hours<br><br>6 hours | |
| **Video Link:** | https://www.youtube.com/watch?v=qHhwkV00KJ8&ab_channel=URBS-LabwithRyanUrbanowicz<br>https://www.youtube.com/watch?v=-IO4fPO0rxk&ab_channel=StanfordOnline<br>https://www.youtube.com/watch?v=l-hh51ncgDI<br>https://www.youtube.com/watch?v=adx04dTgJsw&ab_channel=MuhammadUmarFarooq | 20 hours | |
| **Web References** | https://www.cet.edu.in/noticefiles/271_AI%20Lect%20Notes.pdf | | |
| **Mode of Evaluation** | | | |
| **CIE** | | **ESE** | **Total** |

| ST1 | ST2 | ST3 | TA1<br>5 | TA2<br>5 | Attendance<br>10 | | |
|---|---|---|---|---|---|---|---|
| 30 | | | 20 | | | 50 | 100 |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| **Course Code:** CCSE0304 | **Course Name:** Computer Architecture & Parallel Processing | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|
| **Course Offered in: CSE/CSE-R/IT/ CSE-TWIN/IT-TWIN /M.TECH.INT/DS/CYS** | | **3** | **0** | **0** | **3** |

**Pre-requisite:** Basic knowledge of computer systems, Logic gates and their operations.

**Course Objectives:** This course aims to provide a comprehensive understanding of computer organization and architecture, covering processor design, memory systems, and control units. It explores advanced topics such as cache coherence, parallel architectures, and scalable shared memory systems. Students will gain insight into system performance through instruction execution, arithmetic operations, synchronization, and interconnect strategies.

| **Course Outcome:** After completion of the course, the student will be able to | | **Bloom's Knowledge Level (KL)** |
|---|---|---|
| **CO1** | Explain the basic structure and operation of a digital computer system. | K2 |
| **CO2** | Analyze the design of arithmetic & logic unit and understand the fixed point and floating-point arithmetic operations. | K4 |
| **CO3** | Implement control unit techniques and the concept of pipelining. | K3 |
| **CO4** | Analyze parallel architectures and coherence protocols by exploring memory hierarchy, cache coherence mechanisms, and multiprocessor design techniques to ensure correctness and performance in parallel systems. | K4 |
| **CO5** | Analyze scalable shared memory systems by evaluating directory coherence protocols, memory consistency models, synchronization mechanisms, and interconnect strategies to ensure system. | K4 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| **CO-PO Mapping** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 1 | 1 | 1 | - | - | - | - | - | - | - | - | 1 | 1 | 2 | - |
| **CO2** | 2 | 2 | 2 | - | - | - | - | - | - | - | - | 1 | 2 | 2 | - |
| **CO3** | 2 | 2 | 2 | - | - | - | - | - | - | - | - | 1 | 2 | 2 | - |
| **CO4** | 2 | 2 | 2 | - | - | - | - | - | - | - | - | 1 | 2 | 2 | - |
| **CO5** | 2 | 2 | 2 | - | - | - | - | - | - | - | - | 1 | 2 | 2 | - |
| | | | | | | | | | | | | | | | |

**Course Contents / Syllabus**

| Unit 1 | Introduction | 9 hours |
|---|---|---|
| Computer Organization and Architecture, Functional units of digital system and their interconnections, buses, bus architecture, types of buses and bus arbitration and its types. Register, bus and memory transfer. Processor organization, general registers organization, stack organization and addressing modes. | | |
| **Unit 2** | **ALU Unit** | **9 hours** |
| Arithmetic and logic unit: Multiplication: Signed operand multiplication, Booth's algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Arithmetic &logic unit design. IEEE Standard for Floating Point Numbers. | | |
| **Unit 3** | **Control Unit** | **9 hours** |
| Control Unit: Instruction types, formats, instruction cycles and sub cycles (fetch and execute etc.), micro- operations, execution of a complete instruction. Program Control, Reduced Instruction Set Computer, Complex Instruction Set Computer, Pipelining. Hardwire and microprogrammed control, Concept of horizontal and vertical microprogramming, Flynn's classification. | | |
| **Unit 4** | **Introduction to Parallel Architectures** | **9 hours** |
| Introduction to Parallel Architectures, Parallel Programming models and Architectures, Memory Hierarchy-Cache and Virtual memory, Overview of Cache coherence, Coherence Protocols- Snooping, Directory based protocols, VI protocol, MSI, MESI, Dragon protocol and Correctness of coherence protocols- Types of cache misses, update vs invalidate protocol, Snoop based multiprocessor design, Single and multi-level cache with atomic and bus split transaction bus | | |
| **Unit 5** | **Parallel Systems** | **9 hours** |
| Scalable shared memory systems: Directory coherence protocols- Memory based, cache based, correctness, Case study: Origin- Architecture, protocol, correctness; Sequent NUMA Q- Architecture, protocol, correctness, Memory consistency models- Sequential, Relaxed consistency models, Synchronization-LL-SC, point to point, barrier synchronization, Interconnects- Introduction, Topologies, routing, flow control | | |
| | **Total Lecture Hours** | **45 hours** |

**Textbook:**

| SNO | Book Title | Author |
|---|---|---|
| 1 | Computer System Architecture", 3rd Edition, Pearson Publication, 2007. | M. Mano |
| 2 | Computer Architecture and Organization, Tata McGraw Hill, Third Edition, 1998. | John P. Hayes |
| 3 | Computer Organization and Architecture-Designing for Performance, Pearson Education, Seventh edition, 2006. | William Stallings |
| 4 | Parallel Computer Architecture. Morgan- Kaufmann publishers. | D. E. Culler and J. P. Singh with A. Gupta |
| 5 | Computer Architecture: A Quantitative Approach. Morgan- Kaufmann publishers. | J.L. Hennessy and D. A. Patterson |
| 6 | Parallel Computer Organization and Design. Cambridge University Press. | M. Dubois, M. Annavaram, Per Stenstrom, |

**Reference Books:**

| SNO | Book Title | Author |
|---|---|---|
| 1 | Computer Organization, McGraw-Hill, Fifth Edition, Reprint 2012 | Carl Hamacher, ZvonkoVranesic, SafwatZaky |
| 2 | Advanced Microprocessors and Peripherals, TM | Ray A K, Bhurchandi K M |

| | | |
|---|---|---|
| 3 | “Computer Architecture & Parallel Processing” Mcgraw Hill Education | Kai Hwang |

| **Self-Study Content Links:** | | **Duration (in Hrs)** |
|---|---|---|
| **MOOCs** | https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs168<br>https://nptel.ac.in/courses/106104024 | 12 weeks<br>12 weeks |
| **Video Lectures** | https://www.youtube.com/watch?v=L9X7XXfHYdU&list=PLxCzCOWd7aiHMonh3G6QNKq53C6oNXGrX<br>https://www.youtube.com/watch?v=WLgXUPOjKEc<br>https://www.youtube.com/watch?v=BPhWlFIU1rc<br>https://www.youtube.com/watch? v=6R7JDkpG1Wk&list=PLrjkTql3jnm8HbdMwBYIMAd3UdstWChFH<br>https://www.youtube.com/watch?v=nxryfWg5Hm4<br>https://www.youtube.com/watch?v=txAyA_UozmM | 2 hours |
| **Web References** | https://media.geeksforgeeks.org/courses/syllabus/db29b4a6e537cb1a4c4a9119e1ca565f.pdf | - |

**Mode of Evaluation**

| **CIE** | | | | | | **ESE** | **Total** |
|---|---|---|---|---|---|---|---|
| **ST1** | **ST2** | **ST3** | **TA 15** | **TA 25** | **Attendance 10** | | |
| **30** | | | **20** | | | **100** | **150** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

### (An Autonomous Institute)

### School of Computer Science and Information Technology

| LAB Course Code: CCSE0353A | LAB Course Name: Operating Systems Lab | L | T | P | C |
|---|---|---|---|---|---|
| Course Offered in:CSE/CSE-R/IT/ CSE-TWIN/IT-TWIN/M.TECH.INT/CS/AI/AIML/ IOT/DS/CYS | | 0 | 0 | 4 | 2 |

**Pre-requisite:** Basic knowledge of computer fundamentals, C programming, Data structures and Computer organization.

**Course Objectives:** The course aims to provide hands-on experience with Linux and shell programming, while the lab focuses on implementing and analyzing key OS algorithms and simulating modern operating systems.

| **Course Outcome:** After completion of the course, the student will be able to | | **Bloom's Knowledge Level (KL)** |
|---|---|---|
| **CO1** | Execute basic Linux commands and shell scripts to automate file management and system administration tasks. | K3 |
| **CO2** | Implement and compare various CPU scheduling algorithms, process synchronization solutions using semaphores and deadlock handling algorithms. | K4 |
| **CO3** | Simulate memory allocation techniques and page replacement algorithms, disk management strategies and explore modern OS features including virtualization and distributed computing. | K5 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| CO-PO | PO 1 | PO2 | PO3 | PO 4 | PO5 | PO6 | PO 7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 | PSO4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 1 | 1 | 3 | - | - | 2 | 2 | - | 3 | 2 | 1 | 2 | 2 |
| **CO2** | 3 | 3 | 3 | 2 | 2 | - | - | 2 | 2 | - | 2 | 2 | 2 | 3 | 3 |
| **CO3** | 3 | 2 | 3 | 2 | 3 | - | - | 2 | 2 | - | 3 | 2 | 3 | 3 | 3 |

**List Of Practical's (Indicative & Not Limited To)**

| S.No | Experiment | CO Mapping |
|---|---|---|
| 1 | Installation of Linux Operating system. | CO1 |
| 2 | **Implementation of Linux Commands**<br>i. Display system information using uname, hostname, and date etc.<br>ii. File operations using cat, touch, cp, mv, rm, and chmod ,umask etc.<br>iii. Create, view, and navigate directories using mkdir, rmdir, cd, pwd, ls etc.<br>iv. Disk Commands df,du,mount,unmount,mkfs,fsck etc. | CO1 |

| | | |
|---|---|---|
| 3 | **Implementation of Linux Commands (continued)**<br>v. Use redirection and piping in commands.<br>vi. File compression and archiving using tar, gzip, zip, unzip etc.<br>vii. Process commands ps,kill, killall,nice, pgrep, top,htop etc.<br>viii. Network commands ifconfig, ping, netstat, host,ip route etc.<br>ix. Administrator Commands Adduser,Passwd, deluser, usermod, groupadd etc | CO1 |
| 4 | **Shell Scripting Programming**<br>i. Write a shell script to ask your name, program name and enrollment number and print it on the screen.<br>ii. Write a shell script to find the sum, the average and the product of the four integers entered.<br>iii. write shell script to find average of numbers given at command line<br>iv. Write a shell program to exchange the values of two variables.<br>v. Write a shell program to Print Numbers 1 to 10 using while & do while loop.<br>vi. Write a shell program to Print Numbers 1 to 10 using for loop. | CO1 |
| 5 | **Shell Scripting Programming (continued)**<br>vii. Write a shell script to display the digits which are in odd position in a given 5-digit number**.**<br>viii. Write a shell program to search for a given number from the list of numbers provided using binary search method.<br>ix. Write a shell program to concatenate two strings and find the length of the resultant string.<br>x. Write a shell script to find the smallest of three numbers.<br>xi. Write a shell program to count number of words, characters, white spaces and special symbols in a given text. | CO1 |
| 6 | Implement FCFS CPU Scheduling algorithm. | CO2 |
| 7 | Implement the SJF CPU Scheduling algorithm (For both Pre-emptive and non-pre-emptive). | CO2 |
| 8 | Implement PRIORITY CPU Scheduling Algorithm (For both Pre-emptive and non-pre-emptive). | CO2 |
| 9 | Implement Round-Robin CPU Scheduling Algorithm. | CO2 |
| 10 | Implement Multi-Level Queue CPU Scheduling algorithm. | CO2 |
| 11 | Implement Multilevel Feedback Queue CPU Scheduling Algorithm. | CO2 |
| 12 | Execute the RACE Condition of Process Synchronization. | CO3 |
| 13 | Implement the Producer–consumer problem using semaphores. | CO3 |
| 14 | Design a code and implement the Dinning Philosopher problem. | CO3 |
| 15 | Implement Banker's algorithm of Deadlock Avoidance. | CO3 |
| 16 | Execute an algorithm for Deadlock Detection. | CO3 |
| 17 | Implement Contiguous memory variable size partition scheme. | CO3 |
| 18 | Simulate the First-Fit contiguous memory allocation technique. | CO3 |
| 19 | Simulate the Best-Fit contiguous memory allocation technique. | CO3 |
| 20 | Simulate the Worst-Fit contiguous memory allocation technique. | CO3 |
| 21 | Implement the Non-contiguous Memory Allocation by using Paging. | CO3 |

| | | |
|---|---|---|
| 22 | Implement Contiguous memory variable size partition scheme. | CO3 |
| 23 | Write a Program to simulate the FIFO page replacement algorithm. | CO3 |
| 24 | Write a Program to simulate the LRU page replacement Algorithm. | CO3 |
| 25 | Write a Program to simulate the Optimal page replacement Algorithm. | CO3 |
| 26 | Write a program to simulate FCFS Disk Scheduling Algorithm. | CO3 |
| 27 | Write a Program to simulate the SSTF Disk Scheduling Algorithm | CO3 |
| 28 | Write a program to simulate SCAN Disk Scheduling Algorithm. | CO3 |
| 29 | Write a Program to simulate the C SCAN Disk Scheduling Algorithm. | CO3 |
| 30 | Write a Program to simulate the LOOK Disk Scheduling Algorithm. | CO3 |
| | | |
| | | **Total Hours: 60 hrs** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| **LAB Course Code:** CCSE0351 | **LAB Course Name:** Data structure and algorithms lab-II | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|
| **Course Offered in: CSE/CS/CSR-R/M.TECH(INT)/ CSE-TWIN/IT-TWIN /IT//CSE(AI)/CSE(AIML)/CSE(DS)/CSE(CS)/CYS** | | **0** | **0** | **4** | **2** |

**Pre-requisite:** C, Python

**Course Objectives:**. Learn to implement linear data structures

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

| **SNO** | **Course Outcome** | **Bloom's Level** |
|---|---|---|
| 1. | Implementation of tree data structures for basic operations like insertion, deletion, searching and traversal | K3 |
| 2 | Implementation of algorithms based on graph data structures for solving real world problems. | K3 |
| 3 | Implementing Dynamic Programming, Backtracking, Branch and Bound algorithms to solve complex data efficiently and effectively. | K3 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| **CO-PO Mapping** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** | **PSO 4** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | - | 2 | 1 | 2 | 1 | |
| **CO2** | 3 | 3 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | - | 2 | 1 | 2 | 1 | |
| **CO3** | 3 | 3 | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | - | 2 | 1 | 2 | 1 | |

| **List Of Practical's (Indicative & Not Limited To)** | **Mapped CO** |
|---|---|
| 1. Write a program to implement an in-order traversal of a binary tree and print the nodes. | CO1 |
| 2. Write a program to implement a pre-order traversal of a binary tree and print the nodes. | CO1 |
| **3.** Write a program to implement a post-order traversal of a binary tree and print the nodes. | CO1 |
| 4. Write a program to count the number of nodes in a binary tree. | CO1 |
| 5. Write a program to find the height of the tree. | CO1 |
| 6. Write a program to check if the binary tree is balanced or not. | CO1 |
| 7. Write a program to search a number in a Binary Search Tree (BST). | CO1 |

| | |
|---|---|
| 8. Write a program to insert a node in a Binary Search Tree (BST). | CO1 |
| 9. Write a program to delete a node from a Binary Search Tree (BST). | CO1 |
| 10. Write a program to implement a max-heap and perform heap sort on an array of integers. | CO1 |
| 11. Write a program to implement a priority queue using a max heap. | CO1 |
| 12. Write a program to create a graph using an adjacency matrix. | CO2 |
| 13. Write a program to create a graph using an adjacency list. | CO2 |
| **14.** Write a program to perform Depth-First Search (DFS) on a graph. | CO2 |
| 15. Write a program to perform Breadth-First Search (BFS) on a graph. | CO2 |
| 16. Write a program to check if there is a path between two nodes in a graph using DFS. | CO2 |
| 17. Write a program to find all the vertices reachable from a given vertex in a graph. | CO2 |
| 18. Write a program to detect a cycle in an undirected graph using DFS. | CO2 |
| 19. Write a program to detect a cycle in a directed graph using DFS. | CO2 |
| 20. Write a program to find the degree of each vertex in an undirected graph. | CO2 |
| 21. Write a program to count the number of connected components in an undirected graph. | CO2 |
| 22. Write a program to implement Dijkstra's Algorithm. | CO3 |
| 23. Write a program to implement Prim's Algorithm. | CO3 |
| 24. Write a program to implement Kruskal's Algorithm. | CO3 |
| 25. Write a program to implement Floyd–Warshall's all-pairs shortest path algorithm. | CO3 |
| 26. Write a program to implement Bellman–Ford Algorithm. | CO3 |
| 27. Write a program to implement Longest Common Subsequence (LCS) using Dynamic Programming.. | CO3 |
| 28. Write a program to implement 0/1 Knapsack using Dynamic Programming. | CO3 |
| 29. Write a program for Matrix Chain Multiplication using Dynamic Programming. | CO3 |
| 30. Write a program to implement Sum of Subsets problem using backtracking. | CO3 |
| **Total Hours:** | **60 hrs** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| LAB Course Code: CCSAI0351 | LAB Course Name: Artificial Intelligence Lab | L | T | P | C |
|---|---|---|---|---|---|
| **Course Offered in: CSE/CSE-R/IT/CSE-TWIN/IT-TWIN/AI/AIML/M.TECH.INT** | | **0** | **0** | **2** | **1** |

**Pre-requisite:** Basic knowledge of Python programming, statistics, linear algebra, and data analysis using libraries like NumPy and Pandas

**Course Objectives:**

This course aims to equip students with practical skills in fundamental AI algorithms, including search techniques, adversarial games, knowledge representation, and reasoning. It also develops proficiency in statistical methods such as Bayesian inference and fuzzy logic, enabling effective problem-solving and decision-making under uncertainty.

| **Course Outcome:** After completion of the course, the student will be able to | | **Bloom's Knowledge Level (KL)** |
|---|---|---|
| **CO1** | Implement BFS, DFS, A* search, and backtracking techniques to solve graph and constraint satisfaction problems | K3 |
| **CO2** | Develop adversarial search algorithms like Minimax and Alpha-Beta Pruning for games and heuristic problem solving. | K3 |
| **CO3** | Apply knowledge representation and uncertainty handling methods to support decision-making in AI systems | K3 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| CO-PO Mapping | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 | PSO4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 2 | 2 | 3 | - | - | 1 | - | - | - | 3 | 1 | 2 | 2 |
| **CO2** | 3 | 3 | 2 | 2 | 3 | - | - | 1 | - | - | - | 3 | 1 | 2 | 1 |
| **CO3** | 2 | 2 | - | 1 | 2 | - | - | - | - | - | - | 3 | 1 | 3 | 2 |

**List Of Practical's (Indicative & Not Limited To)**

| SNO | List of Prcticals | COS |
|---|---|---|
| | **Problem Solving Methodologies** | |
| 1. | Implement Breadth First Search (BFS) for a given graph using Python | **CO1** |
| 2. | Implement Depth First Search (DFS) for a graph represented using an adjacency list. | **CO1** |
| 3. | Write a program to implement A Search algorithm using a given heuristic function. | **CO1** |

| | | |
|---|---|---|
| 4. | Solve the Map Coloring Problem using Backtracking for 4 colors (Constraint Satisfaction Problem) | **CO1** |
| | **Adversarial Search** | |
| 5. | Implement the Minimax algorithm for a two-player game like Tic-Tac-Toe. | **CO2** |
| 6. | Implement Alpha-Beta Pruning to optimize the Minimax algorithm in a game tree | **CO2** |
| 7. | Develop a program to solve the 8-puzzle (tiles problem) using the A search algorithm | **CO2** |
| 8. | Implement Iterative Deepening Search to solve the Water Jug Problem. | **CO2** |
| | **Knowledge Representation and Reasoning** | |
| 9. | Implement a Propositional Logic Evaluator that takes logical expressions and returns their truth values. | **CO3** |
| 10. | Develop a First Order Logic (FOL) knowledge base and implement unification and inference using Python | **CO3** |
| 11. | Create a Semantic Network representation of a small domain (e.g., animal hierarchy) and allow querying relationships.Simulate a simple Rule-Based Expert System (e.g., medical diagnosis or career advisor) with forward chaining. | **CO3** |
| | **Statistical Reasoning** | |
| 12. | Implement a program to calculate conditional probabilities using Bayes' Theorem. | **CO3** |
| 13. | Create a basic Bayesian Network for a small problem (e.g., disease diagnosis) and perform probabilistic inference | **CO3** |
| 14. | Implement a Fuzzy Logic controller for a temperature control system with fuzzy sets and rules.Implement Dempster-Shafer Theory to combine evidence from multiple sources and calculate belief and plausibility | **CO3** |
| | | **Total Hours: 30 hrs.** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| Course Code:CCSE0351 | Course Name: Object Oriented Techniques using Java | L | T | P | C |
|---|---|---|---|---|---|
| Course Offered in: CSE/CS/IT/CSE(AI)/CSE(AIML)/CSE(IOT)/CSE(AI)/CSE(DS)/CSE-R/M.Tech int/ CSE-TWIN/IT-TWIN/CYS | | 0 | 0 | 6 | 3 |

**Pre-requisite:**Basic Programming Knowledge,Knowledge of any programming language (e.g., C, C++, Python)

**Course Objectives:**

The objective of this course is to understand the object-oriented methodology, and its techniques to design stand alone and GUI applications using hands-on engaging activities.

| **Course Outcome:** After completion of the course, the student will be able to | | Bloom's Knowledge Level (KL) |
|---|---|---|
| CO1 | Understand the concepts of object-oriented programming and relationships among them needed in modeling. | K2 |
| CO2 | Demonstrate the Java programs using OOP principles and also implement the concepts of lambda expressions | K3 |
| CO3 | Analyze packages with different protection level resolving namespace collision and implement the error handling concepts for uninterrupted execution of Java program. | K4 |
| CO4 | Analyze Concurrency control, I/O Streams and Java Socket Programming Concepts. | K4 |
| CO5 | Design and develop the GUI based application, Generics and Collections in Java programming language to solve the real-world problem | K6 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| CO-PO Mapping | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 | PSO4 | PSO4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 3 | 3 | 2 | 2 | 1 | - | 1 | - | 2 | 3 | 3 | 2 | 2 | 2 |
| **CO2** | 3 | 3 | 3 | 3 | 2 | 2 | 1 | - | 1 | 1 | 2 | 3 | 2 | 2 | 2 | 2 |
| **CO3** | 3 | 3 | 3 | 3 | 3 | 2 | 2 | - | 2 | 1 | 2 | 3 | 3 | 1 | 2 | 2 |
| **CO4** | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 1 | 2 | 1 | 2 | 3 | 2 | 2 | 1 | 1 |
| **CO5** | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 1 | 2 | 1 | 2 | 3 | 3 | 2 | 2 | 2 |

**Course Contents / Syllabus**

| **Unit 1** | **Basics of Java Programming** | **16 hours** |
|---|---|---|

**Object Oriented Programming:** Introduction and Pillars of OOP with real life example, jvm architecture and its components
**Modelling Concepts:** Introduction, Class Diagram and Object Diagram, UML concepts: Association, Composition, aggregation, realization, and Generalization.
**Control Statements:** Decision Making, Looping and Branching, Argument Passing Mechanism: Command Line Argument, Console Input.

| | | |
|---|---|---|
| **Class and Object:** Object Reference, Constructor, Abstraction: Abstract Class, Interface and its uses, Defining Methods, Use of "this" and "super" keyword, Garbage Collection and finalize () Method etc. | | |
| **Unit 2** | **OOPs features, arrays and lambda expressions** | **16 hours** |
| **Inheritance:** Introduction and Types of Inheritance in Java, Access Modifiers, Constructors and super constructor in Inheritance.<br><br>**Polymorphism:** Introduction and Types, Overloading and Overriding.<br><br>**Lambda expression:** Introduction and Working with Lambda Variables.<br><br>**Arrays:** Introduction and its Types. Jagged Array with example | | |
| **Unit 3** | **Packages, Exception Handling and String Handling** | **16 hours** |
| **Packages:** Introduction and Types, Access Protection in Packages, Import and Execution of Packages.<br>E**xception Handling, Assertions and Localizations:** Introduction and Types, Exceptions vs. Errors, Handling of Exception. Finally, Throws and Throw keyword, Multiple Catch Block, Nested Try and Finally Block, Customized Exceptions, Tokenizer. Assertions and Localizations Concepts and its working.<br>**String Handling:** Introduction and Types, Operations, Immutable String, Method of String class, String Buffer and String Builder class. | | |
| **Unit 4** | **Concurrency in Java and I/O Stream** | **16 hours** |
| **Threads:** Introduction and Types, Creating Threads, Thread Life-Cycle, Thread Priorities, Daemon Thread,Runnable Class, Synchronizing Threads etc.<br>**I/O Stream:** Introduction and Types, Common I/O Stream Operations, Interaction with I/O Streams Classes. character and byte oriented stream classes with example<br>**Java Socket Programming**: Introduction and types(TCP, UDP), java socket program with server-side and client-side by using connection. | | |
| **Unit 5** | **GUI Programming, Generics and Collections** | **16 hours** |
| **GUI Programming:** Introduction and Types, Swing, AWT, Components and Containers, Layout Managersand User-Defined Layout and Event Handling.<br>**Generics:** Introduction to Generic Classes, types of generic defined in brief, bounded type parameter(Upper and Lower bound), Initializing a Generic Object, Classes, Methods and Interfaces Use enumerated type.<br>**Collections:** Introduction, main interfaces of collections(Collection, List Set, Map, Queue), classes of collections(ArrayList,Linked list, HashSet, HashMap and TreeSet) and methods(List, Set Map) Collection using Iterators | | |
| | **Total Lecture Hours** | **80 hours** |
| **Textbook:** | | |
| **S.NO** | **Book Title** | **Author** |
| 1. | " Java: A Beginner's Guide", McGraw-Hill Education 2nd edition | Herbert Schildt |
| 2. | "Programming with Java A Primer", TMH, 4th edition. | E Balagurusamy, |
| **Reference Books:** | | |
| **S.NO** | **Book Title** | **Author** |

| | | |
|---|---|---|
| 1. | "Core Java Volume I – Fundamentals", Prentice Hall | Cay S. Horstmann |
| 2. | " Effective Java", Addison Wesley | Joshua Bloch |
| 3. | Java - The Complete Reference", McGraw Hill Education 12th edition | Herbert Schildt |
| **Self-Study Content Links:** | | **Duration (in Hrs)** |
| **MOOCs** | https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012540972274925568106 3_shared/overview | 46h 13m |
| **Video Lectures** | https://www.youtube.com/watch?v=iw1Xf_33YM0&list=PLQEaRBV9gAFujcBWJhBT2XXsuMllfETBy | 30 h |
| **Web References** | 1. https://www.geeksforgeeks.org/dsa/introduction-of-object-oriented-programming/<br>2. https://en.wikipedia.org/wiki/Object-oriented_programming | 20 m<br>58 m |

**Mode of Evaluation**

| CIE | | | | | | | ESE | Total |
|---|---|---|---|---|---|---|---|---|
| **ST1** | **ST2** | **ST3** | **TA1**<br>**10** | **TA2**<br>10 | **TA3**<br>**10** | **Attendance**<br>**10** | | |
| **30** | | | **20** | | | | **100** | **150** |

| **S.N.** | **List Of Practical's (Indicative & Not Limited To)** | **CO** |
|---|---|---|
| 1 | Understanding Text Editors to Write Programs Compile and run first java file Byte Code and class file | CO1 |
| 2 | Sketch a class and object diagram describing the sales order system of restaurant | CO1 |
| 3 | Sketch a class diagram describing the circle and rectangle class | CO1 |
| 4 | Sketch a class diagram for a college platform including, classroom, playground, chair, table, smart board, teaching staff etc. | CO1 |
| 5 | Sketch a class diagram containing class called Employee, which models an employee with an ID, name and salary. Add method raiseSalary(percent) that increases the salary by the given percentage. | CO1 |
| 6 | Program to display default value of all Primitive data types | CO1 |
| 7 | Implement the code using main() method to calculate and print the Total and Average marks scored by a student from the input given through the command line arguments.<br>Assume that four command line arguments name, marks1, marks2, marks3 will be passed to the main() method in the below class with name TotalAndAvgMarks. | CO1 |

| | | |
|---|---|---|
| 8 | Write code which uses if-then-else statement to check if a given account balance is greater or lesser than the minimum balance. Write a class BalanceCheck with public method checkBalance that takes one parameter balance of type double. Use if-then-else statement and print Balance is low if balance is less than 1000. Otherwise, print Sufficient balance. | CO1 |
| 9 | A class NumberPalindrome with a public method isNumberPalindrome that takes one parameter number of type int. Write a code to check whether the given number is palindrome or not.<br>For example Cmd Args : 333<br>333 is a palindrome | CO1 |
| 10 | Write a class FibonacciSeries with a main method. The method receives one command line argument. Write a program to display fibonacci series i.e. 0 1 1 2 3 5 8 13 21 | CO1 |
| 11 | Write a Java Program to find the Factorial of a given number. | CO1 |
| 12 | Java Program to create a class, methods and invoke them inside main method. | CO1 |
| 13 | Write a Java program to illustrate the abstract class concept. Create an abstract class Shape, which contains an empty method numberOfSides(). Define three classes named Trapezoid, Triangle and Hexagon extends the class Shape, such that each one of the classes contains only the method numberOfSides(), that contains the number of sides in the given geometrical figure. Write a class AbstractExample with the main() method, declare an object to the class Shape, create instances of each class and call numberOfSides() methods of each class. | CO1 |
| 14 | Java program to illustrate the static field in the class. | CO1 |
| 15 | Java Program to illustrate static class. | CO1 |
| 16 | Write a java program to access the class members using super keyword | CO1 |
| 17 | Java program to access the class members using this keyword | CO1 |
| 18 | Implement an interface named MountainParts that has a constant named TERRAIN that will store the String value “off_road”. The interface will define two methods that accept a String argument name newValue and two that will return the current value of an instance field. The methods are to be named: getSuspension, setSuspension, getType , setType. | CO1 |
| 19 | Java program to demonstrate nested interface inside a interface. | CO1 |
| 20 | Java program to demonstrate nested interface inside a class. | CO1 |
| 21 | Java program to explicit implementation of garbage collection by using finalize() method | CO1 |
| 22 | Java program to implement Single Inheritance | CO1 |
| 23 | Java program to implement multi- level Inheritance | CO1 |
| 24 | Java program to implement constructor and constructor overloading. | CO1 |
| 25 | Java program implement method overloading. | CO1 |
| 26 | Java program to implement method overriding. | CO1 |
| 27 | .Java program to implement lambda expression without parameter. | CO1 |
| 28 | Java program to implement lambda expression with single parameter. | CO1 |
| 29 | Java program to implement lambda expression with multi parameter. | CO1 |
| 30 | Java program to implement lambda expression that iterate list of objects | CO2 |
| 31 | .Java program to define lambda expressions as method parameters | CO2 |
| 32 | Write a class CountOfTwoNumbers with a **public** method compareCountOf that takes three parameters one is arr of type int[] and other two are arg1 and arg2 are of type int and returns true if count of arg1 is greater than arg2 in arr. The return type of compareCountOf should be boolean. | CO2 |

| | | |
|---|---|---|
| | Assummptions:<br>• arr is never null<br>arg1 and arg2 may be same | |
| 33 | Java program to show the multiplication of two matrices using arrays. | CO2 |
| 34 | Java Program to search an element using Linear Search | CO2 |
| 35 | .Java program to search an element using Binary Search | CO2 |
| 36 | Java Program to sort element using Insertion Sort | CO2 |
| 37 | Java Program to sort element using Selection Sort– Largest element Method | CO2 |
| 38 | Java program to Sort elements using Bubble Sort | CO2 |
| 39 | Java program to create user defined package. | CO3 |
| 40 | Java Program to create a sub- classing of package | CO3 |
| 41 | Implement the following:<br>1. Import package.*;<br>2. import package.classname;<br>Using fully qualified name. | CO3 |
| 42 | Implement and demonstrate package names collision in java | CO3 |
| 43 | Java program to handle and Arithmetic Exception Divided by zero | CO3 |
| 44 | .Java Program to implement User Defined Exception in Java | CO3 |
| 45 | Java program to illustrate finally block | CO3 |
| 46 | Java program to illustrate Multiple catch blocks | CO3 |
| 47 | Java program for creation of illustrating throw in exception handling. | CO3 |
| 48 | Implement the concept of Assertion in Java Programming Language | CO3 |
| 49 | Implement the concept of Localization in Java Programming Language. | CO3 |
| 50 | Java program to print the output by appending all the capital letters in the input string. | CO3 |
| 51 | Java program that prints the duplicate characters from the string with its count. | CO3 |
| 52 | Java program to check if two strings are anagrams of each other | CO3 |
| 53 | Java Program to count the total number of characters in a string | CO3 |
| 54 | .Java Program to count the total number of punctuation characters exists in a String | CO3 |
| 55 | Java Program to count the total number of vowels and consonants in a string | CO3 |
| 56 | Java Program to show .equals method and == in java | CO3 |
| 57 | Given a string, return a new string made of n copies of the first 2 chars of the original string where n is the length of the string. The string may be any length. If there are fewer than 2 chars, use whatever is there. If input is "Wipped" then output should be "WiWiWiWiWi". | CO3 |
| 58 | Given two strings, a and b, create a bigger string made of the first char of a, the first char of b, the second char of a, the second char of b, and so on. Any leftover chars go at the end of the result. If the inputs are "Hello" and "World", then the output is "HWeolrllod". | CO3 |
| 59 | Java program to show the usage of string builder. | CO3 |
| 60 | Java program to show the usage of string buffer. | CO3 |
| 61 | Creating and Running a Thread | CO3 |
| 62 | Implementing Runnable Interface | CO3 |

| | | |
|---|---|---|
| 63 | Synchronizing Threads with lock | CO3 |
| 64 | Synchronizing Threads without lock | CO3 |
| 65 | Java program to implement even and odd threads by using Thread class . | CO4 |
| 66 | Java program to implement even and odd threads by using Runnable interface. | CO4 |
| 67 | Java program to synchronize the threads by using Synchronize statements and Synchronize block. | CO4 |
| 68 | .Write a program where the client sends a message to the server, and the server prints it by using TCP | CO4 |
| 69 | Implement a server that can handle multiple clients simultaneously using UDP | CO4 |
| 70 | Write a client-server application where the client uploads a file and the server saves it by using TCP/UDP. | CO4 |
| 71 | Java program to implement that read a character stream from input file and print it into output file. | CO4 |
| 72 | .Java program to implement that merge the content of two files (file1.txt, file2.txt) into file3.txt. | CO4 |
| 73 | Write a Java program that reads the contents of one file and copies them toanother file. | CO4 |
| 74 | Write a Java program that reads a text file and counts the number of words in it. | CO4 |
| 75 | Write a Java program that reads a text file and counts the frequency of each word in it. | CO4 |
| 76 | Write a Java program that reads a text file and adds line numbers to each line. The program should create a new file with the line numbers added to the beginning of each line. | CO4 |
| 77 | Write a Java program that reads two binary files and compares them byte by byte to determine if they are identical. Display a message indicating whether the files are the same or different. | CO4 |
| 78 | Program to create a frame with three button in AWT and swing | CO4 |
| 79 | Program to display message with radio buttons in swing | CO4 |
| 80 | Program to display "All The Best" in 5 different colors on screen. (Using AWT/Swing) | CO5 |
| 81 | Program to implement handling in a button “OK” | CO5 |
| 82 | Java Program to implement BorderLayout | CO5 |
| 83 | .Java Program to implement GridLayout | CO5 |
| 84 | Java Program to implement BoxLayout | CO5 |
| 85 | Java Program to implement CardLayout | CO5 |
| 86 | Java program to implement Generic class | CO5 |
| 87 | Java program to illustrate Generic methods | CO5 |
| 88 | Java program to implement wildcard in generics | CO5 |
| 89 | Java program to implement of methods of HashSet | CO5 |
| 90 | Java Program to implement methods available in HashMap class | CO5 |
| 91 | Program to add, retrieve, and remove element from ArrayList | CO5 |
| 92 | Create a method which can accept a collection of country names and add it to ArrayList with generic defined as String and return the List. | CO5 |
| 93 | Create a method which can create a HashSet containing values 1-10.The Set should be declared with the generic type Integer.The method should return the Set. | CO5 |
| 94 | ava program to implement autoboxing | CO5 |
| 95 | Java program to implement unboxing | CO5 |
| 96 | Develop a java class with a method *storeEvenNumbers(int N)* using ArrayList to store even numbers from 2 to N, where N is a integer which is | CO5 |

| | | |
|---|---|---|
| | passed as a parameter to the method *storeEvenNumbers().* The method<br>should return the ArrayList (A1) created. | |
| 97 | Create a method that accepts the names of five countries and loads them to an array list and returns the list. | CO5 |
| 98 | Create a method which can accept a collection of country names and add it to ArrayList with generic defined as String and return the List. | CO5 |

| | | | | | |
|---|---|---|---|---|---|
| **NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY**<br>**GREATER NOIDA-201306**<br>**(An Autonomous Institute)**<br>**School of Computer Science and Information Technology** | | | | | |
| **Course Code:** CASCC0401 | **Course Name:** Employability Skill Development - II | **L** | **T** | **P** | **C** |
| **Course Offered in:** IV Semester<br>**Note:** Section C - offered in B.Tech. III/IV SEM | | **2** | **0** | **0** | **2** |

**Pre-requisite: Students should have successfully completed communication skills courses in the first year.**

**Course Objectives:**

**Section A**: Develop proficiency in C++ programming, data management, and project development by applying concepts, teamwork, and problem-solving to build practical applications.

**Section B**: Enhance quantitative aptitude skills through business mathematics concepts, enabling accurate problem-solving in financial calculations, ratios, and real-world numerical applications.

**Section C:** To develop the ability to communicate effectively and empathetically in workplace situations while strengthening active listening, adaptability, and cross-cultural awareness for navigating complex professional environments.

| **Course Outcome** | | Bloom's Knowledge Level (KL) |
|---|---|---|
| **CO1** | Apply C++ programming concepts, file handling, and STL techniques to develop efficient programs for data storage and manipulation. | **K3** |
| **CO2** | Design and implement a C++ -based project by applying planning, teamwork, and problem-solving skills. | **K6** |
| **CO3** | Solve real-life business math problems involving percentages, profit and loss, discounts, interest, averages, mixtures, and ratios using appropriate mathematical methods. | **K3** |
| **CO4** | Apply spoken English skills to participate effectively in standard workplace routines, including daily stand-ups, reporting, and collaborative problem-solving. | **K3** |
| **CO5** | Evaluate and adapt communication styles to suit dynamic workplace communication situations and deliver context-sensitive, and logically structured verbal responses under time constraints. | **K6** |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| **C I O-PO Mapping** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | - | 2 | 3 | 1 | - | 3 | 2 | 2 | 2 | - | - | - | - |
| **CO2** | 3 | 3 | 3 | 3 | 3 | 2 | 1 | 3 | 2 | 2 | 2 | - | - | - | - |
| **CO3** | 3 | 3 | 2 | 1 | - | - | - | 1 | 2 | 2 | 2 | - | - | - | - |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO4** | - | - | - | - | 1 | 2 | - | 3 | 2 | 2 | - | - | - | - | - |
| **CO5** | - | 2 | 1 | - | 1 | 2 | - | 3 | 2 | 2 | 1 | - | - | - | - |

**Course Contents / Syllabus**

**Section-A**

| **Module 1:** | **Foundations of C++ Programming and Data Management Techniques** | **8 hours** |
|---|---|---|

Core Fundamentals of C++, Functions & Program Structure, Arrays & Strings, File **Handling:** File streams (ifstream, ofstream, fstream), Reading/writing text files, saving user data (scores, records, logs) **STL Basics:** vector, array, map, unordered_map set Algorithms (sort, find)

| **Module 2** | **Project/Game Development** | **8 hours** |
|---|---|---|

Project Planning & Development (Teams, roles, idea pitching and implement in C++), Mini Project, Project Demonstration and Review

**Section-B**

| **Module 3** | **Business Math I** | **16 hours** |
|---|---|---|

Percentage, Profit and Loss, Discount, Simple Interest and Compound Interest, Average, Ratio & Proportion, Partnership, Mixture & Allegation, Clock , Calendar.

**Section-C**

| **Module-4** | **Workplace Interaction Essentials** | **12 hours** |
|---|---|---|

- The Professional Shift: Navigating the transition from academic to professional tone. Using appropriate register when addressing peers vs. supervisors.
- Active Listening & The Feedback Loop: Moving beyond passive hearing. Practicing the SBI (Situation-Behaviour-Impact) model for peer feedback and clarifying multi-step technical instructions.
- Structured Micro-Speaking: The art of the concise update. Delivering 1-2 minute "Stand-up" reports (What I did, What I will do, Blockers) and explaining basic technical processes to laymen.
- Guided Interpersonal Simulations: Role-playing everyday scenarios: politely interrupting a meeting, seeking assistance on a delayed task, and confirming understanding of complex briefs using paraphrasing techniques.
- Digital & Hybrid Nuances: Verbal signalling in virtual meetings, managing "overtalk," and translating concise written chat messages into spoken professional updates.

| **Module-5** | **Workplace Communication in Action** | **12 hours** |
|---|---|---|

- Dynamic Problem-Based Communication: Simulating the "War Room." Explaining sudden technical failures, proposing immediate solutions, and negotiating revised deadlines with stakeholders.
- Listening Under Pressure & Global Adaptability: Interpreting fast, accented, or incomplete instructions (incorporating CEFR-level listening benchmarks). Decoding tone, urgency, and indirect cultural communication styles.
- Collaborative Conflict Resolution: Managing disagreements in team settings using "I-statements" and the "Yes, and…" technique to build consensus without escalating tension.
- The Persuasive Professional: Delivering evidence-based arguments using the PREP (Point, Reason, Example, Point) framework to justify a technical choice or resource allocation.
- Real-Time "Crisis" Simulation (Capstone): A comprehensive role-play where student groups are given a sudden project change or client issue. They must conduct an emergency meeting, delegate tasks verbally, and present a united response to the "client."

**Total: 56 Hours**

**Reference Books:**

| S. No. | Book Title | Author |
|---|---|---|
| 1. | A Project-Based Introduction to Programming | Access Point Publishing |
| 2. | Programming: Principles and Practice Using C++ | Bjarne Stroustrup |
| 3. | Effective Modern C++ | Scott Meyers |
| 4. | Quantitative Aptitude | R S Aggarwal |
| 5. | Verbal & Non-Verbal Reasoning | R S Aggarwal |
| 6. | *Crucial Conversations Tools* | Kerry Patterson et al. |
| 7. | *HBR Guide to Persuasive Presentations* | Nancy Duarte |
| **Self Study Content Link:** | | Duration(in hours) |
| **MOOCs** | nptel.ac.in/courses/109104107 | 8 weeks |
| Video Lectures | Employability skills - YouTube | 33 m |
| Web Reference | Employability Skills Self-Management Guide \| PDF \| Goal \| Self Esteem | - |

**Mode of Evaluation**

| CIE | | | | ESE | Total |
|---|---|---|---|---|---|
| **Section-A (ST)** | **Section-B (ST)** | **Section-C (ST)** | **Attendance 10** | | |
| **30** | **30** | **30** | **10** | | **100** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| Course Code: CCSE0402 | Course Name: Software Engineering and Design | L | T | P | C |
|---|---|---|---|---|---|
| **Course Offered in: CSE /CSE-R/IT/CYS/ CSE-TWIN/IT-TWIN/M.TECH.INT** | | 3 | **0** | **0** | 3 |

**Pre-requisite:** Fundamental knowledge of programming concepts, data structures, object-oriented concepts

**Course Objectives:** To help students understand all phases of the Software Development Life Cycle (SDLC) both theoretically and practically, enabling them to systematically apply principles of analysis, design, development, testing, and maintenance to build cost-effective software solutions and become competent software engineering professionals.

| **Course Outcome:** After completion of the course, the student will be able to | | Bloom's Knowledge Level (KL) |
|---|---|---|
| **CO1** | Understand software engineering principles, software process models, and Agile methodologies for effective software project development and management. | K2 |
| **CO2** | Demonstrate SRS documentation and software quality assurance practices to ensure software design and development meet applicable standards. | K3 |
| **CO3** | Compare and contrast various design concepts to develop effective software design solutions. | K4 |
| **CO4** | Analyze software testing techniques along with test management practices to ensure and improve software quality. | K4 |
| **CO5** | Analyze software maintenance, project management, estimation techniques, software metrics, risk management, and configuration management concepts in software development. | K4 |

| **CO-PO Mapping** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 2 | - | - | - | - | - | - | - | 1 | 2 | - | 2 | - |
| **CO2** | 2 | 2 | 2 | - | 2 | - | 2 | - | - | 1 | 1 | 3 | 2 | - | 2 |
| **CO3** | 2 | 2 | 3 | - | 2 | - | - | - | - | - | - | 2 | 1 | 3 | - |
| **CO4** | - | 2 | 2 | 2 | 3 | - | - | - | - | - | 1 | 2 | 2 | - | 3 |
| **CO5** | - | - | 1 | - | 2 | - | - | - | - | 2 | 3 | 1 | 3 | 2 | 2 |

**Course Contents / Syllabus**

| **Module 1** | **Introduction to Software Engineering and Development models** | **9 hours** |
|---|---|---|

Evolving role of software, Software Characteristics, Software crisis, silver bullet, Software myths, Software Engineering Phases, Team Software Process (TSP), Emergence of software engineering, Software process, project and product, Software Process Models: Waterfall Model, Prototype Model, Spiral Model, Iterative Model, Incremental Model, Agile Methodology: Scrum Sprint, Scrum Team, Scrum Master, Product Owner, Kanban framework, detailed Case study on Agile methodology.

| Module 2 | Software Requirement and Quality Assurance | 9 hours |
|---|---|---|
| Software Requirement Specifications (SRS): Functional and non – functional requirements, Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modelling, Use Case Diagram, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS. Software Quality Assurance (SQA): Quality concepts, SQA activities, Formal approaches to SQA; Statistical software quality assurance; CMM, The ISO standard. | | |
| **Module 3** | **Software Design** | **9 hours** |
| Design principles, the design process; Design concepts: refinement, modularity: Cohesion, Coupling, Effective modular design: Functional independence, Design Heuristics for effective modularity, Software architecture: Function Oriented Design, Object Oriented Design: OOPs concepts-Abstraction, object, classification, inheritance, encapsulation, UML Diagrams-Class Diagram, Interaction diagram, Activity Diagram, control hierarchy: Top-Down and Bottom-Up Design, structural partitioning, software procedure. | | |
| **Module 4** | **Software Testing** | **9 hours** |
| Software Testing: Testing Objectives, 7 Principals of Testing, Levels of Testing: Unit Testing, System Testing, Integration Testing, User Acceptance Testing, Regression Testing, testing for Functionality and Testing for Performance, Top Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural (White Box Testing Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha, and Beta Testing of Products. Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards, Test Management, Test Planning and Estimation, Test Monitoring and Control, Configuration Management, Risks and Testing, Defect Management, Tool Support for Testing, Effective Use of Tools. | | |
| **Module 5** | **Project Maintenance and Management Concepts** | **9 hours** |
| Software Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Need of Maintenance. Project management concepts, Planning the software project, Estimation: Software Measurement and Metrics, Various Size Oriented Measures-LOC based, FP based, Halestead's Software Science, Cyclomatic Complexity Measures: Control Flow Graphs, Use-case based empirical estimation COCOMO- A Heuristic estimation technique, staffing level estimation, team structures, risk analysis and management. Configuration Management, Software reengineering, reverse engineering, restructuring forward engineering, Clean Room software engineering. Case Tools. | | |
| | | **Total Lecture: 45 Hours** |

**Textbook:**

| S.No | Book Title | Author |
|---|---|---|
| 1 | Software Engineering | KK Aggarwal and Yogesh Singh |
| 2 | Software Engineering: A Practitioner's Approach | RS Pressman |
| 3 | Fundamentals of Software Engineering | Rajib Mall |

**Reference Books:**

| S.No | Book Title | |
|---|---|---|
| 1 | Software Engineering | Pankaj Jalote |
| 2 | Fundamentals of Software Engineering | Ghezzi, M. Jarayeri, D. Manodrioli |
| 3 | Software Engineering | Kassem Saleh |
| 4 | Software Engineering | Ian Sommerville |

| Self-Study Contents Links: | Duration |
|---|---|

| | | |
|---|---|---|
| Moocs | https://onlinecourses.nptel.ac.in/e-learning/preview/noc20_cs68 | 18(hr) |
| Video Lectures | https://www.youtube.com/playlist?list=PLh11ucJN276JGE-oNKj2DDcBQmRKV_SQB | 13(hr) |
| | https://www.youtube.com/watch?v=NlLM3sVF8wY&t=20969s | 6(hr) |
| | https://www.youtube.com/playlist?list=PLZ2ps__7DhBYt5yvXrYAjjWtf5O399Xea | 17(hr) |
| **Web References** | Software Engineering Tutorial - GeeksforGeeks | |

**Mode of Evaluation**

| CIE | | | | | | | ESE | Total |
|---|---|---|---|---|---|---|---|---|
| **ST1** | **ST2** | **ST3** | **TA1**<br>**10** | **TA2**<br>10 | **TA3**<br>**10** | **Attendance**<br>**10** | | |
| **30** | | | **20** | | | | **100** | **150** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| **Course Code:** CCSE0401 | **Course Name: Database Management Systems** | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|
| **Course Offered in:** CSE/CSE-R/IT/M.Tech Int./CSE (Twin)/IT (Twin)/CSE(Prof)/IT (Prof)/M&C/AI/AI (TWIN)/AIML/AIML (TWIN)/CS/CYS/DS/IOT | | **2** | **0** | **0** | **2** |

**Pre-requisite:** Basic understanding of computer fundamentals such as architecture, storage and hardware. In addition, familiarity with data structures, algorithms, and basic programming concepts will be beneficial.

**Course Objectives:** The objective of the course is to introduce about database management systems, with an emphasis on how to organize, maintain and retrieve - efficiently, and effectively - information in relational & non-relational databases.

| **Course Outcome:** After completion of the course, the student will be able to | | Bloom's Knowledge Level (KL) |
|---|---|---|
| **CO1** | Apply ER model for conceptual design of the database. | K3 |
| **CO2** | Execute SQL and apply the normalization to improve the database design. | K3 |
| **CO3** | Understand transaction and concurrency control and execute the concept of PL/SQL. | K3 |
| **CO4** | Analyze and implement effective Relational and Non-Relational database for real-world problems. | K4 |
| **CO5** | Analyze and implement effective Relational and Non-Relational database for real-world problems. | K4 |

| **CO \ PO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 3 | 3 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 3 | 1 | 2 | 1 |
| **CO2** | 3 | 2 | 3 | 3 | 2 | 2 | 1 | 2 | 1 | 2 | 2 | 3 | 2 | 2 | 1 |
| **CO3** | 3 | 2 | 2 | 2 | 2 | 2 | 1 | - | 1 | 1 | 1 | 3 | 3 | 2 | 2 |
| **CO4** | 2 | 2 | 2 | 2 | 3 | 2 | 1 | - | 1 | 2 | 2 | 3 | 3 | 2 | 2 |
| **CO5** | 2 | 2 | 2 | 2 | 3 | 2 | 1 | - | 1 | 2 | 2 | 3 | 3 | 2 | 2 |

**Course Contents / Syllabus**

| **Module 1** | **Introduction about the Database Conceptual Designing** | **7 hours** |
|---|---|---|

**Basic Concept:** Database system concept, architecture, History of Database, Data Independence, Database system Vs File system, Data models & Types of Data Models, schema and instances.

**Data Modelling using the Entity Relationship Model:** ER model concepts, Degree of relationship, Notation for ER diagram, mapping constraints, Generalization, Aggregation, Reduction of an ER diagrams to tables. Extended ER Diagram & reduction of EER. Codd Rules.

**Introduction on SQL:** Types of SQL commands: -DDL, DML, DCL, TCL. Basic of Relation Algebra & Operations, Query Optimization.

| **Module 2** | **Basic of SQL & Normalization** | **7 hours** |
|---|---|---|

**Keys & Types of Keys:** Super key, Candidate Key, Primary Key, Alternative Key, Foreign Key, unique. Constraints and Types of Constraints.

**Use of Functions and Clause:** Aggregate Functions, Scalar Functions, Where, Group by, Having and Order by.

**Use of Predicates:** SQL Operators. Like, Between, Aliases, distinct, limit.

Functional Dependencies, attribute Closure, Normalization & Types of Normalization, Candidate Key, FD Preservation, Lossy and Lossless decomposition, Canonical Cover of FD's, Denormalization.

| **Module 3** | **Transaction and Concurrency Control** | **8 hours** |
|---|---|---|

**Transaction system:** Life cycle of transaction, ACID Properties Schedule & Types of Schedule, Serializability, Recoverability, Deadlock Handling.

**Concurrency Control Techniques:** Concurrency Control, Concurrency control Techniques: Locking Techniques, Timestamping, Validation Based Protocol, Transaction & Data Control: -Grant, Revoke, commit & Rollback.

| **Module 4** | **Introduction of NoSQL With MongoDB** | **8 hours** |
|---|---|---|

**Introduction to NoSQL:** NoSQL Data Models, Overview of NoSQL Databases with their Types, Uses & Features of NoSQL Document Databases, CAP theorem, BASE Vs ACID, Comparison of relational databases to NoSQL stores, uses and deployment: MongoDB, Cassandra, HBASE, Neo4j, Redis and Riak.

**Introduction to MongoDB:** Introduction and Features of MongoDB, MongoDB Operators, MongoDB Collection & Document, MongoDB Shell & their commands, CRUD operations.

Introduction of Cloud Database. MongoDB Cloud product : Stitch, Atlas & Cloud Manager.

| **Total Lecture Hours** | **30 hours** |
|---|---|

**Textbook:**

| **S.No** | **Book Title** |
|---|---|
| **1.** | Database Concepts, McGraw Hill, 7th Edition, 2020 |
| **2.** | Fundamentals of Database Systems, Addision Wesley, 7th edition, 2016 |

**Reference Books:**

| **S.No** | **Book Title** |
|---|---|
| **1.** | Database Systems: A practical Approach to Design, Implementation and Management. Pearson Education,3rd Edition, 2007. |
| **2.** | NoSQL and SQL Data Modeling: Bringing Together Data, Semantics, and Software, Ted Hills, 1st Edition,2016. |

| **Self-Study Contents Links:** | | Duration |
|---|---|---|
| MOOCs | nptel.ac.in/courses/106105175 | 8Weeks |

| | | |
|---|---|---|
| Video Lectures | DBMS L1 Inauguration & Introduction<br>DBMS L2 Introduction to Relational Model<br>DBMS L8C Entity Relationship Model<br>DBMS L8D Entity Relationship Model (Problem Solving and Discussion )<br>DBMS L4A Joins, Set Operations and Aggregate Functions<br>DBMS L4B Joins, Set Operations and Aggregate Functions<br>DBMS L9B Relational Database Design<br>DBMS L9C Relational Database Design<br>DBMS L5A Nested Subqueris<br>DBMS L15 Transactions<br>DBMS L16A Concurrency Control<br>DBMS L16B Concurrency Control<br>DBMS L16C Concurrency Control<br>L64: Non-relational database management systems \| nosql & cap theorem<br>DBMS L19 Distributed Data Stores and NoSQL Databases | 30h |
| | | |
| | | |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| **Course Code: CCSE0403** | **Course Name: Theory of Computation** | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|
| **Course Offered in:CSE/IT/CSE R/CSE(Twin)/IT(Twin)/CSE(AI)/CSE(AI Twin)/CSE(AIML)/CSE(AIML TWIN)** | | **3** | **0** | **0** | **3** |

**Pre-requisite**: Mathematical Foundations , Fundamentals of Computer System

**Course Objectives:** The primary objective of this course is to provide a foundational understanding of Automata Theory and its role in the language, processing systems, also explores their application in fields like Natural Language Processing (NLP), speech recognition,

| **Course Outcome:** After completion of the course, the student will be able to | | Bloom's Knowledge Level (KL) |
|---|---|---|
| **CO1** | Identify the fundamental concepts of automata theory, formal languages and compiler components. | K2 |
| **CO2** | Understand the role of finite automata, regular expressions, and grammar rules in language processing. | K2 |
| **CO3** | Demonstrate context-free grammars, pushdown automata, and syntax-directed translation to construct intermediate code for language processors. | K3 |
| **CO4** | Analyze various parsing strategies, code translation methods, and intermediate representations in compiler phases. | K4 |
| **CO5** | Analyze the functioning of Turing machine models and optimization techniques in code generation for performance improvement. | K4 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| **CO-PO Mapping** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | 2 | 2 | - | - | - | - | - | - | 2 | 3 | 2 | 3 | 2 |
| **CO2** | 3 | 3 | 3 | 2 | 1 | - | - | - | - | - | 2 | 3 | 2 | 3 | 2 |
| **CO3** | 3 | 3 | 3 | 3 | 1 | - | - | - | - | - | 2 | 3 | 2 | 3 | 2 |
| **CO4** | 3 | 3 | 3 | 3 | 1 | - | - | - | - | - | 2 | 3 | 2 | 3 | 2 |
| **CO5** | 3 | 3 | 3 | 3 | 1 | - | - | - | - | - | 2 | 3 | 2 | 3 | 2 |

**Course Contents / Syllabus**

| **Module 1** | **Introduction to Finite Automata and Compiler** | **9 hours** |
|---|---|---|

**Introduction:** Role of Automata and Formal languages, Alphabet, String, Grammar, Language, Chomsky Hierarchy of languages.

**Introduction to Finite State Machine**: Deterministic Finite Automaton (DFA) and Non-Deterministic Finite Automaton (NFA), NFA with ∈-Transition, Equivalence of NFA and DFA,

**Introduction to Compiler:** Translators, Language Processing System, Phases and passes of compilation

| | | |
|---|---|---|
| **Use Case 1 :** Role of Finite Automata in NLP and Speech Recognition. | | |
| **Module 2** | **Regular Expression and Tokenization** | **9 hours** |
| **Regular Expression:** Regular Expression, Regular Sets, Properties of Regular Expression, Finite Automata and Regular Expression, Arden's Theorem, Regular Grammars-Right Linear and Left Linear grammars.<br>**Lexical Analyzer**: Role of lexical Analyzer, Specifications and Recognition of tokens, Lex, | | |
| **Module 3** | **Context free grammar and Push Down Automata** | **9 hours** |
| **Context Free Grammar (CFG):** Definition and Language, Derivations, Parse Trees and Ambiguity, Simplification of CFG,<br>**Push Down Automata:** Definition of the Pushdown Automata, Languages of PDA | | |
| **Module 4** | **Parser and Intermediate Representation** | **9 hours** |
| **Parser**: Role of parser, Top down Parsing-LL (1) parser, Bottom up parsing- shift reduce parser and LR (0), SLR parser, Introduction to Syntax directed Translation,<br>**Intermediate-Code Generation:** Three-Address Code- Quadruples, Triples, Indirect triples<br>**Use Case-2**:Role of CFG and parsing in Voice Assistant | | |
| **Module 5** | **Turing machine and Optimization** | **9 hours** |
| **Turing Machine**: Basic Concept of Turing Machine, Variants of Turing Machine, Universal Turing Machine<br><br>**Code optimization and generation:** Basic Block, Flow graph, DAG, Optimization Techniques | | |
| | **Total Lecture Hours** | **45 hours** |

**Textbook:**

| S.No | Book Title | Author |
|---|---|---|
| 1 | Theory of Computer Science-Automata, Languages and Computation | K.L.P. Mishra, and N. Chandrasekharan , |
| 2 | Compilers: Principles, Techniques and Tools | Alfred V. Aho, Jeffrey D Ullman |

**Reference Books:**

| S.No | Book Title | Author |
|---|---|---|
| 1 | Introduction to languages and the theory of computation | J Martin |
| 2 | Compiler Design in C | Allen I. Holub |

| Self-Study Content Links: | Duration (in Hrs) |
|---|---|

<table>
<tr><td>MOOCs</td><td>https://archive.nptel.ac.in/courses/106/106/106106049/<br>https://archive.nptel.ac.in/courses/106/108/106108113/<br>https://archive.nptel.ac.in/courses/106/108/106108113/<br>https://archive.nptel.ac.in/courses/106/106/106106049/<br>https://archive.nptel.ac.in/courses/106/106/106106049/<br>https://archive.nptel.ac.in/courses/106/108/106108113/</td><td>18 HOURS</td></tr>
<tr><td>Video Lectures</td><td>https://www.youtube.com/watch?v=539Bk9fFOyo<br>https://www.youtube.com/watch?v=6b40kKe2SFg<br>https://www.youtube.com/watch?v=1qOMlqE6LhU<br>https://www.youtube.com/watch?v=1qOMlqE6LhU<br>https://www.youtube.com/watch?v=BR6fHjKFqa0</td><td>42 HOURS</td></tr>
<tr><td>Web References</td><td>Last Minute Notes - Theory of Computation - GeeksforGeeks</td><td>-</td></tr>
<tr><td rowspan="3"></td><td colspan="7">CIE</td><td>ESE</td><td>Total</td></tr>
<tr><td>ST1</td><td>ST2</td><td>ST3</td><td>TA1 10</td><td>TA2 10</td><td>TA3 10</td><td>Attendance 10</td><td></td><td></td></tr>
<tr><td colspan="3">30</td><td colspan="4">20</td><td>100</td><td>150</td></tr>
<tr><td colspan="9"></td></tr>
</table>

| | |
|---|---|
| NIET Greater Noida Autonomous Institute | **NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY**<br>**GREATER NOIDA-201306**<br>**(An Autonomous Institute)**<br>**School of Computer Science and Information Technology** |

| **CourseCode: CCSEO404** | **Course Name: Machine Learning Techniques** | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|
| **Course Offered in: CSE/CSE(R)/IT/M.Tech Int. (CSE)** | | **3** | 0 | 0 | 3 |

**Pre-requisite:** Basic knowledge of python programming

**Course Objectives:** The course objective is to apply foundational machine learning concepts, including data preprocessing, supervised and unsupervised learning algorithms and ensemble methods, to address real-world problems

| **Course Outcome:** After completion of the course, the student will be able to | | Bloom's Knowledge Level (KL) |
|---|---|---|
| **CO1** | Explain fundamental concepts of machine learning including learning paradigms, feature engineering, preprocessing techniques, model building, bias-variance tradeoff, and activation functions. | K2 |
| **CO2** | Apply classification algorithms such as Decision Trees, Support Vector Machines, and K-Nearest Neighbors to solve binary and multiclass classification problems and evaluate model performance using appropriate metrics. | K3 |
| **CO3** | Analyze regression models including linear and logistic regression, evaluate error measures, and implement optimization techniques such as gradient descent for predictive modeling. | K4 |
| **CO4** | Demonstrate probabilistic learning, ensemble methods, clustering techniques, and reinforcement learning approaches for solving supervised and unsupervised learning problems. | K3 |
| **CO5** | Develop Artificial Neural Network models including Perceptron, Multilayer Perceptron, CNN, and RNN architectures for applications such as image classification, text classification, healthcare, e-commerce, and smart city systems. | K4 |

| **CO \ PO** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | 1 | 1 | 2 | – | – | – | – | – | 2 | 3 | – | 2 | 1 |
| **CO2** | 3 | 3 | 2 | 2 | 3 | – | – | – | – | – | 2 | 3 | – | 2 | 2 |
| **CO3** | 3 | 3 | 2 | 2 | 3 | – | – | – | – | – | 2 | 3 | – | 2 | 2 |
| **CO4** | 3 | 3 | 2 | 2 | 3 | – | – | – | – | – | 3 | 3 | – | 3 | 2 |
| **CO5** | 3 | 3 | 3 | 2 | 3 | 1 | 1 | – | – | – | 3 | 3 | 1 | 3 | 2 |

| Course Contents / Syllabus | | |
|---|---|---|
| **Module 1** | **INTRODUCTION TO MACHINE LEARNING** | **9 hours** |
| Learning, Types of Learning, well defined learning problems, designing a Learning System, History of ML, Introduction of Machine Learning Approaches, Features and their types, handing missing data, Dealing with categorical features, Feature Scaling, Feature selection, outlier detection and removal. Introduction to Model Building, Sensitivity Analysis, Underfitting and Overfitting, Bias and Variance, Activation Functions, Data Science Vs Machine Learning. | | |
| **Module 2** | **CLASSIFICATION** | **9 hours** |
| **Classification:** Classification problem, Binary and Multiclass Classification, Decision Boundary<br>**Measuring Performance of Classification Model:** Accuracy, Precision, Recall, F1-Score, ROC-AUC curve, Confusion matrix. **Decision Trees:** Decision Tree Classification Algorithm, Attribute Selection Measures, Variants of Decision Trees. **Large Margin Classifier:** Support Vector Machines, **Instance Based Learning:** K Nearest Neighbor Algorithm. | | |
| **Module 3** | **REGRESSION** | **9 hours** |
| Logistic regression model for binary classification, Cost function of logistic regression model, Regularized logistic regression, One Vs. All strategy for multiclass classification using logistic regression.<br>**Introduction to Regression:** Regression Problem**,** Linear regression model (Simple, Multiple and Polynomial), **Errors Measures:** Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), **Cost function:** Formula, intuition and visualization, Bias and Variance, Overfitting and Underfitting, **Gradient Descent:** Algorithm, intuition, learning rate and Gradient Descent for linear regression. | | |
| **Module 4** | **PROBABILISTIC & CLUSTERING TECHNIQUES** | **9 hours** |
| **Probabilistic Methods for Classification:** Bayesian Learning, Naive Bayes Classifier, Bayesian belief networks. **Tree Ensembles:** Using multiple decision trees, Bagging Vs. Boosting, Sampling with replacement, Random Forest Algorithm, Adda boost, Gradient Boosting, Xgboost Algorithm.<br>**Unsupervised Learning &Clustering:** The problem of Feed, Distance measures<br>**Partition Based Methods:** K-Means Clustering (Intuition, Algorithm, Optimization objective, Initializing K-Means, Choosing number of clusters) K-Medoids and K-Mode clustering algorithms.<br>**Hierarchical Clustering** (Single Linkage, Complete Linkage, AGNES (Agglomerative Nesting), DIANA<br>(Divisive Analysis). Reinforcement Learning | | |
| **Module 5** | **ARTIFICIAL NEURAL NETWORKS** | **9 hours** |
| Neural Networks: Introduction, Perceptron, Multilayer Perceptron, Feed Forward Networks, Back Propagation Networks, Back-propagation through time (BTT), Different types of RNNs. Recurrent Neural Network Model, Notation, hyper-parameter tuning, Image classification, Text classification, Image classification. Understanding and visualizing a CNN.<br>Case Study: Health Care, E Commerce, Smart Cities. | | |
| **Total Lecture Hours** | | **45 hours** |
| **Textbook:** | | |

| S.No | Book Title | Author |
|---|---|---|
| 1. | Machine Learning: A Constraint-Based Approach, Morgan Kaufmann. 2017 | Marco Gori |
| 2. | Machine Learning: The New AI, MIT Press-2016 | Ethem Alpaydin |
| 3. | Neural Networks for Pattern Recognition. New York, NY: Oxford University Press, 1995. | Bishop, Christopher |
| 4. | “Machine Learning”, McGraw-Hill, 2010 | Tom M. Mitchell |

**Reference Books:**

| S.No | Book Title | Author |
|---|---|---|

| | | |
|---|---|---|
| **1.** | Machine Learning: An Artificial Intelligence Approach, Volume 1, Elsevier. 2014 | Ryszard, S., Michalski, J. G. Carbonell and Tom M. Mitchell, |
| **2.** | Machine Learning: An Algorithmic Perspective | Stephen Marsland |
| **3.** | "Introduction to Machine Learning (Adaptive Computation and Machine Learning)", Taylor & Francis 2009, The MIT Press | Ethem Alpaydin, (2004) |
| **4.** | Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies 1st Edition | John D. Kelleher |

| **Self Study Content Link:** | | Duration |
|---|---|---|
| **MOOCs** | 1. https://nptel.ac.in/courses/106106139?<br>2. https://www.coursera.org/learn/machine-learning? | 40 Hrs<br>60 Hrs |
| Video Lectures | 1. https://www.youtube.com/watch?v=9_LY0LiFqRQ<br>2. https://www.youtube.com/watch?v=EYeF2e2IKEo<br>3. https://www.youtube.com/watch?v=wTF6vzS9fy4<br>4. https://www.youtube.com/watch?v=7enWesSofhg | 5 Hrs |
| Web Reference | 1. https://www.tensorflow.org/tutorials?<br>2. https://scikit-learn.org/stable/? | 2 Hrs<br>2 Hrs |

**Mode of Evaluation**

| CIE | | | | | | | End Sem | Total |
|---|---|---|---|---|---|---|---|---|
| **ST1** | **ST2** | **ST3** | **TA1**<br>**10** | **TA2**<br>**10** | **TA3**<br>**10** | **Attendance**<br>**10** | | |
| **30** | | | **20** | | | | **100** | **150** |

| **Course Code:** CAS0401 | **Course Name:** Statistics and Probability | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|
| **Course Offered in: B. Tech. Second Year: Semester-III/IV**<br>**AI/AIML/AI (TWIN)/AIML(TWIN)/CYS/DS/CS/CSE/CSE(TWIN)/CSE R/IT/ IT- Twin/ M. Tech. (Int.)/MCT** | | **3** | **1** | **0** | **4** |

**Pre-requisite:** B. Tech. First Year Syllabus

**Course Objectives:**

1. The objective of this course is to familiarize the students with concepts of Probability and statistical techniques.
2. It aims to equip the students with adequate Knowledge of statistics that will enable them in formulating Problems and solving problems analytically

| **Course Outcome:** After completion of the course, the student will be able to | | Bloom's Knowledge Level (KL) |
|---|---|---|
| **CO1** | Apply the concept of moments, skewness and kurtosis in relevant field. | K3 |
| **CO2** | Apply the concept of correlation, regression and curve fitting with real world problems. | K3 |
| **CO3** | Apply the concept of probability and random variable. | K3 |
| **CO4** | Apply the concept of Mathematical Expectations and Probability Distribution in real life problems. | K3 |
| **CO5** | Apply the concept of hypothesis testing and statistical quality control to create control charts. | K3 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| CO-PO Mapping | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | 2 | 3 | 1 | 1 | - | 1 | 2 | 2 | 2 | 3 | 1 | 1 |
| **CO2** | 3 | 2 | 2 | 3 | 1 | 1 | - | 1 | 2 | 2 | 2 | 3 | 2 | 2 |
| **CO3** | 3 | 2 | 1 | 2 | - | - | - | - | 1 | 2 | 2 | 2 | 1 | 1 |
| **CO4** | 3 | 2 | 2 | 3 | 1 | 1 | - | 1 | 2 | 2 | 2 | 3 | 2 | 2 |
| **CO5** | 3 | 2 | 2 | 3 | 1 | 1 | - | 1 | 2 | 2 | 2 | 3 | 2 | 2 |

**Course Contents / Syllabus**

| **Module 1** | Statistical Techniques-I | **9 hours** |
|---|---|---|

Introduction: Measures of central tendency: Mean, Median, Mode, Standard deviation, Quartile deviation, Moments, Skewness, Kurtosis.

| **Module 2** | Statistical Techniques-II | **9 hours** |
|---|---|---|

Curve Fitting: Straight line, Second-degree parabola, Exponential curve by method of least squares, Correlation and Rank correlation, Linear regression, nonlinear regression and multiple linear regression.

| | | |
|---|---|---|
| **Module 3** | Probability and Random Variable | **9 hours** |
| Random Variable: Definition of a Random Variable, Discrete Random Variable, Continuous Random Variable, Probability mass function, Probability Density Function, Distribution functions.<br>Bivariate Random Variables: Joint probability mass function, joint density and distribution Function, Properties of Joint Distribution function, Marginal Probability mass function, Marginal density Functions, Conditional Distribution for Bivariate Random Variables, Statistical Independence, Central Limit Theorem (Proof not expected) | | |
| **Module 4** | Expectation and Probability Distributions | **9 hours** |
| Expectations of single Random Variable, Mean, Variance, Moment Generating Function, Binomial, Poisson, Normal, Exponential distribution. | | |
| **Module 5** | Hypothesis Testing and Control Charts | **9 hours** |
| Testing a Hypothesis, Null hypothesis, Alternative hypothesis, Level of significance, Confidence limits, Test of significance of means by Z-test and t-test, Chi-square test, F-test, One way ANOVA. Statistical Quality Control (SQC), Control Charts, Control Charts for variables (Mean and Range Charts), Control Charts (p, np and C charts) | | |
| | **Total Lecture Hours** | **45hours** |

**Textbook:**

1. Textbook of Engineering Mathematics- IV Bali, N.P.
2. Advanced engineering mathematics Jain, R.K.
3. Higher engineering mathematics Grewal, B.S.
4. Statistical methods Gupta, S.P.
5. Advanced engineering mathematics ZILL, DENNIS G.

**Reference Books:**

1. Introduction to Probability Models Ross, Sheldon M
2. Probability, Random Variables and Stochastic Processes Papoulis, Athanasios
3. Advanced engineering mathematics Kreyszig, E

| Self-Study Content Links: | | Duration (in Hrs) |
|---|---|---|
| MOOCs | https://www.futureskillsprime.in/course/probability-and-statistics/ | 30 hrs |
| Video Lectures | https://onlinecourses.nptel.ac.in/e-learning/preview/noc21_ma74<br><br>https://onlinecourses.nptel.ac.in/e-learning/preview/noc23_ma77 | 13 hrs |

| | | |
|---|---|---|
| | https://onlinecourses.nptel.ac.in/e-learning/preview/noc23_mg10 | |
| Web Reference | https://www.geeksforgeeks.org/maths/probability-and-statistics/ | 3 hrs |
| | https://www.geeksforgeeks.org/machine-learning/statistics-for-machine-learning/ | 2 hrs |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| LAB Course Code: CCSE0452 | LAB Course Name: Software Engineering and Design Lab | L | T | P | C |
|---|---|---|---|---|---|
| **Course Offered in:CSE /CSE R/IT/M.TechInt** | | **0** | **0** | **2** | **1** |

**Pre-requisite:** Fundamental knowledge of programming concepts, data structures, object-oriented concepts

**Course Objectives:**

The course develops students' skills in software development across different phases of the Software Development Life Cycle (SDLC). It emphasizes software quality assurance, software development models, UML-based design, and test case generation. Students gain practical knowledge of structural, behavioral, and architectural modeling using StarUML along with software testing techniques and test suite design.

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

| | | |
|---|---|---|
| **CO1** | Apply Software Engineering principles and appropriate software development life cycle models in software projects to develop IEEE standard SRS documents. | K3 |
| **CO2** | Design and represent software projects using different UML diagrams based on software design concepts. | K3 |
| **CO3** | Perform software testing using various testing techniques and generate test cases to ensure software quality. | K4 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| CO-PO Mapping | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 | PSO4 | PSO4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 2 | - | 2 | - | 2 | - | - | 1 | 2 | 2 | - | 2 | 1 | 1 |
| **CO2** | 2 | 2 | 3 | - | 2 | - | - | - | - | - | - | 1 | 2 | 2 | - | - |
| **CO3** | 2 | 3 | 2 | 2 | 3 | - | - | - | - | 1 | 2 | 3 | 2 | - | 3 | 3 |

**List Of Practical's (Indicative & Not Limited To)**

| S.No | Experiment (Case Study Based Question) | CO Mapping |
|---|---|---|
| 1 | Team formation and allotment of Mini project: Problem statement, Literature survey, Requirement. Analysis. OR Case Study : (Airline/Railway Reservation System, College ERP System, Banking System, Food Delivery Application, ATM Management System, Online Ticket Booking System, Mobile Recharge Application, Vehicle Rental System etc. | CO1 |
| 2 | To draw the Use Case Diagram for the selected mini project or case study identified in Experiment 1 using UML modelling techniques | CO2 |
| 3 | To draw the Data Flow Diagram (DFD) at different levels for the mini project or case study identified in Experiment 1 | CO2 |
| 4 | To design an Entity Relationship (ER) Diagram with multiplicity for the mini project or case study identified in Experiment 1. | CO2 |
| 5 | To prepare a Software Requirement Specification (SRS) document based on IEEE recommended standards for the mini project or case study identified in Experiment 1. | CO2 |

| | | |
|---|---|---|
| 6 | To draw the State Chart Diagram for the mini project or case study identified in Experiment 1 using UML modelling techniques. | CO3 |
| 7 | To design Object and Class Diagrams for the mini project or case study identified in Experiment 1. | CO3 |
| 8 | To create a Sequence Diagram for Software Design Description (SDD) of the mini project or case study identified in Experiment 1 | CO3 |
| 9 | To create a Collaboration Diagram for Software Design Description (SDD) of the mini project or case study identified in Experiment 1 | CO3 |
| 10 | To draw the Activity Diagram for the mini project or case study identified in Experiment 1 | CO3 |
| 11 | To create the Component Diagram for the mini project or case study identified in Experiment 1 | CO3 |
| 12 | To design the Deployment Diagram for the mini project or case study identified in Experiment 1 | CO3 |
| 13 | Create a test plan document for any application (e.g., Library Management System, identified in Experiment 1) | CO3 |
| 14 | To Create Test Scenario for any application( e.g. Google pay login module) | CO3 |
| 15 | To Design test cases for Gmail inbox, Google Search page. | CO3 |
| 16 | To design test cases for a banking application or Login System using the Boundary Value Analysis (BVA) testing technique. | CO3 |
| 17 | To Design Test Cases for Social Media Application Using Regression Testing. | CO3 |
| 18 | To Design Test Cases Using Equivalence Class Partitioning for Login Authentication System. | CO3 |
| 19 | To design test cases for an Online Payment Gateway using the Decision Table testing technique. | CO3 |
| 20 | Create test cases for a program which determine whether an integer is prime or not by using path testing. | CO3 |
| 21 | Create test cases for a program which determine whether an integer is prime or not by using Cyclomatic complexity. | CO3 |
| 22 | To generate independent execution paths and test cases for a factorial program using Cyclomatic Complexity | CO3 |
| 23 | Consider the code to arrange the nos. in ascending order. Generate the test cases for loop coverage and path testing. Check the adequacy of the test cases through mutation testing and compute the mutation score for each. | CO3 |
| 24 | To Estimation of Test Coverage Metrics and Structural Complexity. | CO3 |
| 25 | Perform Analysis and comparisons of at least 3 testing tools (eg. Qtp,winrunner, postman) | CO3 |
| 26 | Perform Analysis and comparisons bug tracking tool (e.g., Bugzilla, Bug bit) | CO3 |
| 27 | Perform Analysis of any web testing tool (e.g., Selenium) | CO3 |
| 28 | Perform Analysis of any open source-Testing tool (e.g., Test link, Test Rail) | CO3 |
| 29 | To estimate software size, development effort, development time, and project cost using LOC (Lines of Code) based estimation techniques.(refer Experiment-1) | CO3 |
| 30 | To estimate the effort, development time, staffing requirement, and cost of a mini project or Case study identified in experiment 1 using the Basic COCOMO Model. | CO3 |
| | | **Total Hours: 30 hrs.** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| LAB Course Code: CCSE0451 | LAB Course Name: Database Management Systems Lab | L | T | P | C |
|---|---|---|---|---|---|
| **Course Offered in:** CSE/CSE-R/IT/M.Tech Int./CSE (Twin)/IT (Twin)/M&C/AI/AI (TWIN)/AIML/AIML (TWIN)/CS/CYS/DS/IOT | | 0 | 0 | 4 | 2 |

**Pre-requisite:** Basic knowledge of computer fundamentals, programming, data structures, relational database concepts.

**Course Objectives:** To familiarize the students to the basics of database design and implementation.

| **Course Outcome:** After completion of the course, the student will be able to | | **Bloom's Knowledge Level (KL)** |
|---|---|---|
| **CO1** | Design and implement ER/EER models to solve real-world problems and transform them into relational schemas using appropriate database tools. | K3 |
| **CO2** | Apply SQL and PL/SQL to create complex data queries, and procedural operations including triggers and functions, along with database connectivity. | K3 |
| **CO3** | Analyze and ensure database integrity using constraints, and implement unstructured databases using MongoDB with appropriate query operations. | K4 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| COs | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO 10 | PO11 | PSO1 | PSO2 | PSO3 | PSO4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 3 | 3 | 2 | 1 | - | 1 | 2 | 1 | 2 | 3 | 1 | 2 | 1 |
| **CO2** | 3 | 3 | 3 | 3 | 2 | 2 | - | 2 | 1 | 2 | 2 | 3 | 3 | 2 | 1 |
| **CO3** | 2 | 2 | 2 | 2 | 3 | 2 | - | - | 1 | 2 | 2 | 3 | 3 | 2 | 2 |

**List Of Practical's (Indicative & Not Limited To)** | **CO**

| S.No | Experiment | CO Mapping |
|---|---|---|
| 1 | Understand and implement the different ER diagram notation with their relationship and cardinalities. | CO1 |
| 2 | Choose a case study on different domains<br>I. E-commerce Platform<br>II. Inventory Management<br>III. Railway System | CO1 |

| | | |
|---|---|---|
| | IV. Hospital Data Management<br>V. Company System<br>VI. Banking System | |
| 3 | Create an ER diagram for the above chosen case-study (Program-2). Identify the suitable entities and find the relationship and cardinalities between the entities with their relevant attributes. | CO1 |
| 4 | Implement DDL, DML, DCL & TCL commands. | CO2 |
| 5 | Implementation of I/O Constraints: Primary Key, Composite primary Key, Foreign Key, Unique Key. | CO2 |
| 6 | Implementation of Business Constraints: Null, Not Null, Default, Check. | CO2 |
| 7 | Practicing Queries using Like, Between, Aliases, distinct Operator & Predicate and Implement Aggregate functions. | CO2 |
| 8 | Implementation of Queries using Where, Group by, Having and Order by Clause. | CO2 |
| 9 | Create a table EMPLOYEE with following schema:-(Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id, Designation, Salary)<br><br>Write SQL statements for the following query.<br><br>i. List the E_no, E name, Salary of all employees working for MANAGER.<br>ii. Display all the details of the employee whose salary is more than the Sal of any IT PROFF.<br>iii. List the employees in the ascending order of Designations of those joined after 1981.<br>iv. List the employees along with their Experience and daily.<br><br>v. List the employee who are either 'CLERK' or 'ANALYST'.<br>vi. List the employees who joined on 1-MAY-81, 3-DEC-81, 17-DEC-81.<br>vii. List the e_name those are starting with 'S'.<br>viii. Display total salary spent for each job category.<br>ix. Display lowest paid employee details under each manager.<br>x. Display number of employees working in each department and their department name.<br>xi. Display the details of employees sorting the salary in increasing order. | CO2 |

| | | |
|---|---|---|
| | xii. Show the record of employee earning salary greater than 16000 in each department.<br>xiii. Add constraints to check, while entering the empno value (i.e) empno> 100.<br>xiv. Define the field DEPTNO as unique.Create a primary key constraint for the colum(EMPNO). | |
| 10 | Implementation of Queries using set theory operators UNION, INTERSECT, MINUS. | CO2 |
| 11 | Implementation of Queries using Inner Join and Outer Join along with their types. | CO2 |
| 12 | Implementation of Queries ,nested Queries or Sub Queries: IN, NOT IN, Exists, Not Exists, All and Any. | CO2 |
| 13 | Apply the set theory operators, joins and nested queries on the chosen above case-study( Program-2) | CO2 |
| 14 | Understand & implement the Database Connectivity with any programming language. | CO2 |
| 15 | Implementation of Indexing, Views and sequence. | CO2 |
| 16 | I. Write a PL/SQL Program to Add Two Numbers<br>II. Write PL/SQL Program for Fibonacci Series with exception handling<br>III. Write PL/SQL Program to Find Greatest of Three Numbers | CO2 |
| 17 | Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named Areas, consisting of two columns Radius and Area. | CO2 |
| 18 | Write a PL/SQL code block that will accept an account number from the user, check if the users balance is less than the minimum balance, only then deduct Rs.100/- from the balance. | CO2 |
| 19 | Create an implicit cursor for the customers table to update the salary of all customers whose age is less than or equal to 25 by ₹500 and determine whether any rows are affected or not. | CO2 |
| 20 | Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values. | CO2 |
| 21 | Implementation of commit and rollback statement with amount transfer example. | CO2 |
| 22 | Study of Open Source NOSQL Database and installation of MongoDB. | CO3 |
| 23 | Implementation of the MongoDB Shell commands. | CO3 |
| 24 | Implementation of the CRUD Operations in MongoDB. | CO3 |
| 25 | Implementation of Aggregate Functions in MongoDB. | CO3 |
| | | **Total Hours: 60 hrs.** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

**(An Autonomous Institute)**

**School of Computer Science and Information Technology**

| Course Code: CCSE0455 | Course Name: Web Technology | L | T | P | C |
|---|---|---|---|---|---|
| Course Offered in: CSE/CS/IT/CSE(AI)/CSE(AIML)/CSE(IOT)/CSE(AI)/CSE(DS)/CYS/CSE-TWIN/IT-TWIN/CSE-R/M.TECH.INT | | 0 | 0 | 6 | 3 |

**Pre-requisite:** Basic Programming Knowledge, Knowledge of any programming language (e.g., C, C++, Python/Java), Familiarity with basic concepts of Internet.

**Course Objectives:** This course covers different aspect of web technology such as HTML, CSS, Java Script and provide fundamental concepts of Internet, Web Technology and Web Programming. Students will be able to build a proper responsive website.

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

| S.No | Course Outcome | Bloom's Level |
|---|---|---|
| CO 1 | Identify the basic facts and explaining the basic ideas of Web technology and internet. | K2 |
| CO 2 | Applying and creating various HTML5 semantic elements and application with working on HTML forms for user input. | K6 |
| CO 3 | Understanding and applying the concepts of Creating Style Sheet CSS3 and bootstrap. | K3 |
| CO 4 | Analysing and implementing concept of JavaScript and its applications. | K6 |
| CO 5 | Creating and evaluating dynamic web pages using the concept of PHP. | K6 |

CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)

| CO-PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 | PSO 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 3 | 3 | 2 | - | 2 | - | - | - | 2 | 2 | - | 3 | 1 | 1 | 1 |
| CO2 | 3 | 2 | 3 | 2 | 3 | - | - | - | 2 | 1 | 2 | 1 | 2 | 3 | 1 |
| CO3 | 3 | 2 | 3 | - | 3 | - | - | - | 2 | 2 | 2 | 2 | 3 | 2 | 1 |
| CO4 | 3 | 3 | 3 | 2 | 3 | - | 1 | - | 2 | 2 | 2 | 3 | 2 | 3 | 2 |
| CO5 | 3 | 3 | 3 | 2 | 3 | - | - | - | 2 | 2 | 2 | 2 | 1 | 2 | 3 |

**Course Contents / Syllabus**

| Unit 1 | Basics of Web Technology & Testing | 16 hours |
|---|---|---|
| Introduction: Introduction to Web Technology, History of Web and Internet, Connecting to Internet, Introduction to Internet services and tools, Client-Server Computing, Protocols Governing Web, Basic principles involved in developing a web site, Planning process, Types of Websites, Web Standards and W3C recommendations.<br>Web Hosting: Web Hosting Basics, Types of Hosting Packages, Registering domains, Defining Name Servers, Using Control Panel, Creating Emails in Cpanel, Using FTP Client, Maintaining a Website. | | |
| **Unit 2** | **Introduction to HTML & XML** | **16 hours** |
| HTML: What is HTML, DOM- Introduction to Document Object Model, Basic structure of an HTML document, Mark up Tags, Heading-Paragraphs<br>, Line Breaks, Understand the structure of HTML tables. Lists, Working with Hyperlinks, Image Handling, Understanding Frames and their needs, HTML forms for User inputs. New form Elements- date, number, range, email, search and data list, Understanding audio, video and article tags.<br>XML: Introduction, Tree, Syntax, Elements, Attributes, Namespaces, Display, HTTP request, Parser, DOM, XPath, XSLT, XQuerry, XLink, Validator,<br>DTD, Schema, Server. | | |
| **Unit 3** | **Concepts of CSS3 & Bootstrap** | **16 hours** |
| Concept of CSS 3: Creating Style Sheet, CSS Properties , CSS Styling(Background, Text Format, Controlling Fonts) , Working with block elements and objects , Working with Lists and Tables , CSS Id and Class, Box Model(Introduction, Border properties, Padding Properties, Margin properties) CSS Advanced(Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute sector) , CSS Color, Creating page Layout and Site.<br>Bootstrap: Introduction, Bootstrap grid system, Bootstrap Components. | | |
| **Unit 4** | **JavaScript and ES6** | **16 hours** |
| JavaScript Essentials: Introduction to Java Script , Javascript Types , Var, Let and Const Keywords, Operators in JS , Conditions Statements , Java Script Loops, JS Popup Boxes , JS Events , JS Arrays, Working with Arrays, JS Objects ,JS Functions , Using Java Script in Real time , Validation of<br>Forms, Arrow functions and default arguments, Template Strings, Strings methods, Callback functions, Object de-structuring, Spread and Rest | | |
| Operator, Typescript fundamentals, Typescript OOPs- Classes, Interfaces, Constructor etc. Decorator and Spread Operator Difference == & ===, Asynchronous<br>Programming in ES6, Promise Constructor, Promise with Chain, Promise Race. | | |
| **Unit 5** | **JavaScript and ES6** | **16 hours** |
| Introduction to PHP, Basic Syntax, Variables & Constants, Data Type, Operator & Expressions, Control flow and Decision making statements, Functions, Strings, Arrays.<br>Working with files and directories: Understanding file& directory, Opening and closing, a file, Coping, renaming and deleting a file, working with directories, Creating and deleting folder, File Uploading & Downloading.<br>Session & Cookies: Introduction to Session Control, Session Functionality What is a Cookie, Setting Cookies with PHP. Using Cookies with Sessions, Deleting Cookies, Registering Session variables, Destroying the variables and Session. | | |
| | **Total Lecture Hours** | **80 hours** |
| **Textbook:** | | |

| S.No. | Book Details | Author |
|---|---|---|

| | | |
|---|---|---|
| **1** | “Web Technology and Design”, 1nd Edition 2003, New Age International. | C Xavier |
| **2** | “Internet and Web Technologies”, 2nd Edition 2017,Mc Graw Hill Education. | Raj Kamal |
| **3** | “Beginning PHP Laravel”,2nd Edition 2020, kindle Publication. | Oluwafemi Alofe, |
| **Reference Books:** | | |
| **S.No.** | **Book Details** | **Author** |
| **1** | “Collaborative Web Development” 5th Edition 1999, Addison Wesley Publication. | Burdman, Jessica |
| **2** | “Fundamentals of Web Development”,3rd Edition 2016 | Randy Connolly |
| **3** | ” HTML, DHTML, Java Script, Perl & CGI”, 4th Edition 2010 BPB Publication | Ivan Bayross |
| **Self-Study Content Links:** | | **Duration (in Hrs)** |
| **MOOCs** | **https://onlinecourses.swayam2.ac.in/e-learning/preview/nou24_cs09** | **12 Weeks** |

| | | |
|---|---|---|
| **Video Lectures** | https://youtu.be/96xF9phMsWA https://youtu.be/Zopo5C79m2k<br>https://youtu.be/ZliIs7jHi1s<br>https://youtu.be/htbY9-yggB0<br>https://youtu.be/vHmUVQKXlVo https://youtu.be/qz0aGYrrlhU<br>https://youtu.be/a8W952NBZUE<br>https://youtu.be/BsDoLVMnmZs<br>https://youtu.be/1Rs2ND1ryYc<br>https://youtu.be/vpAJ0s5S2t0<br>https://youtu.be/GBOK1-nvdU4<br>https://youtu.be/Eu7G0jV0ImY<br>https://youtu.be/-qfEOE4vtxE<br>https://youtu.be/PkZNo7MFNFg<br>https://youtu.be/W6NZfCO5SIk<br>https://youtu.be/DqaTKBU9TZk<br>https://youtu.be/_GMEqhUyyFM<br>https://youtu.be/ImtZ5yENzgE<br>https://youtu.be/xIApzP4mWyA<br>https://youtu.be/qKR5V9rdht0 | **17m**<br>**8.50m** |
| **Web References** | **Web Development Technologies - GeeksforGeeks** | **-** |

**Mode of Evaluation**

| CIE | | | | | | | ESE | Total |
|---|---|---|---|---|---|---|---|---|
| **ST1** | **ST2** | **ST3** | **TA1**<br>**10** | **TA2**<br>10 | **TA3**<br>**10** | **Attendance**<br>**10** | | |
| **30** | | | **20** | | | | **100** | **150** |

**List Of Practical's (Indicative & Not Limited To)**

| Sr. No. | Program Title | CO |
|---|---|---|
| 1 | A.Overview and Installation of various code editors. | CO1 |
| 2 | B. Overview and Installation of various servers | CO1 |

| | | |
|---|---|---|
| 3 | Implementing HTML program that represent in the document as a start tag, which gives the name and attributes | CO2 |
| 4 | Implementing HTML program that represents a document | CO2 |
| 5 | Implementing HTML program to display your simple CV | CO2 |
| 6 | Creating html document that represents document object model | CO2 |
| 7 | To Create a table to show your class time table. | CO2 |
| 8 | Apply various colors to suitably distinguish keywords , also apply font styling like italics, underline and two other fonts to words you find appropriate , also use header tags. | CO2 |
| 9 | Create a webpage with HTML describing your department use paragraph and list tags | CO2 |
| 10 | Implementing HTML program that for Heading | CO2 |
| 11 | Implementing program that implement paragraph and line-break | CO2 |
| 12 | Use tables to provide layout to your HTML page describing your college infrastructure. | CO2 |
| 13 | Use  and  tags to provide a layout to the above page instead of a table layout | CO2 |
| 14 | Create links on the words e.g. —Wi-Fi and —LAN to link them to Wikipedia pages | CO2 |
| 15 | Insert an image and create a link such that clicking on image takes user to other page | CO2 |
| 16 | Change the background color of the page; At the bottom create a link to take user to the top of the page. | CO2 |
| 17 | Creating HTML program to implement three articles with independent, self-contained content. | CO2 |
| 18 | Creating a XML document that defines the self-descriptive tags | CO2 |
| 19 | Designing XML document that store various book data such as: book category, title, author, year and price | CO2 |

| | | |
|---|---|---|
| 20 | To Describe the various types of XML key components | CO2 |
| 21 | Design XML DTD to define the structure and legal element and attribute of XML document | CO2 |
| 22 | To implement internal and external DTD | CO2 |
| 23 | Use frames such that page is divided into 3 frames 20% on left to show contents of pages, 60% in center to show body of page, remaining on right to show remarks. | CO2 |
| 24 | Design a HTML registration form that takes user name, user password and mobile number with submit button control | CO2 |
| 25 | Design a HTML5 document that implement of date, number, range, email, search and data list. | CO2 |
| 26 | Implementation in HTML5 that include native audio and video support without the need for Flash. | CO2 |
| 27 | Create a simple form to submit user input like his name, age, address and favourite subject, movie and singer. | CO2 |
| 28 | Add few form elements such as radio buttons, check boxes and password field. Add a submit button at last. | CO2 |
| 29 | Design a XML Schema that describes the structure of an XML document. | CO2 |
| 30 | Design a XML document that describe the well-formed XML document | CO2 |
| 31 | Design a XML document of CD Catalog through each <CD> element, and displays the values of the <ARTIST> and the <TITLE> elements in an HTML table | CO2 |
| 32 | Create a XSL document for and taken xml document by you. | CO2 |
| 33 | Create a XSLT document for and taken xml document by you with all steps | CO2 |
| 34 | Design a web page by applying CSS grouping and dimensions property. | CO3 |
| 35 | Demonstrating the CSS Box model with consists of: borders, padding, margins, and the actual content. | CO3 |
| 36 | Add CSS property assign a style or behavior to an HTML element such as: color, border, margin and font-style. | CO3 |

| | | |
|---|---|---|
| 37 | Add To Style Text Elements with Font, Size, and Color in CSS | CO3 |
| 38 | Applying a block element in CSS acquires up the full width available for that content. | CO3 |
| 39 | Design a web page by applying CSS Display and Positioning property. | CO3 |
| 40 | Design a web page by applying CSS Display and Positioning property . | CO3 |
| 41 | Design a web page by applying CSS pseudo classes. | CO3 |
| 42 | Creating a Java Script code to implement all data types. | CO4 |
| 43 | Design a basic structure of Bootstrap Grid system. | CO4 |
| 44 | Design All Bootstrap Components with example. | CO4 |
| 45 | Implementing a program in Java script to implement augmented function. | CO4 |
| 46 | Implementing a program to implement calculator application as real time. | CO4 |
| 47 | Design a HTML form validation using Java Script. | CO4 |
| 48 | Write a program to implement Arrow function with default argument in ES6 | CO4 |
| 49 | Implementing a program in ES6 to implement Template string concepts | CO4 |
| 50 | Implementing a program in ES6 to implement all string methods. | CO4 |
| 51 | Creating a Java Script program to implement Dialog, Confirm and Message Popup Boxes. | CO4 |
| 52 | Implementing a Java Script program to implement onClick and onSubmit event | CO4 |
| 53 | Creating a java script code to implement ‘let’ keyword | CO4 |

| | | |
|---|---|---|
| 54 | Creating a java script code to implement ‘const’ keyword | CO4 |
| 55 | Implementing a program to implement call back functions in ES6. | CO4 |
| 56 | Implementing a program for de-structuring of an array in ES6 | CO4 |
| 57 | Javascript code to implement object and class concepts in Typescript. | CO4 |
| 58 | Write a Typescript program that implement interface and constructor | CO4 |
| 59 | Write a code in typescript that implement decorator and spread operator | CO4 |
| 60 | Create a constant by using define() function with its proper syntax | CO4 |
| 61 | Creating PHP script that return any data types whatever you use. | CO4 |
| 62 | Implementing a code in Java Script to implement Spread and rest operator | CO4 |
| 63 | Javascript code that should compile by Typescript compiler as’tsc’ | CO4 |
| 64 | Write a code in typescript that implement Asynchronous Programming concepts. | CO4 |
| 65 | Write a program in Typescript that implement promise constructor | CO4 |
| 66 | Implementing promise and chain concepts in Typescript | CO4 |
| 67 | Write a code in typescript that implement Promise.race() static method. | CO4 |
| 68 | Crating a program that implement control flow and decision making statement. | CO4 |
| 69 | Creating PHP to implements parameterized function | CO4 |
| 70 | Creating program in PHP to store multiple string and concatenate these string and print it. | CO5 |

| | | |
|---|---|---|
| 71 | Write a PHP script to create and delete directory structure | CO5 |
| 72 | Program to upload and download a file in PHP | CO5 |
| 73 | Implements single dimension array in PHP | CO5 |
| 74 | Write a PHP code to open and close a file in a proper manner | CO5 |
| 75 | Write a PHP script to copying, renaming and deleting a file. | CO5 |
| 76 | PHP program to create and destroy a session. | CO5 |
| 77 | PHP program to set and delete a cookie. | CO5 |
| 78 | PHP program to manually register the session variable | CO5 |
| 79 | PHP program to manually destroy the session variable | CO5 |
| 80 | PHP program to store the session data on one page and would be available on second page. | CO5 |

| | |
|---|---|
| NIET Greater Noida | **NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY**<br>**GREATER NOIDA-201306**<br>**(An Autonomous Institute)**<br>**School of Computer Science and Information Technology** |

| **LAB Course:** CCSCC0452 | **LAB Course Name: Problem Solving Approaches** | **L** | **T** | **P** | **C** |
|---|---|---|---|---|---|
| **Course Offered in: CSE/IT /CSE-TWIN/IT-TWIN** | | **0** | **0** | **2** | **1** |

**Pre-requisite:** Programming Language C/C++ or Java or Python

**Course Objectives:**

Problem-solving in computer programming involves a structured approach to identifying, analyzing, and resolving coding challenges. The process typically includes thoroughly understanding the problem, decomposing it into smaller, manageable parts, designing an appropriate algorithm, implementing the solution through code, and performing testing and debugging to ensure correctness and efficiency

| **Course Outcome:** After completion of the course, the student will be able to | | **Bloom's Knowledge Level (KL)** |
|---|---|---|
| **CO1** | Develop logic-based solutions using control statements, recursion and bit manipulation to solve basic and intermediate computational problems. | K6 |
| **CO2** | Implement and manipulate arrays and strings using fundamental and advanced searching sorting techniques. | K3 |
| **CO3** | Analyze and debug code for logical errors and improve the efficiency of the solution using appropriate data structures and algorithmic patterns. | K4 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| **CO-PO Mapping** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 3 | 2 | 2 | 1 | 2 | 2 | - | - | 2 | 3 | 3 | 2 | 2 |
| **CO2** | 3 | 3 | 2 | 2 | 2 | - | 2 | - | - | - | 2 | 2 | 2 | 2 | 2 |
| **CO3** | 3 | 3 | 2 | 2 | 3 | 1 | 2 | 2 | - | - | 3 | 3 | 3 | 2 | 2 |

**List Of Practical's (Indicative & Not Limited To)**

**Problem Statements need to be discussed in lab session:** Control Statements

**1. Secure Password Generator**

A company wants to create a secure password generator for their employees. The password must be based on specific numeric properties to enhance its complexity and security. Write a program to validate and generate a secure password according to the following rules:

1. ***Prime Number Validation:***
   - The user must input a 3-digit number. The program should first check if the number is a prime number.
   - If it is not a prime number, the user should be prompted to enter another number until a valid prime number is provided.
2. ***Sum of Digits Check:***
   - Once a valid prime number is entered, calculate the sum of its digits. If the sum of the digits is not divisible by 3, ask the user to enter another prime number until a valid one is found.
3. ***Armstrong Number Check:***
   - Check entered prime number is Armstrong or not? If Armstrong are found, prompt the user to enter another prime number and repeat the process.

**Password Generation:**

Concatenate the 1 if entered prime number is Armstrong otherwise 2 with the sum of the digits of the valid prime number to form the secure password.

**Example Scenario:**

***Sample Input***

Enter a 3-digit prime number: 153 Sum of digits of 153 = 9

The sum is divisible by 3. 153 is Armstrong number ***Sample Output***

Secure Password: 19

**2.** Write a function to input electricity unit charges and calculate total electricity bill according to the given condition: For first 50 units Rs. 0.50/unit

For next 100 units Rs. 0.75/unit For next 100 units Rs. 1.20/unit For unit above 250 Rs. 1.50/unit

An additional surcharge of 20% is added to the bill

Vrite a method to generate a secure code which the sum of all possible palindrome numbers between given two numbers. For Example:

**Input**: 10, 80

**Output**: 308

**Explanation**: All palindrome numbers between 10 & 80 are: 11,22,33,44,55,66,77 Password= 11+22+33+44+55+66+77 = 308

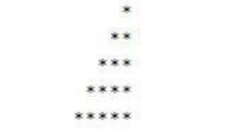
**4.** Draw **the following Patterns for N=5**

Right Triangle Star Pattern

Hollow Right Triangle Star Pattern

Pyramid Star Pattern

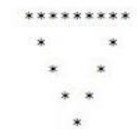
Hollow Pyramid Star Pattern



Mirrored Right Triangle Star Pattern

Hollow Mirrored Right Triangle Star Pattern

Inverted Pyramid Star Pattern



Hollow Inverted Pyramid Star Pattern

Diamond Star Pattern

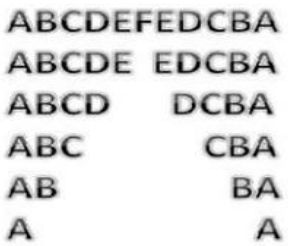
Hollow Diamond Star Pattern

Number pattern 18

**For N=3 print above pattern**

Right Arrow Star Pattern

Left Arrow Star Pattern




**Problem Statements need to be discussed in lab session:** Recursive Approach (Basic)

**1.** Write a program that takes an integer n as input and prints the multiplication table of n from n * 1 to n * 10. The output should clearly show each multiplication step.

| |
|---|
| **2.** Write a program to calculate the sum of all integers from 1 to a given number N. The program should take N as input and output the total sum using iteration or recursion. |
| **3. Find the GCD of Two Numbers Using Recursion:**<br>Write a recursive function to calculate the Greatest Common Divisor (GCD) of two numbers using Euclid's algorithm. The function should take two integers as input and return their GCD. |
| **4. Find the LCM of Two Numbers Using Recursion:**<br>Write a program to compute the Least Common Multiple (LCM) of two numbers using recursion. You may use the relationship LCM(a, b) = \|a * b\| / GCD(a, b) and a recursive function for GCD. |
| **Problem Statements need to be discussed in lab session:** Bit Manipulation |

| |
|---|
| **1.** Write a program to count the number of set bits (1s) in the binary representation of a given integer. The program should efficiently use bitwise operations to perform the task without converting the number to a string. |
| **2.** Write a program that takes a number and a bit position as input and checks whether the bit at that position is set (1) or clear (0). Use bitwise operators to perform the check |
| **3.** Given a number and a position, write a program to toggle (invert) the bit at the given position using bitwise operations. The result should reflect the updated value of the number after flipping the bit. |
| **4.** Write a program to compute the XOR of all numbers from 1 to n using a mathematical pattern (not a loop). Use bitwise XOR properties to achieve an efficient solution. |
| **5.** Given an array of size n-1 containing unique elements from 1 to n, find the missing number using bit manipulation (preferably XOR approach) without sorting or using extra space. |
| **6.** Given an array where all elements repeat twice except two elements that appear only once, write a program to find the two non-repeating elements using bitwise operations in linear time and constant space. |
| **7.** Write a program to check if a given number is a power of two using bit manipulation. A number is a power of two if it has exactly one set bit in its binary representation. |
| **8.** Given two integers A and B, write a program to count how many bits need to be flipped to convert A to B. Use XOR to find differing bits and count the number of set bits. |
| **9.** Write an efficient program to count the total number of set bits in binary representations of all numbers from 1 to n. Optimize the approach using bitwise logic and recursion. |
| **10.** Write a program to calculate the square of a number using only bitwise operations and addition. Do not use multiplication, division, or any power functions. |

| |
|---|
| **11.** Write a function to add two integers using bitwise operations only. Avoid using the + or - operators. Implement logic using XOR and AND operations for binary addition. |

**12.** Write a program to generate the power set (all subsets) of a given set using bitwise representation. Each subset can be represented by a binary number where each bit indicates inclusion of the corresponding element.

**Problem Statements need to be discussed in lab session:** Arrays (Try to use sliding window, prefix sum, cadence, recursion, bit manipulation, two pointer approaches)

**1.** Sarah is assisting the "MathMinds Club" in creating passwords for their online platform. They have a list of numbers, some stable and some unstable. Define a function that can help Sarah calculate the password according to the given scenario.

Scenario:

- There are N numbers provided.
- A number is stable if each digit appears the same number of times.
- A number is unstable if the frequency of its digits is not the same.
- The password is computed as the sum of all stable numbers minus the sum of all unstable numbers.
- Consider only those numbers in the list that have more than equal to three digits.

For example:

Input: N=5 List: 12, 1313, 122, 678, 898

Output: Password: 971

**2.** Given an array of integers, including possible negative values, you are allowed to modify at most one element by doubling its value. The goal is to find the maximum possible sum of any subarray after making this modification.

**Input**:

arr = [-2, 1, -3, 4, -1, 2, 1, -5, 4]

Expected Output:

- Original Maximum Subarray Sum: 6 (achieved from [4, -1, 2, 1])
- Maximum Sum After Modification: 10(achieved from [8, -1, 2, 1], where the value 4 is doubled to 8).

**3.** For a given string, generate a pattern based on the following rules:

**Input:** A string of characters (e.g., "HAT").

*Output: Generate patterns by replacing characters with the numeric value 1 and process the patterns as described below:*

1. **Replace one character at a time with 1:**
   - For each character in the string, replace it with 1, keeping the other characters unchanged.
   - Example for "HAT": 1AT, H1T, HA1
2. **Replace two characters at a time with 1:**
   - Replace every combination of two characters with 1, keeping the remaining character unchanged.
   - If 1s are consecutive, replace them with their sum (e.g., 11T becomes 2T).
   - Example for "HAT":
     11T → 2T, H11 → H2, 1A1
3. **Replace all characters with 1:**
   - Replace all characters in the string with 1.
   - If there are consecutive 1s, sum them up (e.g., 111 becomes 3).
   - Example for "HAT": 111 → 3

*Final Output*

For the string "HAT", the output should be: 1AT, H1T, HA1, 2T, H2, 1A1, 3.

**4.** Given a sorted array arr [] and a target value, the task is to count triplets (i, j, k) of valid indices, such that arr[i] + arr[j] + arr[k] = target and i < j < k.

**Examples**:

Input: arr[] = [-3, -1, -1, 0, 1, 2], target = -2

Output: 4

You are given an array prices[] where prices[i] represents the price of a given stock on day i. You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock. Write a program to return the maximum profit you can achieve from this transaction. If no profit is possible, return 0.

**6.** Find **the "Kth" max and min element of an array:**

Given k, find the k-th smallest and k-th largest element in the array. Input: arr = [7, 10, 4, 3, 20, 15], k = 3

Output: Kth Smallest: 7, Kth Largest: 10

**7.** Sort a binary array with values 0, 1, and 2 using constant space and one pass

**Input:** [0, 2, 1, 2, 0]

**Output:** [0, 0, 1, 2, 2]

**8.** Find **longest consecutive subsequence:**

Return the length of the longest consecutive elements sequence. Input: [1, 9, 3, 10, 4, 20, 2]

Output: 4 (Sequence: 1, 2, 3, 4)

Given a number of bits and a number K. In one flip you can toggle exactly K consecutive bits. With only this flip operation available, convert the string into all 1.

Input String: 0000110000 and K=3

Following are four flip operations by using which all bits converted into 1's. Flip1-1110110000 Flip2-1110110111

Flip3-1111000111 Flip4- 1111111111

If it is not possible to convert all bits into one's then print "IMPOSSIBLE".

Given a list of non-negative integers, arrange them in such a way that they form the largest possible number. Since the result can be very large, return it as a string in **O(N log N)** time complexity.

| **Example-1** | **Example-2** |
|---|---|
| **Input**:<br>N = 5<br>Arr[] = {3, 30, 34, 5, 9}<br>**Output**: 9534330 | **Input**:<br>N = 4<br>Arr[] = {54, 546, 548, 60}<br>**Output**: 6054854654 |

**11.** Given an array arr[] of size n containing distinct integers within the range [1, n+2], find the two missing numbers from the first n+2 natural numbers.

**Constraints:**

- The solution must run in **O(N) time** and use **O(1) extra space**.
- The array does not contain duplicate values.

**Examples:**

**Input:** arr[] = [1, 2, 4, 6, 3, 8], n = 6

**Output:** 5, 7

Given a string str of lowercase alphabets and a number k, the task is to print the minimum value of the string after removal of k characters. The value of a string is defined as the sum of squares of the count of each distinct character present in the string. Return the minimum possible required value. **Examples:**
**Input:** str = “abccc”, k = 1
**Output:** 6
**Input**: str = “aabcbcbcabcc”, k = 3
**Output**: 27
**Expected Time Complexity:** O(n+klog(p))
**Note**: Here n is the length of string and p is number of distinct alphabets and k number of alphabets to be removed.

**13.** Given a non-negative integer **S** represented as a string, remove **K** digits from the number so that the new number is the smallest possible.
**Note :** The given *num* does not contain any leading zero.
**Expected Time Complexity:** O(|S|).

| **Example 1:** | **Example 2:** |
| --- | --- |
| **Input:** | **Input**: |
| S = "149811", K = 3 | S = "1002991", K = 3 |
| **Output:** | **Output:** |
| 111 | 21 |

You are given a two-dimensional grid board[][] of size n * m consisting of English letters and a string target. Your task is to determine whether the target word can be formed by sequentially connecting letters from the grid. You may move to adjacent cells **horizontally or vertically** (not diagonally), and **a cell may not be reused** once it is part of the current path.
**Examples:**

**Input:**
board[][] = [['C', 'A', 'T'], ['R', 'A', 'K'], ['T', 'O', 'N']],
target = "CART" **Output:** true
**Explanation:**
You can trace the word "CART" through the path: C → A → R → T (moving horizontally and vertically, without repeating cells).

**15.** Given an encoded string **s**, the task is to decode it. The encoding rule is:

- **k[encodedString],** where the **encodedString** inside the square brackets is being repeated exactly **k** times. Note that **k** is guaranteed to be a positive integer, and encodedString contains only lowercase english alphabets.

**Note:** The test cases are generated so that the length of the output string will never exceed 10^5.
**Examples:**
**Input:** s = "1[b]"
**Output:** "b"
**Input:** s = "3[b2[ca]]"
**Output:** "bcacabcacabcaca"

****Competitive coding list will be shared with the students.***

**Total Hours: 30 hrs.**

**Mode of Evaluation**

| CIE | | | PE (If mentioned in curriculum) | Total |
|---|---|---|---|---|
| **PS1 10** | **PS2 20** | **PS3 20** | | |
| **50** | | | | **50** |

# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## GREATER NOIDA-201306

(An Autonomous Institute)

School of Computer Science and Information Technology

| Course Code: CNC0401 | Course Name: Artificial Intellgence and Cyber Ethics | L | T | P | C |
|---|---|---|---|---|---|
| Course Offered in: All B.tech branches | | 2 | 0 | 0 | NA |

**Pre-requisite: Basic understanding of AI, Cybercrime, Computer System and Ethics**

**Course Objectives:** The course aims to foster critical thinking about ethical issues, promote responsible use of technology, and ensure students can identify, analyze, and address ethical dilemmas in Artificial Intelligence and cyber domains.

| **Course Outcome:** After completion of the course, the student will be able to | | **Bloom's Knowledge Level (KL)** |
|---|---|---|
| **CO1** | Learn key principles of AI ethics, summarizing ethical considerations and applications in AI development and deployment. | K2 |
| **CO2** | Apply policies and framework for Fairness in AI and Machine Learning. | K3 |
| **CO3** | Apply privacy and security concepts, risk management and regulatory compliance in the field of AI and Cyber Security. | K3 |
| **CO4** | Understand the nature of cybercrimes, the principles of intellectual property rights (IPR), and the legal measures necessary to address and prevent these issues. | K2 |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| CO-PO Mapping | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | - | 1 | - | - | | 1 | 2 | - | - | - | 2 | 1 | 1 | 1 |
| CO2 | 2 | 3 | 2 | - | 2 | 1 | 2 | - | 2 | 1 | 2 | 2 | 1 | 1 |
| CO3 | 2 | 3 | 2 | 1 | 2 | 3 | 3 | - | 2 | 2 | 2 | 2 | 2 | 1 |
| CO4 | 2 | 2 | - | - | 1 | 3 | 3 | - | 2 | 2 | 2 | 2 | 1 | 1 |

**Course Contents / Syllabus**

| Module 1 | An Overview to AI Ethics | 6 hours |
|---|---|---|

DefinitionofAIEthical principles in AI, Sources ofAI data, Legal implications of AI Security Breaches, Privacy and AI Regulations, KeyPrinciplesofResponsibleAI, Transparency and Accountability,Dual-Use Dilemma, Human-Centric Design, Introduction to Cyber Laws and Ethics, Historical Development ofCyberlaws,Legal frameworks.

| Module 2 | Fairness and Favoritism in Machine Learning | 8 hours |
|---|---|---|

| | | |
|---|---|---|
| Introduction to Fairness and Bias in AI, Types of Fairness and Bias, Impact of Bias and Fairness in AI, Techniques for Measuring Fairness and Bias, Techniques for Mitigating Bias, Current Policies and Frameworks for Fairness in AI, Bias in Data Collection,<br>Fairness in Data Processing, Generative AI, Types of Bias in GenerativeAI. | | |
| **Module 3** | **AI Ethics and Cybersecurity Principles** | **8 hours** |
| Importance of Privacy and Security in AI, AI specific Security Tools and Software, Privacy-Preserving Machine Learning (PPML) and Privacy-Preserving Data Mining (PPDM),Risk Management: Risk Assessment and Incident Response, Regulatory Compliance:<br>GDPR, HIPAA, Case Studies:Implementation of AI Ethics guidelines and bestpractices in engineering projects. | | |
| **Module 4** | **Cybercrimes, IPR and Legal Measures** | **8 hours** |
| Types of Cybercrimes and their Impact, Legal measures for Cybercrime Prevention andProsecution, IPR: Copyrights, Trademarks, Patents, and Trade Secrets, Ethical Implications of Intellectual Property, Cyber Securityand Privacy Issues, Cyber Crime Investigations and Digital Evidence Handling, Overview of Indian Cyber Laws (IT Act 2000 and Amendments), Comparative Overview: Indian vs<br>Global Cyber Laws, Case Study: The ATM Heist – Cosmos Bank Cyber Attack (India, 2018). | | |
| | **Total Lecture Hours** | **30 hours** |

**Textbook:**

| S.NO | Book Title | Author |
|---|---|---|
| 1. | Artificial Intelligence: A Guide for Thinking Humans , Penguin Books, 2019. | Melanie Mitchell |
| 2. | Cyber Ethics: Morality and Law in Cyberspace ,7th Edition (2023). | Richard Spinello, Jones & Bartlett Learning |

**Reference Books:**

| S.NO | Book Title | Author |
|---|---|---|
| 1. | Artificial Intelligence and Ethics , BPB Publications, 2023. | S. B. Kishor, Debajit Biswas |
| 2. | Cyber Security and Cyber Laws ,Cengage India, 2022. | Alfred Basta, Nadine Basta, Sattwik Panda, |

| **Self-Study Content Links:** | | **Duration (in Hrs)** |
|---|---|---|
| **MOOCs** | https://swayam-plus.swayam2.ac.in/courses/course-details?id=P_INTEL_02 | 64 hours |
| **Video Lectures** | https://www.youtube.com/watch?v=hVJqHgqF59A<br>https://www.youtube.com/watch?v=VqFqWIqOB1g<br>https://www.youtube.com/watch?v=O5RX_T4Tg24<br>https://www.youtube.com/watch?v=RJZ0pxcZsSQ | 5.2m<br>6.52m<br>.45m<br>5.02m |
| **Web References** | CCS345 Ethics and AI Lecture Notes \| PDF \| Artificial Intelligence \| Intelligence (AI) & Semantics | - |

| **CIE** | | | | | | | **Total** |
|---|---|---|---|---|---|---|---|
| **ST1** | **ST2** | **ST3** | **TA1** | **TA2** | **TA3** | **Attendance** | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | **10** | **10** | **10** | **10** | |
| **30** | | | **20** | | | | **50** |

| Course Code: CNC0302/CNC0402 | Course Name: Environmental Science | L | T | P | C |
|---|---|---|---|---|---|
| Course Offered in: CSE/CS/IT/CSE(AI)/CSE(AIML)/CSE(IOT)/CSE(AI)/CSE(DS)/CSE-R/M.Tech ( Int) / MCT / CSE (CYS) | | 2 | 0 | 0 | NA |

**Pre-requisite:** Basic knowledge of biology, chemistry, ecology, geology, mathematics, and understanding of human impacts on natural systems.

**Course Objectives:**

To help the students in realizing the inter-relationship between man and environment and help the students in acquiring basic knowledge about environment.

| **Course Outcome:** After completion of the course, the student will be able to | | **Bloom's Knowledge Level (KL)** |
|---|---|---|
| **CO1** | Understand the basic principles of ecology and the environment. Ecosystem: Basic concepts, components of ecosystems, food chains and food webs. Ecological pyramids | **K2** |
| **CO2** | Understand the different types of natural recourses like food, forest, Minerals and energy and their conservation | **K2** |
| **CO3** | Understand the different types of pollution, pollutants, their sources, effects, and their control methods. | **K2** |
| **CO4** | Understand the basic concepts of sustainable development, Environmental Impact Assessment (EIA) and different acts related to environment | **K2** |

**CO-PO Mapping (Scale 1: Low, 2: Medium, 3: High)**

| CO-PO MAPPING | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 3 | 2 | 2 | - | 3 | 3 | 2 | 2 | - | 2 | 2 | 2 |
| **CO2** | 3 | 3 | 2 | 2 | - | 3 | 3 | 2 | 2 | - | 2 | 2 | 2 |
| **CO3** | 3 | 3 | 2 | 2 | - | 3 | 3 | 2 | 2 | - | 2 | 2 | 2 |
| **CO4** | 3 | 3 | 2 | 2 | - | 3 | 3 | 3 | 2 | - | 2 | 2 | 2 |

**Course Contents / Syllabus**

| **UNIT 1** | **Basic Principle of Ecology and Biodiversity** | **5 HOURS** |
|---|---|---|

Definition, Scope and basic principles of ecology and environment. Ecosystem: Basic concepts, components of ecosystem. Food chains and food. Webs. Ecological pyramids, Energy flow in ecological systems, Characteristics of different ecosystems. Biogeochemical Cycles: Importance, gaseous and sedimentary cycles. Carbon, Nitrogen, Phosphorus and Sulphur Cycles. Biodiversity and their importance, Threats to biodiversity, major causes, extinction's, vulnerability of species to extinction, IUCN threat categories, Red data book. Strategies for biodiversity conservation, principles of biodiversity conservation in-situ and ex-situ conservation strategies Mega diversity zones and Hot spots, concepts, distribution and importance.

| **UNIT 2** | **Natural Resources and Ecological succession** | **5 HOURS** |
|---|---|---|

Natural resources and associated problems. Forest resources: Use and over- exploitation, deforestation. Timber extraction, mining, dams and their effects on forest and tribal people. Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources. Food resources: World food problems, changes caused by agriculture and over- grazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, and salinity. Land resources: Land as a resource, land degradation, man induced landslides. Equitable use of resources for sustainable lifestyles. Non-Renewable Energy Resources: Fossil fuels and their reserves, Nuclear energy, types, uses and effects, Renewable Energy Resources: hydropower, Solar energy, geothermal, tidal and wind energy, Biomass energy, biogas and its advantages. Ecological succession-Types, stages, examples of ecological succession

| UNIT 3 | Pollution and Waste Management | 5 HOURS |
|---|---|---|
| Air pollution: sources of air pollution, Primary and secondary air pollutants. Origin and effects of SOX, NOX, Cox, CFC, Hydrocarbon, control of air pollution. Water pollution: sources and types of water pollution, Effects of water pollution, Eutrophication, Soil pollution: Causes of soil pollution, Effects of soil pollution, Major sources of and effects of noise pollution on health, Radioactive and thermal pollution sources and their effects on surrounding environment. Solid waste disposal and its effects on surrounding environment, Introduction to E- Waste, Types and classification of E- Waste, Impacts of E- Waste on environment and human health,E-Waste management and recycling., Climate change, global warming, acid rain, ozone layer depletion. | | |
| **UNIT 4** | **Environmental Assessment and Legislation** | **5 HOURS** |
| Women education, Role of NGOs regarding environmental protection, Bio indicators and their role, Natural disasters and disasters management, Aims and objectives of Environmental Impact Assessment (EIA). Salient features of following Acts: Environmental Protection Act, 1986, Wildlife (Protection) Act, 1972.Water (Prevention and control of pollution) Act, 1974. Forest (Conserving) Act, 1980. Definition and concept of sustainability, impacted areas of sustainable development, Global initiative, and issues on sustainable development UN SDGs, System Thinking and Sustainability. | | |
| | **Total Lecture Hours** | **20 HOURS** |
| **Textbook:** | | |
| **S.No** | **Book Title with publication agency & year** | **Author** |
| **1** | Brady, N.C. 1990. The nature and properties of Soils, Tenth Edition. Mac Millan Publishing Co., New York | Brady, N.C |
| **2** | Sodhi G.S. 2005, Fundamentals of Environmental Chemistry: Narosa Publishing House, New Delhi. | Sodhi G.S |
| **3** | Dash, M.C. (1994), Fundamentals of Ecology, Tata Mc Graw Hill, New Delhi. | Dash, M.C |
| **Reference Books:** | | |
| **1** | Rao M.N. and H.V.N. Rao, 1989 : Air Pollution, Tata McGraw Hill Publishing Co. Ltd., New Delhi | Rao M.N. and H.V.N. Rao |
| **2** | A Text Book of environmental Science By Shashi Chawla | Shashi Chawla |
| **Self-Study Content Links:** | | |
| **MOOC'S** | | |
| **VIDEO LECTURES** | https://www.youtube.com/watch?v=T21OO0sBBfc https://www.youtube.com/watch?v=qt8AMjKKPDo<br>https://www.youtube.com/watch?v=brF0RWJyx9w<br>https://www.youtube.com/watch?v=GK_vRtHJZu4 | **7 min 42 sec**<br>**15 min**<br>**3 min 25 sec**<br>**4 min 25 sec** |
| **WEB REFERENCES** | https://www.youtube.com/watch?v=yqev1G2iy20, https://www.youtube.com/watch?v=_74S3z3IO_I,<br>https://www.youtube.com/watch?v=jXVw6M6m2g0 | **28 min**<br>**31 min 35 sec**<br>**11 min** |