NOTIFICATION

No. 170/2026 Dated : 29/09/2026

Subject : Implementation of new syllabus of the subjects of Semester III & IV of the new Under-graduate Program B.E. (Computer Science & Information Technology) as per NEP - 2020 guidelines.

It is notified for general information of all concerned that the authorities of the University have accepted to implement the new syllabus of the subjects of Semester III & IV of the new Under-graduate Program B.E. (Computer Science & Information Technology) as per NEP-2020 guidelines to be implemented from the academic session 2026-2027 onwards as under:

Sd/-
(Dr.Avinash Asanare)
Registrar
Sant Gadge Baba Amravati University

3CSI200PC/3CS200PC/3DS200PC/3AD200PC/3KE200PC/

DISCRETE STRUCTURE AND GRAPH THEORY

Course Pre-requisite: Basic knowledge of Mathematics

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of Discrete Structure by being able to do each of the following:

- Use mathematically correct terminology and notation.
- Constructs correct direct and indirect proofs.
- Apply logical reasoning to solve a variety of problems

Course Outcomes (Expected Outcome):

On completion of the course, the students will be able to:

1. Analyze and express logic sentences in terms of predicates, quantifiers, and logical connectives.
2. Derive the solution for a given problem using deductive logic and prove the solution based on logical inference.
3. Classify algebraic structure for a given mathematical problem.
4. Perform combinatorial analysis to solve counting problems.
5. Perform operation on trees data structures.
6. Develop the given problem as graph networks and solve with techniques of graph theory

SYLLABUS:

Unit I: The Foundations: Logic and Proofs (Hours: 07)

Propositions, Truth Tables, Compound Propositions, Logical Operators, Logic and Bit Operations; Logical Equivalences, Normal Forms, De Morgan's Laws, Satisfiability: Applications and Solving Problems; Predicates, Quantifiers: Restricted Domains, Precedence, Logical Equivalences; Rules of Inference for Propositional Logic.

Unit II: Sets, Functions and Relation (Hours: 07)

Introduction, Venn Diagrams, Subsets, Size of a Set, Power Sets, Cartesian Products, Set Notation with Quantifiers, Truth Sets and Quantifiers, Set Operations ,Functions, Inverse Functions, Compositions and Graphs of Functions ,Partial Functions; Sequences, Summations; Countable Sets, An Uncountable Set; Functions as Relations, Relations on a Set, Properties of Relations, Combining Relations; Representing Relations Using Matrices; Representing Relations, Closures of Relations, Equivalence Relations.

Unit III: Algebraic Structures (Hours: 07)

Algebraic Systems: Examples and General Properties; Semigroups and Monoids: Homomorphism of Semigroups and Monoids, Subsemigroups and Submonoids; Groups: Definitions, Subgroups and Homomorphisms, Cosets and Lagrange's Theorem, Normal Subgroups, algebraic Systems with Two Binary Operations; Group Codes: The Communication Model and Basic Notions of Error Correction, Hamming Distance.

Unit IV: Boolean Algebra (Hours: 07)

Lattices, Boolean Algebra: Boolean Functions, Representing Boolean Functions, sum of product expansions, Product of sum expansion Functional Completeness, Logic Gates, Combinations of Gate, Minimization of Circuits, Karnaugh Maps.

Unit V: Tree (Hours: 07)

Introduction, Rooted Tree, ordered rooted tree, tree as model, Properties of Trees, Applications of tree, Binary Search Trees, Decision Trees, Prefix Codes, Huffman Coding, Game Trees, Tree traversal, Preorder Traversing, Inorder Traversing, Post order Traversing, Spanning Tree, Minimum spanning tree

Unit VI: Graph (Hours: 07)

Graph Models; Basic Terminology, Special Simple Graphs, Bipartite Graphs, Matchings, Applications of Special Types of Graphs, New Graphs from Old; Graph Representation, Adjacency and Incidence Matrices, Isomorphism of Graphs, Determining Isomorphism; Paths, Connectedness in Undirected Graphs and Directed Graphs, Paths and Isomorphism, Counting Paths Between Vertices; Euler Paths and Circuits, Hamilton Paths and Circuits, Applications of Hamilton Circuits; Planar Graphs: Euler's Formula, Kuratowski's Theorem; Graph Coloring: Introduction, Applications of Graph Coloring.

Text Books:
1. Kenneth H. Rosen: Discrete Mathematics and Its Applications, 7th Edition, McGraw-Hill.
2. J. P. Tremblay and R. Manohar: Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw-Hill Edition, McGraw-Hill.

Reference Books:
1. Norman L. Biggs: Discrete Mathematics, 2nd Edition, Oxford University Press.
2. Seymour Lipschutz and Marc Lars Lipson: Schaum's Outline of Theory and Problems of Discrete Mathematics, 3rd Edition, Schaum's Outlines Series, McGraw-Hill.
3. C. L. Liu and D. P. Mohapatra: Elements of Discrete Mathematics: A Computer Oriented Approach, 3rd Edition, Tata McGraw-Hill, McGraw-Hill.

****

3CSI201PC

OBJECT ORIENTED PROGRAMMING IN PYTHON

Course Prerequisite: Computer Programming

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding:
- To explore the principles of Object Oriented Programming (OOP) such as data abstraction, encapsulation, inheritance and polymorphism.
- To use the object-oriented paradigm in program design.
- Infer the Object-oriented Programming concepts in Python.
- To develop the ability to write database applications in Python

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:
1. Apply Object Oriented approach to design software.
2. Implement programs using classes and objects.
3. Specify the forms of inheritance and use them in programs.
4. Analyze polymorphic behavior of objects.
5. Design and develop GUI programs.
6. Develop Applets for web applications.

SYLLABUS

Unit I: Introduction to Object Oriented Programming: (Hours:07)
Introduction, Need of OOP, Principles of Object-Oriented Languages, Procedural Language Vs OOP, Application of OOP, Java Virtual Machine, Java features, Program Structures. Java Programming Constructs: Variables, Primitive data types, Identifier, Literals, Operators, Expressions, Precedence Rules and Associativity, Primitive Type Conversion and Casting, Flow of Control.

Unit II: Classes and Objects: (Hours:07)
Classes, Objects, Creating Objects, Methods, Constructors, Cleaning up Unused Objects, Class Variable and Methods, this keyword, Arrays, Command Line Arguments.

Unit III: Inheritance, Interfaces and Packages: (Hours:07)
Inheritance: Inheritance vs. Aggregation, Method Overriding, super keyword, final keyword, Abstract class. Interfaces: Defining interfaces, Implementing interfaces, Accessing interface variables, Extending interfaces. Packages: Packages, java.lang package

Unit IV: Python Collection class (Hours:07)
Classes and Objects in Python, properties and methods, 'class' key word, Instantiation, init__() Function, self parameter, The pass Statement, Delete Object Properties, Demonstration of class, objects, member Methods and data.

Unit V: Inheritance in Python: (Hours:07)
Inheritance in Python, Parent / Base class, Child / derived class, inheritance syntax, __init__() Function and Inheritance, function over riding, super() method, self, Multi Level Inheritance, Demonstration examples.

Unit VI: Python Functions: (Hours:07)
Syntax, definition def key word, parameters, return data, function call, Function Parameters and Arguments, Arbitrary Arguments, *args, Keyword Arguments, kwargs, Arbitrary Keyword Arguments, **kwargs, Default Parameter Value, Passing a List as an Argument, How Function return Values, The pass Statement. Demonstration programs.

Text Books:
1. Sachin Malhotra and Saurabh Choudhary: Programming in Java, Oxford University Press 2010.
2. Herbert Schildt: Java Complete References (McGraw Hill)

Reference Books:
1. H.M.Dietel and P.J.Dietel, "Java How to Program" Pearson Education/PHI, Sixth Edition.
2. E. Balagurusamy: Programming with Java (McGraw Hill)
3. Dr. R. NageswaraRao: Core Java An Integrated Approach (Dreamtech)

4. Khalid Mughal: A Programmer"s Guide to Java Certification, 3rdEdition (Pearson)
5. MichałJaworski, Tarek Ziadé, "Expert Python Programming", Packt Publishing Ltd., Third Edition, ISBN-9781789808896, 2019.

****

3CS202PC/3DS202PC/3AD202PC/3KE202PC/3CSI202PC

DATA STRUCTURES

Course Pre-requisite:
Fundamentals of programming Language & Logic Building Skills.

Course Objectives:
- To understand the linear and nonlinear data Structures and its memory representations.
- To perform different operations on data structures such as insertion, deletion, searching and
- traversing.
- To understand various data searching and sorting methods with its complexity.
- To introduce various techniques for representation of the data in the real world.

Course Outcomes:
On completion of the course, the students will be able to-
1. Apply various linear and nonlinear data structures
2. Demonstrate operations like insertion, deletion, searching and traversing on various data structures
3. Examine the usage of various structures in approaching the problem solution.
4. Choose appropriate data structure for specified problem domain

SYLLABUS

Unit I: Introduction to Data Structures: (Hours:07)
Introduction to Data structures, Data Structure Operations, Algorithmic Notation, Complexity of algorithms. String processing: storing strings, character data type, string operations, word processing, and pattern matching algorithms.

Unit II: Array& Record Structure: (Hours: 07)
Linear arrays: Memory Representation of arrays, traversing linear arrays, insertion & deletion operations, Bubble sort, Linear search and Binary search algorithms. Multi-dimensional arrays, Pointer arrays. Record structures.

Unit III: Linked lists: (Hours: 07)
Linked lists: Memory Representation of Linked List, traversing a linked list, searching a linked list. Memory allocation & garbage collection. Insertion & deletion operations on linked lists. Header linked lists, Two- way linked lists.

Unit IV: Stack & Queue: (Hours: 07)
Stacks: Sequential Memory Representation of Stack, Arithmetic expressions: Polish notation. Quick sort, Recursion, Tower of Hanoi. Queues: Sequential Memory Representation of Queue, DeQueue, Priority queues.

Unit V: Trees: (Hours: 07)
Introduction to Trees, Binary trees, Memory Representation of Binary Tree, Traversing binary trees, Header nodes, Binary Search Tree, Heap and heap sort, Path length & Huffman's algorithm.

Unit VI: Graphs & Sorting Algorithms: (Hours: 07)
Introduction to Graphs, Memory representation of graphs, Warshalls' algorithm, operations on Graphs, Breadth First Search, Depth First Search. Sorting : Insertion Sort, Selection Sort, Radix sort, Merge Sort.

Text Books:
1. Seymour Lipschutz: Data Structures, Schaum's Outline Series, McGraw-Hill, International Editions.
2. Trembley, Sorenson: An Introduction to Data Structures with Applications, McGraw Hill.

Reference Books:
1. Ellis Horowitz, Sartaj Sahni: Fundamentals of Data Structures, CBS Publications.
2. Data Structure Using C, Balagurusamy.
3. Standish: Data Structures in Java, Pearson Education.

****

3CS400EL / 3KE400EL /3DS400EL /3AD400EL / 3CSI400EL

COMMUNITY ENGAGEMENT PROJECT / FIELD PROJECT

Suggested Activities for Community Engagement/ Field Project

1. Digital Empowerment & Awareness

📌 **Bridging the Digital Divide:**

- Train villagers on **mobile banking**, UPI payments, and accessing government e-services (ration card, Aadhaar updates, PM-KISAN).
- Conduct **awareness workshops** on the **Digital India** mission and help locals register for online services.
- Develop **multilingual video tutorials** on digital literacy, mobile safety, and secure online transactions.
- Provide **training on using government portals** like eNAM (National Agricultural Market) and GeM (Government e-Marketplace).

📌 **Smart Village Initiative:**

- Set up **public Wi-Fi hotspots** in collaboration with local authorities.
- Assist in **digitizing local businesses**, helping small vendors and farmers list their products online.
- Train rural students in **basic coding and website development** for local business promotion.

2. Smart Agriculture & Technology Adoption

📌 **Tech-Driven Farming Solutions:**

- Develop an **AI-based chatbot** in regional languages to assist farmers with crop-related queries.
- Install **IoT-based soil sensors** and educate farmers on precision farming techniques.
- Train farmers in **disease detection using AI-based mobile apps** for plant health monitoring.
- Introduce **solar-powered irrigation systems** to reduce dependency on grid electricity.

📌 **Sustainable Agricultural Practices:**

- Conduct **vermicomposting and organic farming workshops** to reduce chemical usage.
- Set up **bio-fertilizer production units** using locally available waste.
- Promote **agroforestry** as a long-term sustainable practice.

3. Water & Resource Management

📌 **Water Conservation & Management:**

- Conduct **rainwater harvesting awareness** campaigns.
- Develop a **low-cost water filtration system** for villages facing potable water issues.
- Implement **drip irrigation systems** for water conservation in farming.
- Map and revive **traditional water bodies** in the region.

📌 **Energy-Efficient Practices:**

- Promote the use of **solar lamps and bio-gas units** in rural households.
- Conduct workshops on **converting agricultural waste into biomass energy**.

4. E-Commerce & Rural Entrepreneurship Development

📌 **Market Linkage & Skill Development:**

- Assist farmers in selling their products on **e-commerce platforms** like Amazon Kisan Store, Flipkart Krishi, and eNAM.
- Conduct training on **branding, packaging, and direct farm-to-market sales** strategies.
- Develop an **online directory of rural artisans** to promote handicrafts through social media.

📌 **Rural Women Empowerment:**

- Train rural women in **handicraft-based digital entrepreneurship** (e.g., selling products on Etsy, Meesho).
- Organize workshops on **self-help groups (SHGs)** and micro-financing opportunities.
- Introduce women to **low-cost, home-based business ideas** like making organic soaps, candles, or eco-friendly bags.

5. Health & Hygiene Awareness

📌 **Basic Healthcare Services:**

- Conduct **free medical check-up camps** with local health professionals.
- Develop an **AI-powered mobile application** to spread verified medical information in local languages.
- Train rural women on **maternal health, child nutrition, and menstrual hygiene management**.

📌 **Sanitation & Waste Management:**

- Organize **waste segregation drives** and teach villagers about composting.
- Introduce **biodegradable waste recycling** to generate biogas and organic manure.
- Develop **low-cost water purification units** for households.

6. Education & Skill Development

📌 **STEM Awareness Programs:**

- Conduct **STEM (Science, Technology, Engineering, and Mathematics) workshops** in rural schools.
- Develop **interactive STEM kits** for students to experiment with physics and engineering principles.
- Train rural students in **3D printing and CAD software** for technical skill enhancement.

📌 **Career & Competitive Exam Guidance:**

- Offer **guidance sessions for competitive exams** like JEE, NEET, UPSC, and MPSC for rural students.
- Organize **spoken English and communication skill development programs**.
- Help students with **resume writing, online job portals, and freelancing opportunities**.

7. Cyber Security & Online Safety

📌 **Safe Digital Practices:**

- Conduct **cybersecurity awareness workshops** on recognizing fraud, scams, and phishing attacks.
- Train rural youth on **secure online transactions** and social media privacy settings.
- Provide **hands-on training on data encryption and password management**.

8. Renewable Energy & Sustainability

📌 **Eco-Friendly Energy Solutions:**

- Promote **solar-powered microgrids** for electrification in remote areas.
- Educate villagers on **alternative energy sources** like wind and hydro power.
- Develop and implement **low-cost biogas plants** for households.

📌 **Green Village Initiative:**

- Organize **tree plantation drives** in deforested areas.
- Conduct **sustainability workshops** focusing on reducing carbon footprints.

9. Waste Management & Circular Economy

📌 **Plastic-Free Village Campaign:**

- Conduct **awareness programs on plastic pollution and alternatives**.
- Organize **recycling drives and upcycling workshops** for students and villagers.
- Train women-led SHGs in **eco-friendly bag and paper product manufacturing**.

📌 **Low-Cost Housing Solutions:**

- Develop a model for **affordable, sustainable housing** using locally available materials.
- Promote the use of **mud bricks, bamboo, and recycled materials** for low-cost homes.

10. Smart Infrastructure & Community Development

📌 **Smart Village Concept:**

- Develop a **GIS-based village mapping system** for better resource allocation.
- Set up **community radio stations** to disseminate important information.
- Install **smart lighting solutions** (solar-powered street lights) for energy efficiency.

📌 **Community-Driven Innovation:**

- Encourage students to develop **innovative low-cost solutions** to local problems.
- Facilitate **co-creation workshops** where villagers and students collaborate on ideas.

Impact of These Activities

✅ **Bridges the gap between classroom learning and real-world challenges**.
✅ **Encourages students to develop AI/IoT-based solutions for rural challenges**.
✅ **Promotes sustainability and digital literacy among rural communities**.
✅ **Fosters a sense of social responsibility and entrepreneurship in students**.

This course emphasizes experiential learning through community-based projects where students will apply computing knowledge and emerging technologies to solve real-world problems faced by society. Students will actively engage with local communities, identify challenges, and develop technology-driven solutions that foster social development and digital empowerment.

Course Objectives:

- Provide students with practical exposure to how computer science and digital technologies can address societal challenges.
- Enable students to design and deploy small-scale ICT/AI/IoT-based solutions in real community contexts.
- Develop digital awareness, problem-solving, and teamwork skills through field-based activities.
- Encourage social responsibility and ethical use of technology.

Course Outcomes:

After completing the course, students will be able to:
1. Identify societal and community problems that can be addressed through computer science applications.
2. Design small-scale prototypes or awareness modules leveraging ICT, AI, IoT, or data science.
3. Demonstrate hands-on application of emerging technologies in community settings.
4. Collaborate with local stakeholders to deploy and evaluate solutions.
5. Document outcomes of field projects and reflect on experiential learning.

Details of Syllabus / Suggested Activities:

Perform 5 - 6 Practical activities (or equivalent projects) from the list should be performed.

1. **Basic Computer Literacy Sessions** – Teach elderly or underprivileged groups how to use smartphones, send SMS/WhatsApp messages, or browse the internet safely.
2. **Digital Document Assistance** – Help community members create and manage Aadhaar-linked services, PAN, ration card updates, or digital health IDs.
3. **Awareness on Cyber Frauds** – Design and distribute pamphlets/posters on avoiding phishing calls, OTP fraud, and fake links.
4. **Assist Farmers in Using Kisan Apps** – Demonstrate the use of government agriculture apps like *PM-Kisan*, *IFFCO Kisan*, or *Kisan Suvidha*.
5. **Energy Saving Awareness Drive** – Conduct a door-to-door survey and create posters on saving electricity, switching to LED bulbs, and responsible usage.
6. **Plastic-Free Awareness Campaign** – Develop posters or short plays/skits to educate the community on reducing single-use plastics.
7. **Help in Using E-Rickshaw / EV Charging Apps** – Guide local drivers or users in using digital payment methods for EV charging stations.
8. **Community Survey on Waste Management** – Collect data about how households dispose of waste and suggest better segregation methods.
9. **Health Awareness Posters** – Prepare easy-to-understand visual guides about vaccination schedules, nutrition charts, or basic hygiene practices.
10. **Promote Local Library Digitization** – Help libraries or schools catalog their books using Excel or simple databases.
11. **Assist Shopkeepers in Using QR Codes** – Help small vendors adopt UPI/QR code payments and train them in transaction safety.
12. **Create a Local Bus/Transport Timetable** – Collect schedules from local bus stations and prepare a digital or printed timetable for commuters.
13. **Assist in Online Railway / Bus Ticket Booking** – Train rural residents in booking affordable tickets safely through apps/portals.
14. **Organize an "E-Waste Awareness Camp"** – Collect old gadgets/e-waste and educate people about proper disposal/recycling.
15. **Create Safety Awareness Charts** – For road safety, emergency contacts, and first aid—distribute in schools and community centers.
16. **Promote Rainwater Harvesting Practices** – Conduct awareness sessions and create models/demos of small harvesting systems.
17. **Awareness on Social Media Usage** – Teach responsible use of Instagram, Facebook, WhatsApp to school students; warn against oversharing personal data.
18. **Assist Local NGOs** – Help them in preparing basic presentations, brochures, or email communication for their causes.

19. **Basic Typing and Resume Help** – Help unemployed youth learn typing, create email IDs, and prepare resumes using MS Word.
20. **Digital Health Awareness** – Guide villagers in using *Aarogya Setu*, *ABHA Digital Health Card*, or booking doctor appointments online.
21. **Water Conservation Awareness** – Create charts and posters with tips on saving water at household level.
22. **Awareness on Government Scholarships** – Collect and distribute information about scholarships available for students in rural areas.
23. **Assist Schools in Online Examination Practice** – Help students get familiar with mock online tests using simple quiz apps.
24. **Digital Maps for Locality** – Create simple Google Maps entries for important community landmarks (clinics, schools, ATMs).
25. **Conduct a Poster-Making or Slogan Competition** in schools on themes like *Digital India*, *Green Energy*, or *Cyber Safety*.
26. Conduct a **digital literacy workshop** in a rural/urban community: train locals on safe mobile banking, UPI payments, and accessing government portals.
27. Develop and demonstrate a **basic Android app** for local needs (e.g., crop calendar, community notice board, health reminders).
28. Create **AI/ML-based awareness tools** (e.g., chatbot for agricultural queries or student career guidance).
29. Field visit to a rural school to conduct **STEM awareness sessions** using simple coding and robotics kits.
30. Develop a **website or e-commerce page** for local artisans, SHGs, or farmers to promote their products.
31. Implement an **IoT prototype** (e.g., soil moisture sensor, smart street lighting demo) and explain its community utility.
32. Survey local cybercafés/educational institutes to study **cybersecurity practices** and conduct awareness drives on online safety.
33. Develop and present **data-driven insights** on a local issue (e.g., water usage, electricity consumption, health trends) using open datasets.
34. Assist schools/NGOs in digitizing their records using **databases or cloud storage** and provide basic training.
35. Create and distribute **digital awareness content** (videos, infographics) on health, hygiene, and safe technology practices.

And similar relevant activities /projects on the same lines: curated and approved by the concerned Authority from time to time.

Tentative Report Template for Community Engagement Project

Title Page

- Project Title:
- Student Name & Roll Number:
- Faculty Guide Name:
- Date of Submission:

---

1. Introduction
   - Brief overview of the selected activity.
   - Why this issue was chosen.
2. Objectives
   - Clear statement of objectives (What problem is being addressed?).
3. Methodology
   - Approach taken (Survey, Workshops, Development of Solutions, etc.).
   - Tools and Technologies used (if applicable).
4. Execution & Implementation
   - Step-by-step description of how the activity was conducted.
   - Challenges faced and how they were overcome.
5. Observations & Impact Assessment
   - Community response and engagement levels.
   - Quantitative & qualitative impact (e.g., number of farmers trained, improved efficiency, awareness raised).
6. Learnings & Outcomes
   - Key takeaways for students.
   - Skills developed and knowledge gained.
7. Suggestions & Future Scope
   - Recommendations for scaling up the project.
   - How technology can further enhance the solution.
8. References & Acknowledgments
   - Any research material, websites, or experts consulted.
   - Acknowledgment of community members or organizations who contributed.

Evaluation Criteria (Total: 50 Marks)

Evaluation Parameter	Marks (Int)	Marks (Ext)
Problem Identification & Understanding	05	05
Implementation & Execution	05	05
Innovation & Use of Technology	05	05
Community Engagement & Feedback	05	05
Report Quality, Presentation, Future Scope & Recommendations.	05	05

****

3CSI203PC

OBJECT ORIENTED PROGRAMMING IN PYTHON - LAB.

Course Prerequisite: Basics of programming Language

Course Objectives:

Throughout the course, the students will be expected to demonstrate their understanding of Object Oriented Programming in Python Lab by being able to do each of the following:

- Interpret the use of procedural statements like assignments, conditional statements, loops and function calls. Learn the syntax and semantics and create the functions in Python.
- Infer the supported data structures like lists, dictionaries and tuples in Python.
- Illustrate the application of matrices and regular expressions in building the Python programs.
- Discover the use of external modules in creating excel files and navigating the file systems.
- Describe the need for Object-oriented programming concepts in Python.

Course Outcomes (Expected Outcome):

On completion of the course, the students will be able to:

1. Apply the Python language syntax including control statements, loops and functions.
2. Understand the core data structures like lists, dictionaries, tuples and sets in Python to store, process and sort the data.
3. Interpret the concepts of Object-oriented programming as used in Python
4. Identify the external modules for creating and writing data to excel files and inspect the file operations to navigate the file systems.

List of Experiments:

Minimum 12 Experiments based on each concept are to be performed covering the entire syllabus.

At least two experiments should be beyond syllabi based on learning of syllabi, This list is for reference only one can prepare a list according to the syllabus:

1. Write a program to understand basic Python interpreter
2. Write a program to perform String Manipulation and Functions
3. Write a program to implement Python Data structures
4. Implement Classes and Objects using Python
5. Design a program to understand Overloading in Python
6. Design a program to understand Overriding in Python
7. Implement Inheritance using Python
8. Develop a Python code for Information hiding
9. Implement Python's Modules and Packages
10. Writ a program to illustrate concept of File handling
11. Write a code to load Data with library such as Pandas-Numpy.

Text Book: Martin C Brown, "Python: The Complete Reference", McGraw Hill

Reference Books:

1. Larry Lutz, "Python for Beginners: Step-By-Step Guide to Learning Python Programming", Create Space Independent Publishing Platform, First edition, ISBN- 1717410588, 81717410580, 2018
2. Nicholas Ayden, "Python Programming", Independently Published, First Edition, ISBN-1707051933, 9781707051939, 2019.
3. Michał Jaworski, Tarek Ziadé, "Expert Python Programming", Packt Publishing Ltd., Third Edition, ISBN-9781789808896, 2019.

****

3CS204PC/3DS204PC/3AD204PC/3KE204PC/3CSI204PC

DATA STRUCTURE - LAB.

Course Pre-requisite:
Basics of programming Language & Logic Building Skills

Course Objectives:
- To understand the linear and nonlinear data Structures and its memory representations.
- To perform different operations on data structures such as insertion, deletion, searching and traversing.
- To understand various data searching and sorting methods with its complexity.
- To introduce various techniques for representation of the data in the real world.

Course Outcomes:
On completion of the course, the students will be able to-
1. Apply various linear and nonlinear data structure.
2. Demonstrate operations like insertion, deletion, searching and traversing on various data structures
3. Examine the usage of various structures in approaching the problem solution.
4. Choose appropriate data structure for specified problem domain

List of Experiments:
This is a sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi.

1. Write a program to find out largest number from the array and also find it's location.
2. Write a program to traverse an array and find the sum and average of data elements from an array.
3. Write a Program to a) insert an element in an array b) delete an element from an array.
4. To study and execute the Linear search method
5. To study and execute the Binary Search method
6. To study and execute the Pattern matching Algorithms (Slow and Fast)
7. To study and execute Bubble sort method.
8. To study and implement various operations on singly linked list
    a. Traversing the linked list.
    b. Insert a node at the front of the linked list.
    c. Delete a last node of the linked list.
    d. Searching a Linked list.
9. To study and implement following operations on the doubly linked list.
    a. Insert a node at the front of the linked list.
    b. Insert a node at the end of the linked list.
    c. Delete a last node of the linked list.
    d. Delete a node before specified position.
10. To study and implement following operations on the circular linked list.
    a. Insert a node at the end of the linked list.
    b. Insert a node before specified position.
    c. Delete a first node of the linked list.
    d. Delete a node after specified position.
11. Understand the stack structure and execute the push, pop operation on it.
12. Understand the Queue structure and execute the insertion, deletion operation on it.
13. Formulate and demonstrate Transforming Infix Expressions to Postfix Expression using Stack.
14. Formulate and demonstrate the Evaluation of Postfix Expression using Stack.
15. To study and execute Quick sort method.
16. Understand the Tree structure and implement the Pre-order, In-order, post-order traversing operations on it.
17. Understand the concept of Recursion and write a program to calculate factorial of a number using Recursion.
18. Understand the Heap sort and implement it on given data.
19. Understand the Insertion sort and implement it on given data.
20. Understand the Selection sort and implement it on given data.
21. To study and execute Merge sort method.
22. To study and execute Radix sort method.
23. Write a Program to implement the concept of BFS algorithm.
24. Write a Program to implement the concept of DFS algorithm.
    To study and execute Josephus problem.

******

3CS205MD / 3DS205MD / 3KE205MD / 3AD205MD / 3CSI205MD
FOUNDATIONS OF COMPUTING & PROGRAMMING

Type: Theory | **Credits:** 2 **Total Contact Hours:** 22–25 hours

Pre-requisite: Basic familiarity with mathematics and logical reasoning.

Course Objectives:

- To introduce fundamental concepts of computing and its relevance across engineering domains.
- To develop problem-solving skills using algorithmic approaches.
- To provide exposure to basic programming constructs using a high-level language (Python).
- To build foundational understanding required for advanced computational courses.

Course Outcomes (COs):

1. CO1: Understand computing systems and problem-solving logic. (Unit 1)
2. CO2: Apply algorithmic thinking to solve simple problems. (Unit 2)
3. CO3: Implement basic programs using control structures and I/O operations. (Unit 3)

Unit-wise Syllabus:

Unit 1: Basics of Computing & Programming (7 hrs)
Introduction to Computers and Programming Languages, Components of a Computer System and Software Types, Number Systems (Binary, Decimal, Hexadecimal) and Conversions, Basics of Algorithms and Flowcharts, Introduction to Python: Syntax, Variables, Data Types.

Unit 2: Control Structures and Functions (8 hrs)
Conditional Statements: if, if-else, nested if, Looping Constructs: for, while, Introduction to Functions and Modular Programming, Basic Math Operations and Built-in Functions in Python.

Unit 3: Input-Output and Basic Problem Solving (7 hrs)
Input from User and Output Formatting, Lists and String Handling, Simple Problem-Solving , Examples using Loops and Conditions, Writing and Executing Python Scripts.

CO-PO Mapping Table:

CO \ PO	PO1	PO2	PO4	PO5	PO10	PO12
CO1	3	2				1
CO2	2	3	2			2
CO3		3	2	3	2	3

Text Book(s): Reema Thareja, "Python Programming: Using Problem Solving Approach," Oxford University Press, 2017.

Reference Books:

1. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist," O'Reilly Media, 2nd Edition, 2015.
2. Zelle, John M., "Python Programming: An Introduction to Computer Science," Franklin, Beedle & Associates, 2017.
3. Anita Goel, "Computer Fundamentals," Pearson Education, 2010.
4. Charles Severance, "Python for Everybody: Exploring Data Using Python 3," CreateSpace, 2016.
5. Yashavant Kanetkar, "Let Us Python," BPB Publications, 2020.

****

3CS206OE1 / 3DS206OE1 / 3KE206OE1 / 3AD206OE1 / 3CSI206OE1

Basics of Computing for Engineers

Semester: III | **Type**: Theory | **Credits**: 3 **Total Contact Hours**: 45 Hours

Pre-requisites:

- Basic understanding of mathematical logic and problem-solving.
- Familiarity with basic operations on a computer (keyboard, file handling, browsing).
- General curiosity towards technology and engineering applications.

Course Objectives:

- To introduce the fundamentals of computing and its applications across engineering domains.
- To familiarize students with core computing concepts such as data, logic, programming, and storage.
- To provide hands-on understanding of basic programming logic and computational thinking.
- To help students develop problem-solving skills using algorithms and flowcharts.
- To relate computing applications with real-life engineering challenges and tools.

Course Outcomes (COs):
After completing this course, students will be able to:
1. Understand the role of computing in modern engineering practices.
2. Apply basic logical reasoning and computational thinking to solve simple engineering problems.
3. Interpret data and operations using flowcharts, pseudo-code, and basic algorithms.
4. Recognize various computing tools used in engineering industries.
5. Appreciate ethical and societal implications of computing.

Syllabus:

Unit 1: Introduction to Computers & Computing (8 Hours)
Evolution and Applications of Computers in Engineering, Components of a Computer System: Hardware & Software, Operating Systems Basics (Windows/Linux), Introduction to Data, Binary Systems, and Memory Hierarchy, Role of Computers in Engineering Domains (CAD, CAM, Simulation, etc.).

Unit 2: Fundamentals of Programming & Logic Building (8 Hours)
Understanding Algorithms and Flowcharts, Variables, Constants, Data Types, Input/Output, Arithmetic and Logical Operators, Control Structures: if-else, loops (for, while), Problem Solving with Simple Pseudocode Examples.

Unit 3: Introduction to Computational Thinking (8 Hours)
What is Computational Thinking?, Abstraction, Decomposition, Pattern Recognition, Algorithm Design, Case Studies of Computational Thinking in Civil, Mechanical, and Electrical Engineering
Real-World Examples using Block-Based Programming (e.g., Scratch or Blockly),

Unit 4: Data & File Management in Engineering (7 Hours)
Understanding Structured vs Unstructured Data, File Handling Basics (Creating, Reading, Writing Files), Data Representation (Tables, Graphs, Charts), Applications: Excel, Google Sheets, Data Logging in Labs.

Unit 5: Overview of Programming Tools & Environments (7 Hours)
Common Engineering Programming Tools (Python, MATLAB, Excel Macros), Introduction to Online IDEs, Simulators, and AutoCAD Programming, Cloud-Based Tools for Engineers (e.g., Google Colab, Replit), Software Tools in Design and Simulation.

Unit 6: Ethical and Professional Aspects of Computing (7 Hours)
Digital Citizenship and Responsible Use of Technology, Data Privacy, Cyber Ethics and Safety
Open Source Software and Licensing, Introduction to Engineering Software Lifecycle & Automation, Relevance of Computing in Sustainable Engineering Practices.

CO-PO Mapping Table:

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO10	PO12
CO1	3					2		2
CO2	2	3		2	3			2
CO3	2	3		3	3		2	
CO4		2			3		2	2
CO5						3	2	2

Text Book(s):
1. "Computing Fundamentals and Programming in C" – E. Balagurusamy, McGraw Hill Education, 2018
2. "Let Us Compute" – P. K. Sinha, BPB Publications, 2020

Reference Books:
1. "Computational Thinking for the Modern Problem Solver", David D. Riley and Kenny A. Hunt, CRC Press, 2014.
2. "Python Programming for the Absolute Beginner", Michael Dawson, Cengage Learning, 2010.
3. "Introduction to Computation and Programming Using Python", John Guttag, MIT Press, 2016.
4. "Digital Fundamentals", Thomas L. Floyd, Pearson Education, 11th Edition.
5. "Introduction to Computers", Peter Norton, McGraw Hill Education, 7th Edition.
6. "Computational Thinking: A Beginner's Guide to Problem Solving and Programming", Karl Beecher, BCS Learning, 2017.

******

3CS206OE2 / 3DS206OE2 / 3KE206OE2 / 3AD206OE2 / 3CSI206OE2
E-Commerce

Semester: III | **Type**: Theory | **Credits**: 3 **Total Contact Hours**: 45 Hours

Pre-requisites:
- Basic knowledge of the Internet and Computers.
- General awareness of business and commerce.

Course Objectives:
- To introduce students to E-Commerce and its applications in modern business.
- To help students understand different types of online businesses and digital transactions.
- To familiarize students with online payment systems and digital security.
- To explore how technology supports E-Commerce, including mobile commerce and digital marketing.
- To discuss ethical, legal, and security aspects of E-Commerce.
- To provide an overview of future trends and career opportunities in the E-Commerce industry.

Course Outcomes (COs):
After completing this course, students will be able to:
1. Explain the basic concepts of E-Commerce and its significance.
2. Identify different types of online business models and their applications.
3. Understand digital payment methods and their security aspects.
4. Analyze the impact of digital marketing and online customer engagement.
5. Recognize legal, ethical, and cybersecurity challenges in E-Commerce.
6. Explore career opportunities and emerging trends in E-Commerce.

Syllabus:

UNIT 1: Introduction to E-Commerce (7 Hours)
Definition and Scope of E-Commerce, Advantages and Challenges of E-Commerce, Traditional vs. Digital Business, Popular E-Commerce Platforms (Amazon, Flipkart, Alibaba, etc.), Impact of E-Commerce on Society and Business

UNIT 2: E-Commerce Business Models & Online Marketplaces (8 Hours)
Types of E-Commerce Businesses (B2B, B2C, C2C, D2C, etc.), How Online Marketplaces Work (Amazon, eBay, Meesho, etc.), Selling Products and Services Online, Role of Small Businesses & Startups in E-Commerce, E-Commerce in Different Sectors (Retail, Education, Healthcare, etc.)

UNIT 3: Digital Payments and Security (7 Hours)
Overview of Online Payment Methods (UPI, Debit/Credit Cards, Digital Wallets), How Payment Gateways Work (Google Pay, PayPal, Razorpay, etc.), Security Concerns in Online Transactions, Best Practices for Safe Online Shopping, Case Study: Digital Payment Success in India (UPI, RuPay, etc.)

UNIT 4: Digital Marketing & Customer Engagement (7 Hours)
What is Digital Marketing?, Role of Social Media in E-Commerce (Facebook, Instagram, LinkedIn, etc.), Basics of Search Engine Optimization (SEO) & Online Advertising, How Businesses Use Email & Influencer Marketing, Customer Feedback & Personalization in E-Commerce

UNIT 5: Legal, Ethical & Cybersecurity Issues in E-Commerce (8 Hours)
Consumer Rights in Online Shopping, E-Commerce Laws & Regulations (IT Act, Consumer Protection Laws, etc.), Cybersecurity Threats (Hacking, Phishing, Data Privacy Issues), Ethical Concerns in Online Business (Fake Reviews, Data Privacy), How to Protect Yourself While Shopping Online

UNIT 6: Emerging Trends & Career Opportunities in E-Commerce (8 Hours)
Growth of Mobile Commerce (M-Commerce), The Future of E-Commerce (AI, Augmented Reality, Metaverse), E-Commerce and Logistics (Online Delivery & Warehousing), Starting Your Own Online Business (Basic Steps & Challenges), Career Paths in E-Commerce (Marketing, Business, Technology Roles)

Text Book:
"E-Commerce: Fundamentals and Applications", *Henry Chan and Raymond Lee*, John Wiley & Sons, 2001

Reference Books:
1. “E-Commerce and Digital Business” , *D. G. Harkut, K. N. Kasat,* Notion Press, 2025.
2. "E-Commerce Essentials", *Kenneth C. Laudon, Carol Guercio Traver*, Pearson Education, 2014
3. "Electronic Commerce 2018: A Managerial and Social Networks Perspective", *Efraim Turban, Jon Outland, David King, Jae Lee, Ting-Peng Liang*, Springer, 2018
4. "Digital Marketing for Dummies", *Ryan Deiss, Russ Henneberry*, John Wiley & Sons, 2020
5. "Click Millionaires: Work Less, Live More with an Internet Business You Love", *Scott Fox*, AMACOM, 2012
6. "Cybersecurity and Cyberlaw", *Pavan Duggal*, Universal Law Publishing, 2019
7. "The E-Commerce Book: About a Channel that Became an Industry", *Alexander Graf, Holger Schneider*, Independently published, 2021

****

3CS207EM / 3KE207EM / 3DS207EM / 3AD207EM / 3CSI207EM
ENTREPRENEURSHIP DEVELOPMENT

Course Pre-requisites:
- Basic understanding of engineering problem-solving
- Interest in innovation or business development

Course Objectives: The course aims to:
- Introduce engineering students to the fundamentals of entrepreneurship.
- Develop the ability to identify and evaluate entrepreneurial opportunities.
- Encourage creativity and innovation as key entrepreneurial traits.
- Equip students with knowledge of business planning, models, and funding.
- Emphasize the role of marketing, intellectual property, and ethical practices.

Course Outcomes: At the end of the course, students will be able to:
1. Understand and explain key entrepreneurial concepts and the startup ecosystem.
2. Identify, assess, and evaluate business opportunities using feasibility analysis.
3. Develop a basic business plan including financial, marketing, and legal elements.
4. Recognize the importance of innovation, funding sources, and IP rights in startups.
5. Apply entrepreneurial thinking to engineering problems and real-world challenges.

Syllabus:

Unit I: Introduction to Entrepreneurship & Opportunity Identification (8 Hours)
Evolution and Importance of Entrepreneurship in the Engineering Context, Myths about Entrepreneurs, Role of Engineers as Entrepreneurs, Types of Startups, Entrepreneurial Traits, Ideation Techniques, Innovation vs. Invention, Identifying Opportunities and Gaps in the Market, Tools for Idea Generation (Brainstorming, SCAMPER, TRIZ).

Unit II: Feasibility Analysis, Business Models & Planning (7 Hours)
Feasibility Analysis: Product/Service, Market, Organizational & Financial, Business Model Canvas – Key Components, Elements of a Business Plan, Introduction to Financial Planning – Break-even, Basic Forecasting, Industry Analysis and Competitor Mapping, Value Proposition Design.

Unit III: Funding, Marketing & Intellectual Property (7 Hours)
Sources of Funding: Angel Investors, VCs, Bootstrapping, Government Schemes, Role of Incubators and Accelerators, Marketing for Startups – 4Ps, Digital Marketing Basics, Importance of IP: Patents, Copyrights, Trademarks, Trade Secrets, Ethics in Entrepreneurship, Role of Entrepreneurship in Sustainable Development.

Text Book:
Bruce R. Barringer & R. Duane Ireland, *Entrepreneurship: Successfully Launching New Ventures*, Pearson Education, 3rd Edition

Reference Books:
1. Poornima M.C., *Entrepreneurship Development – Small Business Enterprises*, Pearson
2. Ram Chandran, *Entrepreneurial Development*, Tata McGraw Hill
3. Arya Kumar, *Entrepreneurship*, Pearson
4. Khanka, S. S., *Entrepreneurial Development*, S. Chand & Co.
5. Kanishka Bedi, *Management and Entrepreneurship*, Oxford University Press
6. Badhai, B., *Entrepreneurship for Engineers*, Dhanpat Rai & Co.

****

SEMESTER - IV

4CS209PC/4DS209PC/4AD209PC/4KE209PC/4CSI209PC

DATA COMMUNICATION AND NETWORKING

Course Pre-requisite: Computer and Data Communication Requirements

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of Data Communication and Networking by being able to do each of the following:

- Study the basic taxonomy and terminology of the digital communication system & computer networking and enumerate the layers of OSI model and TCP/IP model.
- Acquire knowledge of Application layer paradigms and protocols.
- Study Transport layer design issues, Transport layer services, and protocols.
- Gain core knowledge of Network layer routing protocols and IP addressing.
- Study data link layer concepts, design issues, and protocols.

Course Outcomes (CO's): On completion of the course, the students will be able to:

1. Describe the functions of each layer in OSI and TCP/IP model.
2. Describe the Transport layer and Transport layer services.
3. Classify the routing protocols and analyze how to assign the IP addresses for the given network.
4. Explain the functions of Application layer and Presentation layer paradigms and Protocols.
5. Describe the functions of data link layer and explain the protocols.
6. Explain the types of transmission media with real time applications.

SYLLABUS:

Unit I: Introduction to Network Models (Hours: 07)
Data Communication Components: Basic network types, switching mechanisms, Internet standards. **Layered Architecture**: Overview of the OSI and TCP/IP models, focusing on key functions of each layer.

Unit II: (Hours: 07)
Transmission Media: Guided vs. Unguided media, key characteristics, and real-time application examples. **Switching Techniques**: Circuit, Packet, and Virtual Circuit Switching.

Unit III: Application Layer Protocols - (Hours: 07)
Application Layer Overview: Client-server model, APIs, P2P networking. **Key Protocols**: HTTP, FTP, SMTP, DNS, focusing on their importance in internet communication.

Unit IV: Transport Layer Fundamentals - (Hours: 07)
Transport Layer Overview: Principles of connectionless and connection-oriented services. **Key Protocols**: UDP, TCP – Concepts of reliability, congestion control, flow control. **Multiplexing and Demultiplexing**: Basic mechanisms and their relevance in data transmission.

Unit V: Network Layer Protocols and IP Addressing - (Hours: 07)
Network Layer Overview: Services provided by the network layer, datagram vs. virtual circuit approaches. **IP Addressing**: IPv4, IPv6, NAT, DHCP, and ICMP functionalities. **Routing**: Introduction to routing algorithms (Distance Vector and Link-State), basic concept of forwarding.

Unit VI: Data Link Layer and MAC Protocols - (Hours: 07)
Data Link Layer Services: Framing, error detection (CRC, Checksum), and correction techniques. **MAC Layer**: Concepts of LAN addressing, ARP, CSMA/CD, and PPP. **Link Layer Protocols**: HDLC and Point-to-Point Protocol.

Text Books:
1. Behrouz A. Forouzan: Data Communication and Networking, (5/e) (TMH)
2. James F. Kurose & K W Ross: Computer Networking, Pearson Education (LPE)

Reference Books:
1. William Stallings: Data & Computer Communications, 6/e, Pearson Education
2. William L. Schweber: Data Communication, McGraw Hill
3. Douglas E. Comer: Computer Network & Internet, Addison Wesley.
4. Andrew S. Tanenbaum: Computer Networks, PHI (5E)
5. Leon Garcia & Widjaja: Communication Networks, TMH

****

4CS210PC/4DS210PC/4AD210PC/4KE210PC/ 4CSI210PC

OPERATING SYSTEM

Course Pre-requisite: Discrete Structures, Data Structure, Any programming Language

Course Objectives:
1. To make students aware of the kernel and shell structure of the operating systems.
2. To make students aware of the purpose, structure and functions of operating systems
3. To equip students with understanding of the various scheduling algorithms in OS.
4. To make students aware of understanding of memory management in different OS.

Course Outcomes: (Expected Outcomes):
On completion of the course, the students will be able to:
- Explain memory management issues like external fragmentation, internal fragmentation.
- Illustrate multithreading and its significance.
- List various protection and security mechanisms of OS.
- Analyze and solve the scheduling algorithms.
- Analyze the deadlock situation and resolve it.
- Compare various types of operating systems

SYLLABUS:

Unit I: Introduction to OS (Hours: 07)
Introduction: Operating System definition, OS Evolution, Components and Services, Process Concept, Process Scheduling, Operations on Processes, Cooperating Processes, Interprocess Communication, Threads Overview, Multithreading Models, Threading Issues, Java Threads

Unit II: Process Scheduling (Hours: 07)
Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR, Priority, Multilevel Queue, Multilevel Feedback Queue Scheduling

Unit III: Process Synchronization (Hours: 07)
Process Synchronization Basics: The Critical-Section Problem, Synchronization Hardware, Semaphores, Monitors, Deadlocks: Definition & Characterization, Deadlocks Prevention, Avoidance, Detection and Recovery from Deadlock

Unit IV: Memory Management (Hours: 07)
Memory Management Background, Swapping, Contiguous Memory Allocation Schemes, Paging, Segmentation, Virtual Memory Management: Background, Demand paging scheme, Process Creation, Page Replacement Policies, Allocation of Frames, Thrashing.

Unit V: File System (Hours: 07)
File-System Interface; Directory Structure, File-System Mounting, File Sharing &Protection, File- System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free- Space Management.

Unit VI: I/O System (Hours:07)
I/O Systems: Overview, I/O Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O to Hardware Operations , Disk Scheduling, Disk Management, Swap-Space Management, RAID Structure.

Text Book: Avi Silberschatz, P.B.Galvin, G.Gagne: “Operating System Concepts” (9/e) John-Wiley & Sons.

Reference Books:
1. A.S.Tanenbaum “Modern Operating Systems” Pearson Education.
2. William Stallings “Operating Systems” Prentice-Hall.
3. D. M. Dhamdhere “Operating Systems” Tata McGraw-Hill.
4. P. Balkrishna Prasad: “Operating Systems” Scitech Publications (I) Pvt. Ltd.

****

4CS211PC/4KE211PC/4CSI211PC

THEORY OF COMPUTATION

Course Pre-requisites:
- Discrete Mathematics
- Data Structures

Course Objectives:
- To understand automata theory and its operation.
- To understand mathematical expressions for formal languages.
- To learn implementing the state machines for solving real life problems.
- To study computing machines and comparing different types of computational models.
- To understand the fundamentals of problem decidability and Un-Decidability.

Course Outcomes: (Expected Outcome):
On completion of the course, the students will be able to:
1. To construct finite state machines to solve problems in computing.
2. To write regular expressions for the formal languages.
3. To construct and apply well defined rules for parsing techniques in compiler.
4. To construct and analyze Push Down, Turing Machine for formal languages
5. To express the understanding of the Chomsky Hierarchy.
6. To express the understanding of the decidability and un-decidability problems.

SYLLABUS:

Unit I: Finite State Machines: (Hours: 07)
Alphabet, String, Formal and Natural Language, Operations, Definition and Design DFA (Deterministic Finite Automata), NFA (Non-Deterministic Finite Automata), Equivalence of NFA and DFA: Conversion of NFA into DFA, Conversion of NFA with epsilon moves to NFA, Minimization Of DFA, Definition and Construction of Moore and Mealy Machines, Inter conversion between Moore and Mealy Machines. Minimization of Finite Automata. (Construction of Minimum Automaton).

Unit II: Regular Expression and Regular Grammar: (Hours: 07)
Definition and Identities of Regular Expressions, Construction of Regular Expression of the given Language, Construction of Language from the RE, Conversion of FA to RE using Arden's Theorem, Inter-conversion RE to FA, Pumping Lemma for RL, Closure properties of RLs (proofs not required), Regular grammar, Equivalence of RG (RLG and LLG) and FA.

Unit III: Context Free Grammar and Languages: (Hours: 07)
Introduction, Formal Definition of Grammar, Notations, Derivation Process: Leftmost Derivation, Rightmost Derivation, Derivation Trees, Construction of Context-Free Grammars and Languages, Pumping Lemma for CFL, Simplification of CFG, Normal Forms (CNF and GNF), Chomsky Hierarchy.

Unit IV: Pushdown Automata: (Hours: 07)
Introduction and Definition of PDA, Construction of PDA, Acceptance of CFL, Equivalence of CFL and PDA: Inter-conversion, Introduction of DCFL and DPDA, Enumeration of properties of CFL, Context Sensitive Language, Linear Bounded Automata.

Unit V: Turing Machines: (Hours: 07)
Formal definition of a Turing Machine, Design of TM, Computable Functions, Church's hypothesis, Counter machine, Variants of Turing Machines: Multi-tape Turing machines, Universal Turing Machine.

Unit VI: Decidability and Un-Decidability: (Hours: 07)
Decidability of Problems, Halting Problem of TM, Un-Decidability: Recursive enumerable language, Properties of recursive & non-recursive enumerable languages, Post Correspondence Problem, Introduction to Recursive Function Theory.

Text Books:
1. Hopcraft H.E. & Ullman J: Introduction to Automata Theory, Languages and Computation
2. Peter Linz: An Introduction to Formal Languages and Automata

Reference Books:
1. Rajesh K. Shukla: Theory of Computation, CENGAGE Learning, 2009.
2. K V N Sunitha and N Kalyani: Formal Languages and Automata Theory, McGraw Hill,2010
3. Lewis H.P. and Papadimition C.H.: Elements of Theory of Computation
4. Mishra &Chandrashekharan: Theory of Computation
5. C. K. Nagpal: Formal Languages and Automata Theory, Oxford University Press, 2011.
6. Vivek Kulkarni : Theory of Computation, OUP India, 2013.

****

4CS212PC/4DS212PC/4AD212PC/4KE212PC/4CSI212PC

DATA COMMUNICATION & NETWORKING- LAB.

Course Pre-requisite: Computer and Data Communication Requirements

Course Objectives:
- To understand the working principle of various communication protocols
- To understand and analyze the signal flow in a digital communication system.
- To analyze error performance of a digital communication system in presence of noise and other interferences.
- To evaluate the errors using various error detection & correction techniques.
- To understand network based protocols in data communication and networking.

Course Outcomes : On completion of the course, the students will be able to-
1. Analyze performance of various communication protocols
2. Implement Configure various network protocols.
3. Compare IP Address classes of networks

List of Experiments:
This is a sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi

1. To study various LAN topologies and their creation using network devices, cables and computers.
2. To connect the computers in Local Area Network.
3. Familiarization with Networking Components and devices: LAN Adapters, Hubs, Switches, Routers etc.
4. Write a program of bit stuffing used by Data Link Layer
5. Write a program to implement CRC(Cyclic Redundancy Check)
6. Write a program to implement Checksum
7. Write a program to implement Sliding window
8. Configure Internet connection and use IP-Config, PING / Tracer and Net stat utilities to debug the network issues.
9. Configuration of TCP/IP Protocols in Windows and Linux.

10. Transfer files between systems in LAN using FTP Configuration, install Print server in a LA and share the printer in a network.
11. Write a C Program to determine if the IP Address is in Class A, B, C, D, or E
12. Write a C Program to translate Dotted Decimal IP Address into 32 Bit Address.
13. Configure Host IP, Subnet Mask and Default Gateway in a System in LAN(TCP/IP Configuration)

****

4CS213PC/4DS213PC/4AD213PC/4KE213PC/4CSI213PC

OPERATING SYSTEM - LAB.

Course Pre-requisite: Basic computer programming

Course Objectives:

- To make students aware of the kernel and shell structure of the operating systems.
- To make students aware of the purpose, structure and functions of operating systems
- To equip students with understanding of the various scheduling algorithms in OS.
- To make students aware of understanding of memory management in different OS.

Course Outcomes: On completion of the course, the students will be able to-

1. Explain memory management issues like external fragmentation, internal fragmentation.
2. Illustrate multithreading and its significance.
3. List various protection and security mechanisms of OS.
4. Analyze and solve the scheduling algorithms.
5. Analyze the deadlock situation and resolve it.
6. Compare various types of operating systems

LIST OF EXPERIMENTS:

This is a sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi

1. To study Linux Operating System along with its installation.
2. To Study and Execute basic file commands and process related open source Ubuntu commands
    a. Commands to view all executing, block and suspended process.
    b. Command to check and change the priority of process CPU utilization for executing processes.
    c. Commands to check for child process, sub-processes, process tree, abort & end process and all other basics commands related to processes
3. Write a program for multithreading using C.
4. To simulate First Come First Serve & Shortest Job First process scheduling algorithm
5. To simulate Shortest Job First process scheduling algorithm
6. To simulate Pre-emptive Shortest Job First process scheduling algorithm
7. To implement Round Robin Process scheduling Algorithm
8. To implement Priority Based Process scheduling Algorithm
9. To implement and analyze multi-level queue scheduling algorithm
10. To implement the following file allocation strategies.
11. To simulate paging technique of memory management.
12. To implement the FIFO page replacement policy
13. To implement the LRU page replacement policy
14. To implement the optimal page replacement policy
15. To simulate producer-consumer problem using semaphores.
16. To implement Dining-Philosophers problem to deal with concurrency control mechanism.
17. To implement contiguous memory allocation strategies to detect fragmentation using: First Fit, Best Fit and Worst Fit.
18. To implement FCFS Disk Scheduling algorithm
19. To implement SCAN Disk Scheduling algorithm
20. To implement C-SCAN Disk Scheduling algorithm
21. To simulate Banker’s algorithm for deadlock avoidance
22. To implement following memory management techniques
23. Implement MVT and MFT where memory block size is 100 for 5 processes. Enter no. of blocks for each process and calculate internal fragmentation.
24. To simulate LFU page replacement algorithms
25. To simulate the Single level directory file organization techniques.
26. To Simulate bankers algorithm for Dead Lock Avoidance (Banker‘s Algorithm)

****

4CS214MD / 4CSI214MD

Data Structure and Problem Solving

Type: Theory | **Credits**: 2 **Total Contact Hours**: 22–23 hours

Pre-requisite: Basic understanding of programming (preferably in C or Python).

Course Objectives:

- To introduce fundamental data structures and their applications.
- To develop problem-solving skills through efficient use of data structures.
- To improve algorithmic thinking for common computational tasks.
- To expose students to practical scenarios where data structures are applied in engineering contexts.

Course Outcomes (COs):

1. CO1: Understand and differentiate various data structures and their use cases. (Unit 1)
2. CO2: Apply linear and non-linear data structures in solving engineering problems. (Unit 2)
3. CO3: Analyze algorithm performance and implement solutions using appropriate structures. (Unit 3)

Unit-wise Syllabus:

Unit 1: Introduction to Data Structures (7 hrs)
Concept and Classification of Data Structures, Abstract Data Types (ADT), Arrays, Strings, and Linked Lists, Applications: Polynomial Representation, Sparse Matrices.

Unit 2: Linear and Non-linear Structures (8 hrs)
Stacks and Queues: Operations and Applications (Infix to Postfix, Recursion, Job Scheduling)
Trees: Binary Trees, Tree Traversals, Binary Search Trees, Applications in Expression Trees, Directory Structures.

Unit 3: Searching, Sorting & Hashing (7 hrs)
Searching Techniques: Linear and Binary Search, Sorting Algorithms: Bubble, Selection, Insertion, Merge Sort, Hashing and Collision Resolution Techniques, Time and Space Complexity Basics.

CO-PO Mapping Table:

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO12
CO1	3	2				
CO2	2	3	3	2	2	2
CO3	2	3	3	3	3	2

Text Book(s):

1. Seymour Lipschutz, "Data Structures", Schaum's Outline Series, McGraw Hill Education.
2. Reema Thareja, "Data Structures Using C", Oxford University Press, 2nd Edition, 2014.

Reference Books:

1. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, "Fundamentals of Data Structures in C", Universities Press, 2008.
2. Jean-Paul Tremblay & Paul G. Sorenson, "An Introduction to Data Structures with Applications", Tata McGraw-Hill.
3. Narasimha Karumanchi, "Data Structures and Algorithms Made Easy", CareerMonk Publications.
4. Robert Lafore, "Data Structures and Algorithms in C++", Pearson Education.
5. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Pearson Education.
6. Thomas H. Cormen et al., "Introduction to Algorithms", MIT Press (for higher interest).

****

4CS215VS/4DS215VS/4AD215VS/4KE215VS/4CSI215VS

Computing Skill #1 LAB. (VSEC-III)

(Based on technology like -Python/Django etc. to be decided by Individual Dept. of respective College)

Course Prerequisite: Basic knowledge of any Programming Language

Course Objectives:

- To be able to program design with functions using Python.
- To understand data and information processing techniques.
- To understand to Design a program to solve the problems.
- To be able to access database using python programming.
- To be able to design web applications using python programming.

Course Outcomes : On completion of the course, the students will be able to-

1. Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python
2. Interpret different Decision Making statements, Functions, Object oriented programming in Python
3. Summarize different File handling operations
4. Explain how to design GUI Applications in Python and evaluate different database operations
5. Develop applications using Django framework or Flask

LIST OF EXPERIMENTS:

This is a sample list of Experiments, **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi

1. Write python program to store data in list and then try to print them.
2. Write python program to print list of numbers using range and for loop
3. Write python program to store strings in list and then print them.
4. Write python program in which an function is defined and calling that function prints Hello World.
5. Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST
6. 2017”
7. Write a program to create, append, and remove lists in python.
8. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
9. Write a program to demonstrate working with tuples in python.
10. Write a program to demonstrate working with dictionaries in python.
11. Write a python program to find largest of three numbers.
12. Write python program in which an function(with single string parameter) is defined and calling that function prints the string parameters given to function.
13. Write python program in which an class is define, then create object of that class and call simple print function define in class.
14. Write a Python script that prints prime numbers less than 20.
15. Write a python program to find factorial of a number using Recursion.
16. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
17. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
18. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
19. Write a Python class to convert an integer to a roman numeral.
20. Write a Python class to implement pow(x, n)
21. Write a Python class to reverse a string word by word.
22. Accessing and working with databases using Python.
23. Create data frame from .csv files and operations on it.
24. Plotting various graphs using Python.
25. Developing basic GUI using Python.
26. Developing web applications using Django framework or Flask

Reference Books :

1. “Core Python Programming”, R. NageswaraRao, dreamtech press.
2. “Python Programming A Modular Approach With Graphics, Database, Mobile and WebApplications”, SheetalTaneja, Naveen Kumar, Pearson.
3. Python Web Development with Django By Jeff Forcier, Paul Bissex, Wesley J Chun, Addison-Wesley Professional.
4. Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning
5. Allen B. Downey , “ Think Python: How to Think Like a Computer Scientist”, Second Edition,
6. Shroff/O’Reilly Publishers
7. John V Guttag. “Introduction to Computation and Programming Using Python”, Prentice Hall of India
8. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, “Data Structures and Algorithms in Python”, Wiley
9. Introduction to Computation and Programming using Python, by John Guttag, PHI Publisher, Revised and Expanded version (Referred by MIT)

******

4CS216OE1 / 4DS216OE1 / 4KE216OE1 / 4AD216OE1 / 4CSI216OE1

Intellectual Property Rights

Semester IV | Type: Theory | **Credits**: 2 **Total Contact Hours**: 22–25 Hours

Course Pre-requisite:
Basic understanding of communication skills, general ethics, and awareness of creative work (projects, research, design, coding, etc.)

Course Objectives:
- To provide students with foundational knowledge of Intellectual Property Rights (IPR) and their significance in innovation and research.
- To introduce different forms of intellectual properties such as patents, copyrights, trademarks, and designs.
- To create awareness about the legal and ethical aspects of protecting intellectual creations and innovations in engineering and beyond.

Course Outcomes:
On successful completion of the course, students will be able to:
1. Explain the significance of intellectual property in research, development, and innovation.
2. Identify various types of intellectual property and understand their application in different domains.
3. Demonstrate basic knowledge of procedures involved in protecting intellectual property and the roles of regulatory bodies.
4. Relate IPR knowledge to their project work, industrial applications, and entrepreneurship.

SYLLABUS:

Unit I: Introduction to Intellectual Property Rights (8 Hours)
Creativity, Innovation, and their Role in Development, What is Intellectual Property?, Types of Intellectual Property: Patents, Copyrights, Trademarks, Trade Secrets, Industrial Designs, and Geographical Indications, Importance of IPR in Engineering, Research, Startups, and Academia, Examples and Case Studies from various engineering disciplines.

Unit II: Patent, Copyright, and Design Protection (7 Hours)
Patent: What can be patented? Basic criteria (novelty, utility, non-obviousness), Simple steps of Patent Filing in India, Use of patents in project work or research, **Copyright**: For literature, music, software codes, and digital content, **Industrial Design**: Protection of new designs and their aesthetic value, Ownership, Licensing, and Rights of Creators.

Unit III: Trademarks, Geographical Indications & IPR in the Digital Era (7 Hours)
Trademarks: Brand names, logos, slogans – importance in business and marketing, **Geographical Indications**: Indian examples (e.g., Darjeeling Tea, Nagpur Orange), Role of IPR in Startups and Product Commercialization, Government Initiatives: National IPR Policy 2016, Startup India, Atmanirbhar Bharat, Career Opportunities in IPR, Case Studies and Recent Trends.

Text Books:
1. K. V. Nithyananda (2019), “Intellectual Property Rights: Protection and Management”, Cengage Learning India.
2. P. Neeraj and D. Khusdeep (2014), “Intellectual Property Rights”, PHI Learning.

Reference Books:
1. Deborah E. Bouchoux, “Intellectual Property for Paralegals”, 4th Edition, Cengage Learning, 2012.
2. M. M. S. Karki, “Intellectual Property Rights: Basic Concepts”, Atlantic Publishers, 2009.
3. V. K. Ahuja, “Law relating to Intellectual Property Rights”, Lexis Nexis, 2017.
4. Prabuddha Ganguli, “Intellectual Property Rights – Unleashing the Knowledge Economy”, Tata McGraw Hill, 2001.

****

4CS216OE2 / 4DS216OE2 / 4KE216OE2 / 4AD216OE2 / 4CSI216OE2

INFORMATION SYSTEMS FOR ENGINEERS

Semester IV | Type: Theory | **Credits**: 2 **Total Contact Hours**: 22–25 Hours

Pre-requisites:
- Basic understanding of engineering processes and workplace practices.
- Familiarity with computer usage and basic productivity tools (Word, Excel, etc.).

Course Objectives:
1. To introduce the fundamental concepts of Information Systems (IS) and their relevance in engineering and business domains.
2. To explore how engineers can use IS for decision-making, automation, and efficiency.
3. To familiarize students with various types of information systems including MIS, ERP, and industrial systems.
4. To understand the challenges, risks, and future trends associated with the implementation of MIS.

Course Outcomes (COs):
After successful completion of the course, students will be able to:
1. Understand the basic structure and components of Information Systems used in engineering.
2. Identify and evaluate applications of Information Systems across different engineering disciplines.
3. Demonstrate understanding of ERP, MIS, and database systems and their integration with engineering workflows.

Unit-wise Syllabus:

Unit 1: Basics of Information Systems in Engineering (7–8 Hours)
Introduction to Information Systems: Definition, Need, and Importance, Components of IS: Hardware, Software, Data, People, Processes, Types of IS: Transaction Processing Systems (TPS), Management Information Systems (MIS), Decision Support Systems (DSS), Role of IS in different Engineering Domains (Manufacturing, Civil, Electrical, etc.).

Unit 2: Enterprise Systems and Engineering Applications (7–8 Hours)
Overview of Enterprise Resource Planning (ERP) Systems, Functional Modules in ERP: Finance, Production, Inventory, HR, Industrial Automation Systems & SCADA, Role of IS in Supply Chain, Maintenance, Quality, and Smart Manufacturing (Industry 4.0).

Unit 3: Emerging Trends and Case Studies in IS (7–8 Hours)
Cloud-Based Information Systems, Role of IoT and Data Analytics in Engineering IS, Engineering Workflow Automation with IS Tools (Examples: AutoCAD Vault, PLM, BIM), Case Studies: Use of IS in Smart Cities, Energy Management, Water Resource Planning, Challenges in IS Implementation: Data Security, Change Management, Cost.

CO–PO Mapping Table:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO12
CO1	3	2			2			2
CO2	2	3	2		2		2	2
CO3		2	3	2	3	2		3

Text Books:
1. Laudon & Laudon, *Management Information Systems: Managing the Digital Firm*, Pearson Education, 2021
2. James A. O'Brien, *Introduction to Information Systems*, McGraw-Hill, 2020

Reference Books:
1. Rainer & Prince, *Introduction to Information Systems*, Wiley India, 2019
2. Turban et al., *Information Technology for Management*, Wiley, 2021
3. Monk & Wagner, *Concepts in Enterprise Resource Planning*, Cengage Learning, 2016
4. Mahadeo Jaiswal & M. Mital, *Management Information Systems*, Oxford University Press, 2015
5. Rajesh Ray, *Enterprise Resource Planning*, Tata McGraw Hill, 2011
6. Ghosh, *Management Information Systems*, PHI Learning, 2014

****

[4ME217EM / 4ET217EM / 4CS217EM / 4KE217EM / 4AD217EM / 4DS217EM / 4EP217EM / 4EE217EM / 4CH217EM / 4CT218EM]

PRINCIPLES OF ECONOMICS & MANAGEMENT / ENGINEERING ECONOMICS / SOCIAL SCIENCE & ENGINEERING ECONOMICS

Lecture: 02 Hrs/w
University Exam: 30
Credits: 02
Theory Paper (Hrs.): 02:00 Hrs
Tutorial: 0
Internal Assessments: 20
Teaching Scheme: 20 Hours

Course Objectives:
To provide engineering students from diverse disciplines with basic economic reasoning, cost awareness, and entrepreneurial
orientation, enabling them to understand business, startup, and societal applications of engineering knowledge.

Course Outcomes (Cos):
After completing this course, the students will be able to:
- Understand basic economic concepts relevant across disciplines
- Apply cost and demand concepts in real-life engineering and business situations
- Identify entrepreneurial opportunities and prepare a basic project idea

Syllabus:

UNIT–I: Basics of Engineering Economics: [Marks: 10] (7 Hours)
Meaning and Scope of Engineering Economics. Role of Economics in Multidisciplinary Engineering Decisions. Basic Economic Concepts: Scarcity, Choice, Opportunity Cost, Utility and Marginal Utility. Demand Analysis: Meaning and Types of Demand, Determinants of Demand, Elasticity of Demand. Demand Forecasting: Purpose and Importance, Methods of forecasting.

UNIT–II: Cost Concepts and Break-even Analysis: [Marks: 10] (7 Hours)
Factors of Production and Production Function. Cost Concepts: Fixed and Variable Cost, Average and Marginal Cost. Break-even Analysis: Concept, Assumptions and Significance, Break-even Chart. Depreciation: Meaning and Need, Straight Line and Written Down Value Method, Basic Idea of Taxation for Engineers.

UNIT–III: Entrepreneurship and Project Awareness: [Marks: 10] (6 Hours)
Concept and Importance of Entrepreneurship in Engineering, Entrepreneurial Traits and Competencies. Types of Enterprises: Manufacturing, Service, Startup, MSME. Government Support System: Role of District Industries Centre (DIC), Banks and Financial Institutions. Product / Service Selection: Basic Factors. Introduction to Project Report: Market Feasibility, Technical Feasibility, Financial Feasibility Overview. Concept of Working Capital.

Reference Books:

1. Managerial Economics by D. Salvatore, Tata McGraw Hill.
2. Entrepreneurship Development by C. B. Gupta & P. Srinivasan.
3. MSME & Startup India official publications.
4. Management Accounting by Khan M. Y., Jain P. K., Tata McGraw Hill, 1995.
5. Environmental Engineering and Management by Suresh K. Dhamija, S. K. Kataria and Sons, New Delhi.
6. Entrepreneurship: New Venture Creation – David H. Holt.
7. Handbook of Small Scale Industry by P. M. Bhandari.
8. Managerial Economics, V. L .Mote McGraw-Hill

****