Program Structure and Syllabus of B. Tech II Year

Computer Science and Engineering

(Academic Regulations - R24)

Venkatapur (V), Ghatkesar (M),
Medchal-Malkajgiri (Dt.), Hyderabad, Telangana, INDIA
deancse@anurag.edu.in; http://anurag.edu.in

B. TECH II YEAR I SEMESTER

S.No	Course Code	Category	Course	Hours per week			Credits
				L	T	P	
1	EMA2X17	Theoretical (PC)	Computer Organization and Architecture	3	0	0	3
2	EMA2X02	Theoretical (PC)	Discrete Mathematics	3	0	0	3
3	EMA2120	Theoretical & Practical (PC)	Advanced Data Structures	3	0	2	4
4	EMA2X19	Theoretical & Practical (PC)	Programming in Java	3	0	2	4
5	EMD2X04	Theoretical (PC)	Foundations of Software Engineering	3	0	0	3
6	EVA2X25	Practical (HS)	Dynamics of Group Discussion	0	0	2	1
7	ESE2X23	Practical (BS)	Quantitative Aptitude and Logical Reasoning - I	0	0	2	1
8	EVA2101	Exploratory (PC)	Integrated Project - I	0	0	2	1
			TOTAL	**15**	**0**	**10**	**20**

B. TECH II YEAR II SEM (4th semester)

S.No	Course Code	Category	Course	Hours per week			Credits
				L	T	P	
1	EMD2X01	Theoretical (BS)	Computer Oriented Statistical Methods	3	0	0	3
2	EMA2221	Theoretical (PC)	Fundamentals of Computer Algorithms	2	0	0	2
3	EMA2217	Theoretical & Practical (PC)	Database Management Systems	3	0	2	4
4	EMA2X18	Theoretical & Practical (PC)	Programming in Python	3	0	2	4
5	EMA2219	Theoretical & Practical (PC)	Web Technologies	3	0	2	4
6	EAE2225	Practical (HS)	Advanced Reading Comprehension Skills	0	0	2	1
7	ESE2X24	Practical (BS)	Quantitative Aptitude and Logical Reasoning - II	0	0	2	1
8	EVA2201	Exploratory (PC)	Integrated Project – II	0	0	2	1
9	--	IN	Summer Internship-I*	0	0	0	0
			TOTAL	**14**	**0**	**12**	**20**

***Evaluation is done in 3rd year 1st semester**

B. TECH III YEAR I SEM (5th Semester)

S.No	Course Code	Category	Course	Hours per week			Credits
				L	T	P	
1		Theoretical (BS)	Computer Networks	3	0	0	3
2		Theoretical & Practical (PC)	Operating Systems	3	0	2	4
3		Theoretical & Practical (PC)	Artificial Intelligence and Machine Learning	3	0	2	4
4		Theoretical & Practical (PC)	Advanced Algorithms	2	0	0	2
5		Theoretical & Practical (PC)	OE-1	3	0	0	3
6		Practical (HS)	Quantitative Aptitude and Reasoning - III	0	0	2	1
7		Practical (BS)	Hackathon/codeathon/industry certification/MOOCS	0	0	0	1
8		IN	Summer Internship-I	0	0	0	2
TOTAL				**14**	**1**	**10**	**20**

B. TECH III YEAR II SEM (6th Semester)

S.No	Course Code	Category	Course	Hours per week			Credits
				L	T	P	
1		PC	Mobile Application Development	3	0	2	4
2		PC	Big Data Analytics	3	0	2	4
3		PC	Internet of Things	3	0	2	4
4		PC	Fundamentals of Formal Languages and Compiler Design	3	0	0	3
5		PE	PE-1	3	0	0	3
6		Proj	Mini Project	0	0	4	2
7		PC	Summer Internship-II*	0	0	0	0
TOTAL				**14**	**1**	**10**	**20**

*** Evaluation is done in 4th year 1st semester**

B. TECH IV YEAR I SEM (7th Semester)

S.No	Course Code	Category	Course	Hours per week			Credits
				L	T	P	
1		PC	Cyber Security	3	0	2	4
2		PC	PE-II	3	0	0	3
3		PC	PE-III	3	0	0	3
4		PC	PE-IV	0	0	4	2
5		PE	Cloud Computing	3	0	0	3
6		Proj	Managerial Accounts and Financial Management	3	0	0	3
7		PC	Summer Internship-II	0	0	0	2
			TOTAL	**15**	**0**	**6**	**20**

B. TECH IV YEAR II SEM (8th Semester)

S.No	Course Code	Category	Course	Hours per week			Credits
				L	T	P	
1		OE	OE-2	3	0	0	3
2		OE	OE-3	3	0	0	3
3		PROJ	Technical Seminar	0	0	4	2
4		PROJ	Project	0	0	24	12
			TOTAL	**6**	**0**	**28**	**20**

Professional Electives:1-3

- Deep Learning
- Natural Language Processing
- Data Mining and Warehousing
- Data Visualization
- Ethical Hacking
- Cryptography and Network Security
- software testing
- Blockchain Technology
- Quantum Computing
- Augmented Reality (AR) and Virtual Reality (VR)
- Edge Computing
- Autonomous Systems

Professional Elective:4

- Cloud Computing Tools
- Data Mining and Warehousing Tools Data Visualization Tools
- AI Tools

OE-I : Management and Entrepreneurship

- Business Communication
- Financial Management for Engineers
- Marketing Management
- Sustainable Development
- Renewable Energy Sources
- Climate Change and Its Impact
- Environmental Laws and Policies

OE-II Foreign Languages:

- French
- German
- Spanish
- Japanese
- Mandarin Chinese

OE-III Interdisciplinary Electives:

- introduction to Bioinformatics
- Computational Biology
- Biomedical Data Analysis
- Health Informatics

Computer Organization and Architecture

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
		L	T	P	C	CIE	SEE	Total
EMA2X17	Theoretical (PC)	3	0	0	3	50	50	100

Course Outcomes

1. Apply number system conversions and complement methods to represent, manipulate, and perform arithmetic operations, including handling overflow conditions. — L3
2. Design combinational and sequential circuits by constructing truth tables and applying Boolean algebra principles to meet specific functional requirements — L6
3. Identify the instruction format of a given CPU Instruction. – L2
4. Calculate the physical memory address and explain the basic architecture of Microprocessor. –L3
5. Classify various types of memories and data transfer modes between CPU & I/O devices. – L4

UNIT I

Number Systems: Binary, Octal, Hex Decimal, and Conversions, Range of the number system; Representations of negative numbers using 1's and 2's complement; Binary additions and subtractions (using 1c, and 2c); Error correction & Detecting codes: Even parity, Odd parity. Hamming codes.

UNIT II

Boolean Algebra and Digital Logic Gates, Basic Boolean laws and minimization of the Boolean expression and advantages. Definition of Combinational and Sequential circuits, Logical view of Main Memory, CPU.

UNIT III

Instruction: Logical view of Main Memory**,** Instruction Definition, Instruction cycle, instruction storage, types of instruction formats (Zero, one, two and three address).

Addressing modes: Implied, Immediate, Register addressing mode, Direct addressing mode, Indirect addressing mode, Auto increment addressing mode, Indexed addressing mode. Numerical examples and problems.

UNIT IV

CPU-Organization: 8086 –CPU –Block diagram, Introduction to Minimum and Maximum mode, General purpose registers; segment register and generation of 20 bits address, segmentation of main memory, systems bus.

UNIT V

Memory Hierarchy, Main memory, Cache memory, hit and miss ratio, Introduction to Virtual Memory and their advantages, Bootstrap program, Multiprogramming.

I/O interface: Need of I/O interface (introduction), Example of programmed I/O, interrupt-initiated I/O. Daisy-Chaining priority. DMA block diagram, Cycle stealing, Introduction to IOP, Interrupt Service routine

Text Books

1. M.Morris Mano, Computer System Architecture,Revised Third Edition, Pearson/PHI, 2017.
2. Advanced Microprocessors and Peripherals by K. M. Bhurchandi A K Ray McGraw Hill Education (India) Pvt Ltd. 3e.

Reference Books

1. William Stallings, Computer Organization and Architecture, 6th Edition, Pearson/PHI, 2007.
2. Andrew S. Tanenbaum, Structured Computer Organization, 4th Edition, PHI/Pearson.

Online Resources

1. https://nptel.ac.in/courses/106105163

Discrete Mathematics

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
EMA2X02	Theoretical (PC)	L	T	P	C	CIE	SEE	Total
		3	0	0	3	50	50	100

Course Outcomes

After completion of course students will be able to

1. Analyze Statement Logic and Predicate Logic.(L4)
2. Apply the principles of Permutations and Combinations with repetition & without repetitions(L3)
3. Solve Recurrence Relations by using generating functions(L3)
4. Apply the knowledge of Relations and Graph Theory in the field of Computer Science.(L3)
5. Analyze the Algebraic Systems with their properties(L4)

UNIT I

Foundations: Basics, Sets and Operations of Sets, Fundamentals of Logic, Logical Inferences, First order logic and other methods of Proof, Rules of Inference for Quantified Propositions.

(Problems Only and Theorems without Proofs)

UNIT II

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumerating Combinations and Permutations with & without repetitions, constrained repetitions, and Principle of Inclusion and Exclusion. **(Problems Only and Theorems without Proofs)**

UNIT III

Recurrence Relations: Generating Functions, calculating coefficient of Generating Function, Solving Recurrence relations by substitution method and Generating Functions, The Method of Characteristic Roots, Solutions to inhomogeneous recurrence relations. **(Problems Only and Theorems without Proofs**)

UNIT IV

Relations and Lattices: Relations adjacency matrices and Directed Graphs, Operations on Relations, Special Properties of Binary Relations, Equivalence Relations, Ordering Relations, Lattices. **(Problems Only and Theorems without Proofs)**

UNIT V

Algebraic structures: Algebraic systems, examples and general properties, semi groups and monoids, groups, sub groups, homomorphism, isomorphism, Permutation groups and cyclic permutations. **(Problems Only and Theorems without Proofs)**

Textbooks

1. Joe L. Mott, Abraham Kandel, Theodare P. Baker, “Discrete Mathematics for Computer Scientists and Mathematicians”, Second Edition, PHI, 2019.
2. J. P. Tremblay and P. Manohar, “Discrete Mathematical Structures with Applications to Computer Science”, Tata McGraw Hill, 2007

References

1. K. H. Rosen, “Discrete Mathematics and its Applications with Combinatorics and Graph Theory”, 7th Edition, Tata McGraw Hill.
2. S. K. Chakraborthy and B.K. Sarkar,” Discrete Mathematics “, Oxford, 2011.
3. C. L. Liu and D. P. Mohapatra, “Elements of Discrete Mathematics-A Computer Oriented Approach”, 3rd Edition, Tata McGraw Hill.

Advanced Data Structures

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
EMA2120	Theoretical & Practical (PC)	L	T	P	C	CIE	SEE	Total
		3	0	2	4	50	50	100

Prerequisites

Any Programming Language

Course Outcomes

After completing the Advanced Data Structures Course Students will be able to

1. Implement Stack applications. (L3)
2. Apply the principles of Heaps and Binary Search Trees in solving complex problems. (L3)
3. Implement Advanced Trees concepts. (L3)
4. Apply graph-based algorithms to solve real time applications. (L3)
5. Illustrate hashing and collision resolution techniques. (L3)

UNIT I

Applications of Stack -Implementations of Towers of Hanoi, Parenthesis checker, conversions from infix to prefix and infix to postfix.

Trees: Basic terminology, Binary Tree, Complete Binary Tree, Full Binary Tree, Representation of Trees-Using Arrays and Linked lists (advantages and disadvantages), Implementation of tree traversal.

UNIT II

Extended Binary Tree and Threaded Binary Trees with focus on Inorder Threading, Representation of Algebraic Expressions and its implementation.

Representation and Creation of Binary Search Trees (BST), implementation, insertion, deletion and searching in BST.

Heaps: Introduction, Types of Heaps – Min binary heap, Max binary heap.

UNIT III

AVL Trees: Representation and its advantages, Operations in AVL Trees-insertion, deletion and rotation.

B-Trees: Definition and advantages, B-Tree of Order M, Insertion and Searching in B-trees. Introduction to Red-Black Trees and Splay Trees.

UNIT IV

Graphs: Basic terminology, Representation of Graphs: Sequential and linked representation, Graph Traversals-Breadth First Search, Depth First Search with algorithms and implementation.

Spanning Trees: Definition and its properties, Minimum Spanning Tree, Exploring Minimum Spanning Tree Algorithms: Implementation of Prim's and Kruskal's.

Implementation of Dijkstra Algorithms for finding shortest path in graphs.

UNIT-V

Hashing: General Idea, Hash Functions, Collision Resolution Techniques: Separate Chaining, Open Addressing-Linear probing, Quadratic Probing, Double Hashing.

Additional concepts: Rehashing, Extendible Hashing and Implementation of Dictionaries.

Text books

1. Reema Thareja, Data Structures using C, Second Edition, Oxford University Press 2014.

Reference books

1. Horowitz and Sahani, Fundamentals of Data Structures, Galgotia Publications Pvt Ltd. Delhi India, 2015.
2. Richard F. Gillberg & Behrouz A. Forouzan, Data Structures, A Pseudo code Approach with C, Second Edition, Cengage Learning, India Edition, 2005.

E- Resources

1. https://nptel.ac.in/courses/106102064
2. https://www.nesoacademy.org/cs/01-data-structures

Advanced Data Structures Lab

Prerequisites

Data structures

Course Outcomes

After completing the Advanced Data Structures Lab Course Students will be able to

1. Develop programs on stack applications.
2. Demonstrate the implementation of various advanced trees.
3. Design and implementation of programs on Graph Traversals.
4. Develop the programs on Hashing and Dictionaries

Week I

1a. Implement towers of Hanoi 1b. Implement parenthesis checker

Week 2

2a. Implement conversion of infix to postfix notation 2b. Implement evaluation of postfix notations

Week 3

3a. Implement tree traversals

i)In order ii) Preorder iii) Post order

3b. Perform level order traversal of a binary tree in a zigzag fashion:

First level left to right

Second level right to left

Third level left to right, and so on.

3c. Given a binary [S1] tree and two node values, find their Lowest Common Ancestor (LCA).

Week 4

4a. Implement Heaps

i)Min Heap ii) Max Heap

4b. Given an array of N integers and an integer K, find the Kth largest and Kth smallest element.

Week 5

Implement insertion, deletion and searching in Binary Search Tree

Week 6

6a. Implement insertion on AVL Trees

6b. Implement deletion and rotation on AVL Trees

Week 7

Skill test-1

Week 8

Implement B-Trees

i)Insertion ii) Search iii) Display

Week 9

Implement Graph traversals

i)Breadth First Search ii) Depth First Search

Week 10

10a. Implement

i) Prim's algorithm ii) Kruskal's algorithm

10b. Implement Dijkstra algorithm

Week 11

11. Implement Hashing and collision resolution techniques

Week 12

12. Implement Dictionaries

Week 13

Review

Programming in Java

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
EMA2X19	Theoretical & Practical (PC)	L	T	P	C	CIE	SEE	Total
		3	0	2	4	50	50	100

Prerequisites

Data Structures

Course Outcomes

After completing the Course Students will be able to

1. Apply OOP concepts to implement classes, objects in Java. [L3]
2. Develop Java applications using inheritance, polymorphism, and exception handling. [L3]
3. Implement modular code using packages and perform file operations with I/O streams. [L3]
4. Apply collections and multithreading for efficient and concurrent Java programs. [L4]
5. Implement functional-style solutions using Java lambdas and streams in real-world scenarios. [L3]

UNIT I

Java Basics: Introduction to OOPs concepts**,** History of Java, Java Virtual machine, Java features, data types, variables, scope and lifetime of variables.

Classes and Objects- Class declaration, Object declaration, Reference variables, access control, constructors, this keyword, static keyword, garbage collection, passing parameters to methods, nested and inner classes, Arrays, Strings, StringBuilder, StringBuffer.

UNIT II

Inheritance: Introduction, Types of Inheritances, Member access rules, super and final keywords. **Polymorphism**-Introduction, Compile time polymorphism-Method overloading, Run time polymorphism- Method overriding, Dynamic method dispatch, abstract classes, interfaces.

Exception handling – Introduction, exception hierarchy, errors, usage of try, catch, throw, throws and finally, built in exceptions, creating custom exception.

UNIT III

Packages: Understanding the creation and usage of custom packages, access control rules within packages, utilization of built-in Java packages, Date, Random, StringTokenizer, Scanner

IO Streams: InputStream, OutputStream, FileInputStream, FileOutputStream, ObjectInputStream, ObjectOutputStream, Reader, Writer, BufferedReader, FileReader, and FileWriter.

UNIT-IV

Collections: Overview of Java Collection Framework, Collection Interfaces – Collection, Set, List, Map, Collection classes – Array List, Linked List, Hash Set, Tree Set, Hash Map, Tree Map,

Multi-threading: Introduction, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication

UNIT- V

Modern Java Features: Introduction to Functional Interfaces, Lambda Expressions, Method References, Stream API – operations: map, filter, reduce, collect, forEach, Optional Class, Built-in functional interfaces – Predicate, Consumer, Function, Supplier.

Text books

1. Herbert Schildt, “Java: The Complete Reference”, 12th Edition, Tata McGraw Hill Publications, 2020.

References

1. Understanding OOP with Java, updated edition, T. Budd, Pearson Education.
2. Thinking in Java Fourth Edition, Bruce Eckel

Online Resources

1. Java Programming Fundamentals: https://infyspringboard.onwingspan.com/web/en/app/toc/lex_29959473947367270000_shared/overview
2. Programming in Java: https://onlinecourses.nptel.ac.in

Programming in Java Lab

Week-1

1a) Write a program to demonstrate execution of static blocks, static variables and static methods.

1b) Write a Java program to read 10 integers into an array and print the maximum and minimum values

1c) Write a Java program to input a string and count the number of vowels, consonants, digits, and special characters

Week-2

Create a Bank Account class each account has an account Holder Name, account Number, and balance. Create both no argument and parameterized constructors, and use 'this' keyword to initialize the values to class data members. to implement following Methods

- Void createAccount()
- Void deposit(int)
- Void withDraw(int)
- Void displayAccountDeatils();

Week-3

3a) Write a Java program that defines a class containing a member inner class and a static nested class. Demonstrate how to create objects of both types of inner classes and access their members.

3b) Write a program for sorting a given list of names in ascending order

Week-4

4a) Write a program to implement single and multi-level inheritance

4b) Write a program to create an abstract class named Shape that contains two integers and an empty method named printArea (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea () that prints the area of the given shape.

Week-5

5a) Write a program to demonstrate the use of super and final keywords.

5b) Design a Java program to implement multiple inheritance using interfaces to Create an interface Vehicle with methods like start (), stop (), and fuel Type (). Implement this interface in two classes: Car and Bike, each providing its own version of these methods. allow users to create objects of Car and Bike, invoke their methods, and display appropriate outputs,

Week-6

6a) Write a java program to handle multiple exception using try and catch block

6b) Create a custom exception class InvalidAgeException that extends Exception. In the main () method, read the user's age and throw the exception if the age is less than 18. Use a try-catch block to handle the exception and display an appropriate message.

Week-7

Skill Test-1

Week-8

8a) Write a program to create user defined package and demonstrate various access modifiers.
8b) Write a java program to display the employee details using Scanner class
8c) Write a java to implement string tokenizer class

Week-9

9a) Write a Java program that reads the contents of a source text file and writes it to a destination text file using FileInputStream and FileOutputStream
9b) Write a Java program that demonstrates object serialization and deserialization.

Week-10

10a) Create a Java program that uses an ArrayList to store a list of student names. Perform the following operations:

- Add names to the list
- Remove a name by value and by index
- Update a name at a specific index
- Display all names using both a for-each loop and an iterator

10b) Create a HashSet to store a list of unique course names. Add, remove, and search for elements, and display all items using an enhanced for loop.
10c) Create a HashMap to store student roll numbers as keys and their names as values. Perform operations to add, update, remove, and retrieve entries. Display all key-value pairs

Week-11

11a) Write a Java program to simulate a bank account system where multiple threads attempt to deposit money into a shared account.
11b) To implement the classic Producer-Consumer Problem using Java multithreading with proper inter-thread communication using wait() and notify() methods.

Week-12

Write Java programs on Lambda Expressions and Built-in Functional Interfaces

Week-13

13a) Write Java programs on functional style of programming using Stream API operations.
13b) Write Java programs Optional class and Method Reference

Week-14

Recap of the programs from Week 1 to Week 13

Week-15

Case Study Presentations

Week-16

Skill Test-2

Foundations of Software Engineering

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
EMD2X04	Theoretical (PC)	L	T	P	C	CIE	SEE	Total
		3	0	0	3	50	50	100

Prerequisites

Knowledge of any programming language is Desirable.

Course Outcomes

1. Apply the knowledge of appropriate framework activities for a given project (L3).
2. Apply the right process model for a given project (L3).
3. Apply design principles for a given context (L3).
4. Understand various testing techniques for a given project (L2).
5. Identify various risks in project development (L2).

UNIT I

Software and Software engineering: The Nature of software: Defining Software, characteristics of software, Software application Domains, Legacy software; Defining the Discipline; The software process: The process framework, Umbrella Activities; Hooker's Principles of Software Engineering Practice. [Ch 1.1, 1.2, 1.3, 1.4, Book1]

Process Models: A Generic process model, Defining a Framework Activity, Perspective Process Models: Waterfall Model, Prototyping Model, Spiral Model. [Ch 2.1, 2.2, 2.5, Book1]

Human aspects of Software Engineering: Characteristics of a Software Engineer, The Psychology of Software Engineering, The Software Team, Team Structures, The Impact of Social Media.[Ch 5.1, 5.2, 5.3, 5.4, 5.5, Book1]

UNIT II

Agility and Process: What is Agility, What is an Agile Process: Agility Principles, The Politics of Agile Development; Scrum: Scrum Teams and Artifacts, Sprint planning meeting, Daily Scrum meeting, Sprint Review meeting, Sprint Retrospective, The XP Framework, DevOps. [Ch 3.1, 3.3, 3.4, 3.5, Book1]

Understanding Requirements: Requirements Engineering: Inception, Elicitation, Elaboration, Negotiation, Specification, Validation, Requirement Management; Establishing the Groundwork: Identifying Stakeholders, Recognizing Multiple viewpoints, Working Toward Collaboration, Asking the First Questions, Non-functional Requirements, Traceability; Requirement gathering: Collaborative Requirement gathering, Usage Scenarios, Elicitation Work Products; Building the Analysis Model: Elements of Analysis Model; [Ch 7.1, 7.2, 7.3, 7.5, Book1]

Requirements Modeling: Requirement Analysis, Scenario-Based Modeling, Class-Based Modeling, Functional Modeling, Behavioral Modeling. [Ch 8.1, 8.2, 8.3, 8.4, 8.5, Book1]

UNIT III

Software Design: Design within the Context of Software Engineering, Design Process: Software Quality Guidelines, Evolution of Software design; Design Concepts: Abstraction, Architecture, Patterns, Separation of Concerns, Modularity, Information Hiding, Functional Independence, Stepwise Refinement, Refactoring, Design Classes; The Design Model: Design Modeling Principles, Data Design Elements, Architectural Design Elements, Interface Design Elements, Component-Level Design Elements, Deployment-Level Design Elements**.** [Ch 9.1, 9.2, 9.3, 9.4, Book1]

Component -Level Design: What is a Component: An object-oriented View, The Traditional View, A Process-Related View; Designing Class-Based Components: Basic Design Principles, Component - Level Design Guidelines, Cohesion, Coupling; [Ch 11.1, 11.2, Book1]

User Experience Design: User Experience Design Elements, The Golden Rules, User Interface Analysis and Design, User Experience Analysis, User Experience Design, User Interface Design, Design Evaluation, Usability and Accessibility, Conventional Software UX and Mobility. [Ch 12.1, 12.2, 12.3, 12.4, 12.5, 12.6, 12.7, 12.8, 12.9, Book1]

UNIT III

Software Testing-Component level: A Strategic Approach to Software Testing: Verification and Validation; Organizing for Software Testing; Test-Case Design: Requirements and Use Cases, Traceability; White-Box Testing: Basis Path Testing, Control Structure Testing, Black Box Testing: Interface Testing, Equivalence Partitioning, Boundary Value Analysis; [Ch 19.1, 19.3, 19.4, 19.5, Book1]

Software Testing-Integration level: Software Testing Fundamentals: Black-box Testing, White-box Testing; Integration Testing: Top-Down Integration, Bottom-Up Integration, Continuous Integration, Integration Test Work Products; Validation Testing [Ch 20.1, 20.2, 20.5, Book1]

Software Metrics: Software Measurement: Measures, Metrics and Indicators, Attributes of Effective Software Metrics; Product Metrics: Metrics for Requirement Model, Design metrics for conventional software, Design metrics for Object oriented Software, Metrics for source code; Software Measurement, Metrics for Software Quality, An Example of FP Based Estimation. [Ch 23.1, 23.3, 23.7, 23.8, 25.6.4, Book1]

UNIT IV

Risk Management: Reactive Versus Proactive Risk Strategies, Software Risk, Risk Identification, Risk Projection, Risk Refinement, RMMM, RMMM Plan**.** [Ch 26.1, 26.2, 26.3, 26.4, 26.5, 26.6, 26.7, Book1]

Software process Improvement: What is SPI, SPI Process, CMMI; [Ch 28.1, 28.2, 28.3, Book1]

Emerging Trends in Software Engineering: Observing Software Engineering Trends, Identifying "Soft Trends", Technology Directions, Tools-Related Trends. [Ch 29.3, 29.4, 29.5, 26.6, Book1]

Text books

1. Roger S. Pressman, Bruce R. Maxim: Software Engineering - A Practitioner's Approach, Ninth Edition, McGraw-Hill Education, 2020

Reference books

1. Ian Somerville: Software Engineering, 9th Edition, Pearson Education,2016
2. Rajib Mall: Fundamental of Software Engineering, 5th Edition, PHI Learning Private Limited, 2018.

Online Resources

https://nptel.ac.in/courses/106/105/106105182/

Dynamics of Group Discussion

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
EVA2X25	Practical (HS)	L	T	P	C	CIE	SEE	Total
		0	0	2	1	50	50	100

Course Outcomes

1. Collaborate with others to generate creative solutions to problems and make informed decisions as a group
2. Present ideas confidently and persuasively, using appropriate language and nonverbal cues
3. Resolve conflicts and disagreements diplomatically
4. Demonstrate ethical behavior and responsibility in communication during group discussions.
5. Apply group discussion skills to real-world scenarios

UNIT-I

Introduction to Group Discussions, Importance of GDs, Major Areas of GD: Subject knowledge, Clarity of thought and expression

UNIT-II

Types of GD: Topical, Case Studies, Abstract

UNIT-III

Differences between GDs and Debates, Team Behavior and Leadership Skills

UNIT-IV

Active Listening Skills, Note-taking skills

UNIT-V

Practice GD: Non-Verbal communication, Etiquette: Do's and Don'ts of GD

Text books

1. Anand, Ganguly.Group Discussion for Jobs and Admission Arihanth publications,2020
2. Praba, l Frank Excel at Group Discussions by, Paperback, 2020
3. Group Discussion On Current Topics by P. N. Joshi Paperback, 2024
4. Mukta Mahajani, Let's Talk by Amazon,2020
5. Master the Group Discussion & Personal Interview : Complete Discussion on the topics asked by reputed B-schools & IIMs by Sheetal Dasarda, 201

Quantitative Aptitude and Logical Reasoning - I

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks %**		
ESE2X23	Practical (BS)	L	T	P	C	CIE	SEE	Total
		0	0	2	1	50	50	100

Course Outcomes:

After completing the course, students will be able to

1. Solve problems using divisibility, percentage changes, and profit-loss analysis.
2. Strengthen concepts of ratio, proportion, and partnership-based arithmetic.
3. Develop pattern recognition and algebraic manipulation skills.
4. Enhance accuracy in handling data averages and basic financial calculations.
5. Improve efficiency in solving work-time and rate-flow based problems.

UNIT I

Number System: Number Systems - Divisibility Rules, Unit Digit of a number (Power Cycles)
Number Systems - Highest power of a number in factorial, Number of Trailing Zeroes, Remainder Theorem
Decimal to Fraction Conversion, Factors, LCM & HCF Applications
Percentages: Introduction to percentages, Percentage Increase /Decrease, Results on Population,
Results on Depreciation, Variations, Applications of Percentage
Profit and Loss: Classification of Profit and Loss, Profit/ Loss Percentages, Successive Discount.

UNIT II

Ratio and Proportion: Definition of ratio and Proportion, Finding the resultant ratio.
Ages: Problems based on Ratios and ages.
Partnership: Relation between Partners, Period of Investments and Shares

UNIT III

Number Series: Number, Alphabet and Letter Series.
Algebra: Problems based on algebra.
Logarithms: Formulas and Problems based on Logarithms.
Coding and decoding: Classifications and Problems on Coding and Decoding

UNIT IV

Averages: Average of different groups, change in average by adding, deleting and replacement of objects
Percentages: Introduction to percentages, Percentage Increase /Decrease, Results on Population, Results on Depreciation, Variations, Applications of Percentage
Simple Interest: Simple interest, Principle, Rate, Amount, Applications of Simple interest

UNIT V

Time and Work: Calculating Efficiency, alternate days concept, work and wages.
Pipes and Cisterns: Problems based on Pipes and cisterns.
Chain Rule - Time and Work Concept Problems based on Chain rule.

Text Books

1. Verbal and Non Verbal Reasoning – R.S Agarwal, New Edition -2020, S. Chand.
2. Quantitative Aptitude – R.S Agarwal, New Edition- 2020, S. Chand.

Reference Books

1. Quantitative Aptitude: Abhijeet Guha, New Edition-2020, Mc Graw Hill.

Note: Each Unit has one assessment which consists of 25 Multiple Choice Questions for duration of 40 Minutes and the average of these five tests will be considered as CIE. The SEE consists of 75 Multiple Choice Questions for duration of 120 Minutes

Integrated Project - I

BTech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks %**		
EVA2101	Exploratory (PC)	L	T	P	C	CIE	SEE	Total
		0	0	2	1	80	20	100

Evaluation Guidelines for the course Integrated Project-I as mentioned below.

CIE/SEE	Assessment component	When	Marks
CIE	Project Report - Phase I	After the completion of 7 Weeks	40
	Project Report - Phase II	After the completion of 14 Weeks	40
SEE	Viva voce	Semester End	20
Overall	CIE assessments will be taken for 80 marks and SEE for 20 marks		

Computer Oriented Statistical Methods

B. Tech II Year II Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
EMD2X01	Theoretical (BS)	L	T	P	C	CIE	SEE	Total
		3	0	0	3	50	50	100

Course Outcomes

1. To understand the concept of least squares method, compute the correlation coefficient and regression lines.
2. To understand the various techniques of sampling theory.
3. To understand applications of large samples in a real scenario.
4. To understand applications of small samples and Analysis of Variance in a real scenario.
5. To make better decisions using multiple linear regression.

UNIT I

Moments, Random Variables, Expectation:

Moments-Definition, Central and Non-Central, Skewness, Kurtosis (Based on Moments only). **Random Variables:** Definition of random variable, discrete and continuous random variables, probability mass function and probability density function with applications.

Mathematical Expectation: Definition of expectation, Variance, covariance and their properties with applications. Definition and properties of moment generating function.

UNIT II

Probability Distributions:

Definitions, Derivation of mean and variance of Binomial, Poisson, Normal and Exponential Distributions and their applications.

UNIT III

Correlation and Regression, Testing of Hypothesis and Large sample tests:

Correlation and Regression: Bivariate data, Concept of correlation, computation of Karl-Pearson correlation coefficient. Spearman's rank correlation coefficient. Simple linear regression, correlation verses regression, lines of regression and properties of regression coefficients.

Testing of Hypothesis: Null and Alternative hypothesis, Critical region, two types of errors, Level of significance. One and two tailed tests. Procedure for testing of hypothesis.

Large Samples: Tests for single sample mean, Difference of means, single sample proportion, Difference of proportions.

UNIT IV

Small sample tests:

Degrees of freedom, Tests of significance based on student's t-test for single sample specified mean, difference of means for independent and Paired t-test. Chi-Square test for Goodness of fit and Independence of attributes. F - test for equality of population variances. ANOVA-One way.

UNIT V

Introduction-Queueing system-The arrival pattern-The service pattern-The queue discipline, Symbolic Representation of a Queueing Model –Characteristics of Infinite Capacity, Single server Poisson Queue Model, Queueing Problem-Pure Birth and Death Process-Probability Distribution of Departures (pure Death process)- Basic queueing Models-Measures of the (M/M/1): (∞/FIFO) model characteristic of Finite Capacity.

Text books

1. V. K. Kapoor and S. C. Gupta: Fundamentals of Mathematical Statistics, Sultan Chand & Sons, New Delhi.
2. Sheldon M Ross, Probability and statistics for Engineers and scientists, Academic Press.

Reference books

1. T.T. Soong, Fundamentals of Probability and Statistics for Engineers, John Wiley & Sons Ltd, 2004.
2. Dr T.K.V. Iyengar & Dr B. Krishna Gandhi & S. Ranganadham & Dr M.V.S.S.N. Prasad: Probability and Statistics, Sultan Chand & Sons, New Delhi.

Fundamentals of Computer Algorithms

B. Tech II Year II Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
EMA2221	Theoretical (PC)	L	T	P	C	CIE	SEE	Total
		2	0	0	2	50	50	100

Prerequisites

Knowledge about at least one Programming Language, Data Structures and Discrete Mathematics

Course Outcomes

At the end of this course, students will be able to:

1. Compute time and space complexities with asymptotic notations. (L3)
2. Derive the time complexity of divide and conquer algorithms with their implementation. [L3]
3. Implement optimization techniques on disjoint sets. [L3]
4. Design the graph algorithms. (L3)
5. Implementation of greedy algorithms for finding optimal solutions. (L3)

UNIT I

Introduction: Algorithm, Pseudocode representation, Performance Analysis - Space and Time complexity for iterative and recursive algorithms, Asymptotic Notation - Big oh notation, Omega notation, Theta notation, and little oh notation.

UNIT II

Divide and conquer: General method, Applications - Binary search, Quick sort, and Merge sort with algorithm, example and analysis.

UNIT III

Sets:

Disjoint Sets-Disjoint set operations: simple union, simple find, weighted union and collapse find, Bi-Connected Graph: connected and bi-connected components, finding articulation point.

UNIT IV

Graphs:

Graphs and its representations, Graph Traversal Techniques: Breadth-first search, depth-first search, spanning tree principles, Minimum Spanning tree - Prims and Kruskal's Algorithms.

UNIT V

Greedy method: General method, applications-Job sequencing with deadlines, Fractional Knapsack problem, Single source shortest path problem-Dijkstras algorithm, Bellman ford Algorithm.

Text Books

1. Ellis Horowitz, Satraj Sahni and Rajasekharam, Fundamentals of Computer Algorithms, Galgotia publications pvt. Ltd, Second Edition, 2007.

Reference Books

1. Thomas H. Cormen,Charles E. Leiserson, Ronald L. Rivert and Clifford Stein, Introduction to Algorithms, Third Edition ,PHI Learning Private Limited , Eastern Economy Edition, 2008.
2. Aho, Ullman and Hopcroft, Design and Analysis of algorithms, Pearson education, Reprint 2002

Online Resources

http://www.digimat.in/nptel/courses/video/106106131/L02.html

Database Management Systems

B. Tech II Year II Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
EMA2217	Theoretical & Practical (PC)	L	T	P	C	CIE	SEE	Total
		3	0	2	4	50	50	100

Course Objectives

1. Model Entity-Relationship diagrams for enterprise level databases [L3]
2. Formulate Queries using SQL and Relational Formal Query Languages [L3]
3. Apply different normal forms to refine the Database schema [L3]
4. Illustrate various concurrency control protocols and recovery algorithms [L2]
5. Apply NoSQL concepts to query MongoDB databases [L3]

UNIT I

Introduction to Database System Concepts: Database-System Applications, Purpose of Database Systems, View of Data, Database Architecture, Database Users and Administrators.

Introduction to the Relation Models and Database Design using ER Model: Structure of Relational Databases, Database Schema, Keys, Components of The Entity-Relationship Model, Constraints and types of relationships.

UNIT II

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Aggregate Functions, Nested Sub queries

Formal Relational Query Languages: The Relational Algebra, Tuple Relational Calculus

UNIT III

Relational Database Design: Features of Good Relational Designs, Atomic Domains and First Normal Form, Functional Dependencies, Closure set of Functional dependencies, second Normal Form, Third Normal Form, Boyce Codd Normal form.

Transactions: Transaction Concept, A Simple Transaction Model, ACID Properties, Serializability

UNIT IV

Concurrency Control: Lock-Based Protocols, Deadlock Handling, Multiple Granularity, Timestamp-Based Protocols, Validation-Based Protocols.

Recovery System: Failure Classification, Recovery and Atomicity, Recovery Algorithm-ARIES, Remote Backup Systems.

UNIT V

No SQL Databases: Why NoSQL, Emergence of NoSQL, Categories of NoSQL databases, Distribution Models: Single Server, Sharding, Master-Slave Replication, Peer-to-Peer Replication, Combining Sharding and Replication, Consistency: Update Consistency, Read Consistency, Relaxing Consistency-The CAP Theorem, Relaxing Durability, Quorums.

Introduction to Mongo DB: Introduction, Documents, Collections, Databases, Data types. (CRUD operations) Creating, Updating, Deleting and Querying the Documents

Text Books

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database System Concepts”, Sixth Edition, Tata McGraw-Hill 2006.
2. Pramod J. Sadalage, Martin Fowler, “NoSQL Distilled : A Brief Guide to the Emerging World of Polyglot Persistence”, Addison-Wesley 2013
3. Kristina Chodorow and Michael Dirolf , “MongoDB: The Definitive Guide”, Second edition, O’Reilly

Reference Books

1. Raghu Rama Kirshna, Johannes Gchrke, Database Management System, Third Edition, TATA MC Graw Hill, 2003.
2. C J Date, AKannan, S Swamynathan, An Introduction to Database Systems, Eigth Edition Pearson 2006
3. P Raja Sekhar Reddy, A MallikarjunaReddy, Foundations of Database Management Systems, Lambert Academic Publishing, 2020 (e-Book)

Online Resources

Database Management Systems: https://onlinecourses.nptel.ac.in/

Database Management System Lab

Week 1

Data Base user creation, Data definition Language commands, Data Manipulation commands, Data Control Language Commands, Transaction Control Language commands.

Week 2

Database Schema for a customer-sale scenario Customer (Cust id: integer, cust_name: string)
Item (item_id: integer,item_name: string, price: integer)
Sale (bill_no: integer, bill_data: date, cust_id: integer, item_id: integer, qty_sold: integer) For the above schema, perform the following—

a) Create the tables with the appropriate integrity constraints

b) Insert around 10 records in each of the tables

c) List all the bills for the current date with the customer names and item numbers

d) List the total Bill details with the quantity sold, price of the item and the final amount

e) List the details of the customer who have bought a product which has a price>200

f) Give a count of how many products have been bought by each customer

g) Give a list of products bought by a customer having cust_id as 5

h) List the item details which are sold as of today

i) Create a view which lists out the bill_no, bill_date, cust_id, item_id, price, qty_sold, amount
Create a view which lists the daily sales date wise for the last one week

Week 3

Database Schema for a Student Library scenario Student (Stud_no : integer,Stud_name: string)
Membership (Mem_no: integer,Stud_no: integer)
Book (book_no: integer, book_name:string, author: string) Iss_rec(iss_no:integer, iss_date: date, Mem_no: integer, book_no: integer) For the above schema, perform the following—

a) Create the tables with the appropriate integrity constraints

b) Insert around 10 records in each of the tables

c) List all the student names with their membership numbers

d) List all the issues for the current date with student and Book names

e) List the details of students who borrowed book whose author is CJDATE

f) Give a count of how many books have been bought by each student

g) Give a list of books taken by student with stud_no as 5

h) List the book details which are issued as of today

i) Create a view which lists out the iss_no, iss _date, stud_name, book name

j) Create a view which lists the daily issues-date wise for the last one week

Week 4:

Database Schema for a Employee-pay scenario employee (emp_id : integer,emp_name: string)
Department (dept_id: integer,dept_name:string)
Paydetails (emp_id : integer,dept_id: integer, basic: integer, deductions: integer, additions: integer, DOJ: date)
Payroll (emp_id : integer, pay_date: date)
For the above schema, perform the following—

a) Create the tables with the appropriate integrity constraints

b) Insert around 10 records in each of the tables

c) List the employee details department wise

d) List all the employee names who joined after particular date

e) List the details of employees whose basic salary is between 10,000 and 20,000

f) Give a count of how many employees are working in each department

g) Give a names of the employees whose netsalary>10,000

h) List the details for an employee_id=5

i) Create a view which lists out the emp_name, department, basic, deductions, netsalary

j) Create a view which lists the emp_name and his netsalary

Week 5

Database Schema for a Video Library scenario Customer (cust_no: integer,cust_name: string)
Membership (Mem_no: integer, cust_no: integer)
Cassette (cass_no:integer, cass_name:string, Language: String)
Iss_rec(iss_no: integer, iss_date: date, mem_no: integer, cass_no: integer) For the above schema, perform the following—

a) Create the tables with the appropriate integrity constraints

b) Insert around 10 records in each of the tables

c) List all the customer names with their membership numbers

d) List all the issues for the current date with the customer names and cassette names

e) List the details of the customer who has borrowed the cassette whose title is " The Legend"

f) Give a count of how many cassettes have been borrowed by each customer

g) Give a list of books which has been taken by the student with mem_no as 5

h) List the cassettes issues for today

i) Create a view which lists outs the iss_no, iss_date, cust_name, cass_name

j) Create a view which lists issues-date wise for the last one week

Week 6

Database Schema for a student-Lab scenario Class (class_no: string,descrip: string)
Student (stud_no: integer, stud_name: string, class_no: string) Lab (mach_no: integer, Lab_no: integer, description: String)
Allotment (Stud_no: Integer, mach_no: integer, dayof week: string) For the above schema, perform the following—

a) Create the tables with the appropriate integrity constraints

b) Insert around 10 records in each of the tables

c) List all the machine allotments with the student names, lab and machine numbers

d) List the total number of lab allotments day wise

e) Give a count of how many machines have been allocated to the 'CSIT' class

f) Give a machine allotment etails of the stud_no 5 with his personal and class details

g) Count for how many machines have been allocatedinLab_no1 for the day of the week as "Monday"

h) How many students class wise have allocated machines in the labs

i) Create a view which lists out the stud_no, stud_name, mach_no, lab_no, dayofweek

j) Create a view which lists the machine allotment details for "Thursday".

Week 7

Review & Skill Test-1

Week 8

Write a program to find largest number from the given three numbers. Simple programs using loop, while and for iterative control statement. Write a program to check whether the given

number is Armstrong or not Write a program to generate all prime numbers below 100.
Write a program to demonstrate the GOTO statement.

Write a program to demonstrate %type and %row type attributes

Week 9

Write a program to demonstrate predefined exceptions Write a program to demonstrate user defined exceptions

Create a cursor, which displays all employee numbers and names from the EMP table.

Week 10

Create a trigger before/after update on employee table for each row/statement. Create a trigger before/after delete on employee table for each row/statement. Create a trigger before/after insert on employee table for each row/statement.

Week 11 – Apply NFs to keep Schema in III NF

Apply NFs to keep Schema in III NF i.

ID	Name	C_ID	Course_Name	Instructor_ID	Instructor_Name	Grade
1	Alice	C101	Math	I01	Dr. Smith	A
1	Alice	C102	Physics	I02	Dr. Brown	B
2	Bob	C101	Math	I01	Dr. Smith	B

ii.

E_ID	E_Name	DID	D_Name	Proj_ID	Proj_Name	Hours_Worked
101	Alice	D1	HR	P1	Recruitment	20
101	Alice	D1	HR	P2	Employee Ret.	15
102	Bob	D2	IT	P3	Network Setup	25
103	Carol	D3	Finance	P4	Audit	30
103	Carol	D3	Finance	P5	Budget Review	10

Week 12

Introduction to Mondo DB lab from the Mongo DB University and need to acquire Mongo DB essentials and MongoDB ATLAS Certificates

Week 13

Review & Skill Test -2

Programming in Python

B. Tech II Year II Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks %**		
EMA2X18	Theoretical and Practical (PC)	L	T	P	C	CIE	SEE	Total
		3	0	2	4	50	50	100

Course Outcomes

At the end of this course, students will be the able to:

1. Develop skills in handling control flow using conditionals and loops. (L4)
2. Analyze various String handling functions and data structures. (L4)
3. Design programs on object-oriented programming concepts. (L4)
4. Solve the problems by using Inheritance and polymorphism. (L3)
5. Illustrate programs on Exception Handling and various packages. (L3)

UNIT I

Introduction to Python: Features of Python literal constants, variables and identifiers, Data Types, Operators, Expressions, type conversions. Decision control statements: Conditional branching statements, loop structures/iterative statements, nested loops, break, continue and pass statements, Standard I/O Operations. **Functions and Modules**: Declaration and Definition Function Calling, More on Defining Functions, Recursive Functions, Modules and Packages.

UNIT II

Strings and Regular Expressions: String Operations, Built-in String Methods and Functions, slice operation, functions in Regular Expression. **Sequence:** List: Introduction, nested list, list operations, list comprehensions. **Tuples:** Introduction, basic operations, advantages of Tuple over list. **Mappings:** Dictionaries, Sets: Introduction and operations, Frozen set, Byte, Byte array.

UNIT III

FILE HANDLING: Introduction, Opening and closing files, reading and writing files, file positions renaming and deleting files. **Implementation of classes and objects:** Classes and Objects, Class Method and Self Argument. The __init__m e t h o d , Class Variables and Object Variables, The del method, Public and Private Data Members, Private Methods, Built-in Functions to Check, Get, Set and Delete Class Attributes, Garbage Collection (Destroying Objects).

UNIT IV

Implementation of Inheritance in Python: Inheriting Classes in Python, Types of Inheritance, Abstract Classes. **Operator Overloading in Python**: Introduction, Implementing Operator Overloading, Overriding Methods. **Exception Handling in Python**: Introduction, Exception hierarchy, Handling Exception, Multiple except Blocks and Multiple Exceptions, Finally Block.

UNIT V

NumPy: NumPy ND array, Data Types, Functions of NumPy Array, Mathematical Functions on Arrays in NumPy. **Pandas:** Pandas Features, Dataset in Pandas, Data Frames, Manipulating the Datasets, Describing a Dataset, group by Function, Filtering, Missing Values in Pandas, Concatenating Data Frames. Import data from csv file. **Introduction to Matplotlib:** Plot, Scatterplot, Introduction to Tkinter, Date and Time Packages.

Text Books

1. Reema Thareja, Python Programming using Problem Solving Approach, Second Edition, Oxford Higher Education, 2022.
2. James Payne, Beginning Python using Python 2.6 and Python 3,1st Edition

Reference Books

1. Charles Dierach, Introduction to Computer Science using Python, 2013.

Online Resources

1. https://www.programiz.com/python-programming
2. https://www.geeksforgeeks.org/python-programming-language

Programming in Python Lab

Week 1 Loops and Operators

1.1 Installation and Environment set up of Python and Programs on Data types

1.2 Demonstrate the working of following functions

a) id( )
b) type( )
c) range( )

1.3 Demonstrate the following Operators in Python with suitable examples.

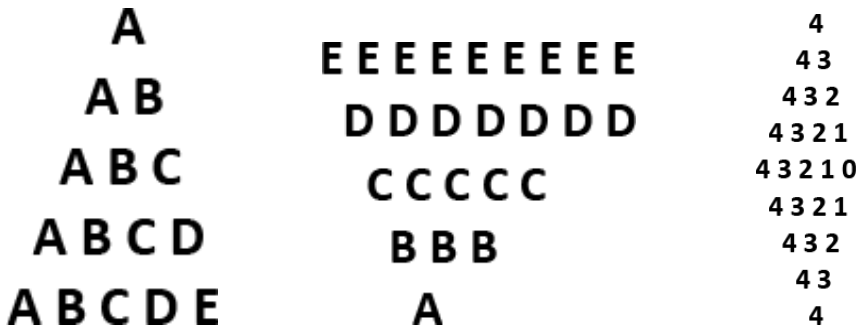
a) Arithmetic Operators
b) Relational Operators
c) Assignment Operator
d) Logical Operators
e) Bit wise Operators
f) Ternary Operator
g) Membership Operators
h) Identity Operators

1.4 Write a program that reverses the digits of a number, adds the reversed number to the original, and checks, if the sum is a palindrome the process is to be stopped. If the sum is not a palindrome, repeat the process (reverse the sum, add it to the original sum, and check again) until the sum becomes palindrome.

1.5 Demonstrate the following control transfer statements in Python with suitable examples.

a) Break
b) Continue
c) pass

1.6 Write Python programs to print the following Patterns:



Week 2 Conditional Statements, Functions

2.1 Write a program to calculate the total amount of money in the piggy bank, given coins of Rs10, Rs5, Rs2, and Rs1.

2.2 Write a program to convert a floating-point number into the corresponding integer.

2.3 Write a program to prepare a grocery bill. For that, enter the name of the items purchased, the quantity in which it is purchased, and its price per unit. Then, display the bill in the following format:

**********BILL**********
Item Name Item Quantity Item Price
****

Total amount to be paid
****

Write a program that shows how to return multiple values from a function

2.4 Write a program to demonstrate various argument types.

Week 3 Strings, Regular Expression

3.1 Perform following operations on strings

i) len( )	ii) strip( )	iii) rstrip( )	iv) lstrip( )	v) find( )
vi) rfind( )	vii) index( )	viii) rindex()	ix) count( )	x) replace()
xi) split( )	xii) join( )	xiii) upper( )	xiv) lower( )	xv) swapcase()
xvi) title( )	xvii) capitalize( )	xviii) startswith()	xix) endswith()	

3.2 Write a program for performing slice operation on strings.

3.3 Write a program to slice a given String into two parts: the first half and the second half. If the string has an odd length, the second half will have one more character than the first half.

3.4 Write a Regular Expression to represent all RGM language (Your own language) identifiers.

Rules:

a) The allowed characters are a-z, A-Z, 0-9, #.
b) The first character should be a lower-case alphabet symbol from a to k
c) The second character should be a digit divisible by 3.
d) The length of identifier should be at least.

3.5 Write a program to check whether the given string is RGM language identifier or not?

3.6 Write a Regular Expression to represent all 10 digit mobile numbers.

Rules:

a) Every number should contain exactly 10 digits.
b) The first digit should be 7 or 8 or 9

3.7 Write a program to check whether the given number is valid mobile number or not?

Week 4 Lists, Tuples

4.1 Demonstrate the different ways of creating list objects with suitable example programs.

4.2 Demonstrate the following functions/methods which operates on lists in Python with suitable examples:

a) list( )
b) len( )
c) count( )
d) index ( )
e) append( )
f) insert( )
g) extend()
h) remove( )
i) pop( )
j) reverse( )
k) sort( )
l) copy( )
m) clear( )

4.3 Demonstrate the following with suitable example programs:
i) List slicing ii) List Comprehensions

4.4 Write a program to store Fibonacci numbers in a list and sum even terms.

4.5 Inventory Management System In this case study, manage an inventory system. The inventory items will be stored as tuples containing the item name and price (immutable), and the list of items in the inventory will be stored as a list for dynamic changes.

- Demonstrate the different ways of creating tuple object with suitable example programs.
- Demonstrate the following functions/methods which operates on tuples in Python with suitable

Examples:

a) len( )
b) count( )
c) index( )
d) sorted( )
e) min ( )
f) max( )
g) cmp( )
h) reversed( )

Week 5 Dictionaries

5.1 Demonstrate the different ways of creating dictionary objects with suitable example programs.

5.2 Demonstrate the following functions/methods which operates on dictionary with suitable examples:

a) dict( )
b) len( )
c) clear( )
d) get( )
e) pop( )
f) popitem( )
g) keys( )
h) values()
i) items( )
j) copy( )
k) update( )

5.3 Write a program to unzip a list of tuples into individual lists and convert them into a dictionary.

5.4 Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

5.5 E-Commerce Product Catalog Management Manage the product catalog for an e-commerce website. The catalog needs to store information about each product, including its ID, name, price, category, and available stock.

Should be able to:

- Add new products.
- Update the details of existing products.
- Search for a product by its ID.
- Calculate the total value of products in stock (based on price and stock).

Week 6 Files

6.1 Write a Program for Handling Files.

6.2 Write a program to Count the Frequency of Characters in a Given File

6.3 Write a program to Compute the Number of Characters, Words, and Lines in a File.

Case Study:

Customer Feedback Management System Using Files

Collect Feedback: Collects feedback from the customer and stores it in a file.

Read Feedback: Reads and displays all feedback entries.

Search Feedback: Searches for feedback entries by customer name.

Summarize Feedback: Counts how many feedback entries are positive or negative based on keywords like "good," "excellent," "bad," and "poor."

Week 7 Class, Objects

7.1 Write a program to create Class variables and instance variable and illustration of self - variable

i. Robot

ii. ATM Machine

7.2 Write a program to differentiate private and public variables in a class.

7.3 Write a program to demonstrate built -in functions to check, sget, set and delete class attributes.

Week 8 Inheritance, Polymorphism

8.1 Write Programs to demonstrate Inheritance and Polymorphism .

8.2 Create a Person class with attributes like name and age, and a Student class that inherits from Person and adds an attribute grade.

8.3 Create a Shape class with methods like area() and perimeter(). Derive Circle, Rectangle, and Triangle classes, each implementing specific calculations.

8.4 Use constructors (__ init__()) in both parent and child classes. Demonstrate how the child class initializes both its own attributes and inherited attributes using super().

8.5 Create a program where the child class overrides a method of the parent class, such as display_details().

8.6 Overload operators like + or * in a class to perform custom operations, e.g., adding two objects of a Vector class.

8.7 Create a base class Product and child classes Electronics, Clothing, and Grocery with a polymorphic method get_discount().

8.8 Create a base class Transport and subclasses Car, Bus, and Bicycle, each overriding a method like fuel_cost().

Week 9 Exception Handling

9.1 Write Programs demonstrating various techniques of exception handling, such as:

- Basic exception handling (try, except).
- Handling multiple exceptions.
- Raising custom exceptions.
- Handling exceptions with files.

Week 10

10.1 Write a NumPy program to compute the cross product of two given vectors.

10.2 Write a program to perform mathematical operations on Numpy arrays.

10.3 Write a program to perform stack Operations using NumPy.

10.4 Write a program to perform Queue Operations using NumPy

10.5 **Case Study:** Sales Data Analysis using NumPy.

A company wants to analyze its sales data for the past year across multiple regions and product categories. The data consists of daily sales figures, and the company needs to:

i. Analyze the sales performance across different months.
ii. Find the highest and lowest sales days.
iii. Calculate the average sales for each region and product category.
iv. Perform statistical analysis like variance and standard deviation.

Week 11

11.1 Demonstration of Pandas Package

11.2 Write a Program to convert a panda module series to python list.

11.3 Write a program to convert a Numpy array to a pandas series.

11.4 Case Study: Employee Performance Analysis using Pandas Problem Statement: A company wants to analyze the performance of its employees over the past year based on various metrics such as:

- Sales performance.
- Number of projects completed.
- Customer satisfaction scores.

The company has a dataset that contains employee information, such as:

- Employee ID
- Name
- Department
- Sales performance
- Number of projects completed
- Customer satisfaction score

11.5 Create a project to get the citation from Google scholar using title and year of publication and volume, pages of journal.

Week 12 Package

12.1 Demonstration of Matplotlib Package

12.2 Case Study: Sales Data Visualization using Matplotlib

Problem Statement: A company has sales data for multiple regions over the past year.

The data includes:

- Monthly sales figures for each region.
- Sales figures for each product category within each region.

The company wants to visualize the following:

- **Overall sales trends**: A line graph showing the sales trend over the months.
- **Sales comparison across regions**: A bar graph to compare sales in different regions.
- **Category-wise sales distribution**: A pie chart showing the share of sales for different categories in each region.
- **Monthly sales distribution across regions**: A stacked bar chart showing how each region contributed to the overall monthly sales.

Week 13

Review

Web Technologies

B. Tech II Year I Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks %**		
EMA2219	Theoretical & Practical (PC)	L	T	P	C	CIE	SEE	Total

Course Outcomes

At the end of this course, students will be able to:

1. Develop static web pages using fundamental HTML elements
2. Create responsive and interactive web applications using JavaScript, ReactJS, and Bootstrap.
3. Design and deploy structured data formats and server-side applications using XML and servlets.
4. Build dynamic, database-driven web pages using JSP and JDBC.
5. Develop robust web applications and RESTful APIs using Spring Boot.

UNIT I

HTML: HTML Basics Elements, Attributes, Tables, Forms div and span tags

CSS: Introduction to cascading style sheet, Types of style sheets, page layout, selectors, pseudo classes and elements

UNIT II

JAVA SCRIPT: Introduction to scripting, control structures, Arrays functions, objects.

DOM: creating nodes, adding nodes, inserting nodes, removing & Replaces Nodes, front end frameworks(bootstrap).

ReactJS: React basics and setup, states and use State, component, props, stylish react application, use Effect, Custom Hook

UNIT III

XML: Basics of XML, Elements, Attributes, Name space.

Servlets: Introduction, Lifecycle, Generic and HTTP servlet, passing parameters to servlet, HTTP servlet Request & Response interfaces, Deploying web Applications, Session Tracking: Hidden form fields, cookies, URL- Rewriting, session

UNIT IV

JSP: Introduction, Difference Between servlets & JSP, Anatomy of JSP page, JSP elements: Directives, comments, Expressions, script lets, Declaration, Implicit JSP objects, using Action elements.

JDBC: Introduction, JDBC Drivers, Loading Driver, establishing connection, Executing SQL statement in JSP pages

UNIT V

MVC Architecture: Spring Framework, Spring Architecture, Setting Spring Development Environment, Spring MVC, Examples

Spring Boot: Introduction to Spring Boot, what is Spring Boot, building an end-to-end application (RESTFull API), Spring Boot Features, Configuration and Customization, Project on Spring Boot

Reference Books:

1. Learning Spring Boot 3.0: Simplify the development of production-grade applications using Java and Spring Kindle Edition by Greg L. Turnquist (Author), DaveSyer (Foreword), Mark Heckler (Foreword), Josh Long (Foreword)

Online learning resources

1. https://onlinecourses.swayam2.ac.in/nou24_cs09/preview
2. https://www.geeksforgeeks.org/web-technology/

Web Technologies LAB

Week - 1

Design the following static web pages for College Information System.

- A header with the College name and a navigation menu.
- A table display the program details menu.

Week - 2

Design Student Inquiries Form using form tag

Week - 3

Apply the CSS properties to College Information System web site

Week - 4

Implement JavaScript Validation for a Student Inquiries Form
Write a JavaScript program to validate the input fields of a student inquiries form. The form should include fields like Name, Email, and Message, and enforce the following validation rules:

- Name: Must not be empty and should only contain alphabets.
- Email: Must follow a valid email format.
- Message: Must not be empty and should have a minimum length of 10 characters.

Week - 5

Develop a Student Registration Portal using Bootstrap to create a responsive and mobile-friendly user interface.

- The portal will allow students to register by filling out a form
- The registration details will be displayed in a table format.

Week - 6

Skill Test -I

Week - 7

Build a Shopping Cart Functionality for the bookstore using ReactJS: Use state management to allow users to:

- Add books to the cart.
- Remove books from the cart.

Week - 8

Explore the basics of XML to represent a catalog of books or a student database by creating an XML document.

The program should demonstrate:

- Defining well-formed XML structure with elements and nested elements.
- Implementing namespaces to avoid naming conflicts in XML documents.

The program should include a practical example, such as creating an XML file to represent a

catalog of books or a student database

Week - 9

Build a web application to demonstrate parameter passing in Servlets by creating a simple student registration system. The application should:

- Collect student details (e.g., Name, Email, Course) through an HTML form.
- Pass these details as parameters to a Servlet.
- Process and display the details dynamically using the Servlet.

Week - 10

Develop a web application that uses session tracking to manage user data across multiple HTTP requests. The application will allow students to log in using their credentials and display personalized information throughout the session.

Features

- Students can log in via a login form.
- After successful authentication, their information will be stored in a session.
- The application will display personalized details on a welcome page.
- Students can log out, which will invalidate the session.

Week - 11

Create a student portal web application to showcase the use of JSP implicit objects. The portal will provide the following functionalities:

- Student Login: Students can log in by providing their username and password.
- Session Management: Student details are stored in a session to maintain user state across multiple pages.
- Logout Functionality: Students can log out, which will invalidate the session and redirect them to the login page.

Week - 12

Build a simple Student Management System that demonstrates the use of JDBC to interact with a database.

- Adding, viewing, updating, and deleting student records from a database

Week - 13

Build a RESTful API for Book Management using Spring Boot:

- GET all books.
- POST a new book.
- DELETE a book by ID.

Week - 14

Recap of the programs from Week 1 to Week 13

Week - 15

Case Study Presentations / Skill Test-2

Advanced Reading Comprehension Skills

B. Tech II Year II Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks %**		
EAE2225	Practical (HS)	L	T	P	C	CIE	SEE	Total
		0	0	2	1	50	50	100

Introduction:

The course on **Advanced Reading Comprehension Skills** is exclusively dedicated to Reading Comprehension and focus deeply on developing strategies and skills to understand and analyze texts effectively for promoting critical thinking skills for all practical purposes.

Learning Objectives:

Course objectives are to:

1. Analyze and interpret complex texts.
2. Exhibit the improved efficiency in reading and answering comprehension questions.
3. Gain confidence in tackling academic and professional reading challenges.

Course Outcomes

At the end of this course, students will be the able to:

1. Develop a systematic approach to analyze and interpret texts effectively.
2. Employ strategies such as skimming, scanning, and critical reading to enhance comprehension.
3. Enable students to confidently answer a diverse range of comprehension questions with precision.
4. Evaluate texts critically to express well-reasoned perspectives.

Exercise – I:

Introduction - Types of Texts: Narrative, Descriptive, Expository and Argumentative -Speed vs. Comprehension (Small paragraphs for Comprehension from the Workbook)

Exercise – II:

Strategies for effective Reading: Pre - Reading Strategies (Skimming, Scanning) - Strategies for **Active Reading:** Annotating, questioning and clarifying) (Interpreting the texts and comprehending the Blogs and Newspaper articles from the Workbook)

Exercise – III:

Critical Reading Skills: Comprehending the text, context clues, (Understanding vocabulary in context), Recognizing inference and implication, differentiating facts from Opinions (Reading and analyzing complex Passages and /Journal articles to comprehend from the Workbook)

Exercise – IV:

Critical Reading Project: Book analysis and Presentation (With the approval of the respective faculty, the student must critically analyze a book and present his/her interpretations)

Reference Books

1. Arun Sharma and Meena Upadhyay “Verbal Ability & Reading Comprehension” McGraw Hill 2024
2. Speed Reading made Easy by Martin Klarkson, Pearson,2020
3. Nick Bell “Reading Skills: How to Read Better and Faster - Speed Reading, Reading Comprehension & Accelerated Learning” Nick Bell 2021
4. John R. Torrance “Proven Speed-Reading Techniques” High Performance Media 2022

Quantitative Aptitude and Logical Reasoning - II

B. Tech II Year II Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks**		
ESE2X24	Practical (BS)	L	T	P	C	CIE	SEE	Total
		0	0	2	1	50	50	100

Course Outcomes

1. Solve real-life problems using speed, distance, and time formulas.
2. Accurately analyze relationships and directions in logical reasoning scenarios.
3. Apply permutation, combination, and probability concepts to structured problems.
4. Solve geometric and algebraic problems using fundamental principles and formulas.
5. Analyze graphical data and accurately solve calendar and clock-based questions.

UNIT I

Time and Distance: Difference between the average, Relative and Effective speed, reaching the destination late and early, stoppage time per hour.
Problems on Trains: Problems based on Trains.
Boats and Streams: Problems based on Boats and Streams.

UNIT II

Direction Sense Test: Sort of directions in puzzle distance between two points, Problems on shadows.
Blood Relations: Defining the various relations among the members of a family, Solving blood relation puzzles by using symbols and notations. Problems on coded relations.
Analogy: Simple, Double, Word and Number Analogy
Inequalities: Problems based on inequalities

UNIT III

Permutations and Combinations: Fundamental rules, Problems on Permutations and Combinations
Probability: Definition, Notations and Problems based on Probability

UNIT IV

Surds & Indices
Geometry: Line, line segment, angle, Triangles and Polygons with their Properties.
Coordinate Geometry: Different types of formulas and Problem based on Coordinate Geometry

UNIT V

Data Interpretation: Tabular, Pie-charts, Bar and line graphs and Problems on all models.
Clocks: Relation between minute and hour hand, angle between hands of a clock, exceptional cases in clocks. Gaining and loosing of time.
Calendars: Classification of years, finding the day of any random calendar date, repetition of calendar years.

Text Books

1. Verbal and Non Verbal Reasoning – R.S Agarwal, New Edition -2020, S. Chand.
2. Quantitative Aptitude – R.S Agarwal, New Edition- 2020, S. Chand.

Reference Books

1. Quantitative Aptitude: Abhijeet Guha, New Edition-2020, Mc Graw Hill

Note: Each Unit has one assessment which consists of 25 Multiple Choice Questions for duration of 40 Minutes and the average of these five tests will be considered as CIE. The SEE consists of 75 Multiple Choice Questions for duration of 120 Minutes.

Integrated Project - II

B. Tech II Year II Semester					Dept. of Computer Science and Engineering			
Code	**Category**	**Hours / Week**			**Credits**	**Marks %**		
EVA2201	Exploratory (PC)	L	T	P	C	CIE	SEE	Total
		0	0	2	1	80	20	100

Evaluation Guidelines for the course Integrated Project-II as mentioned below.

CIE/SEE	Assessment component	When	Marks
CIE	Project Report - Phase I	After the completion of 7 Weeks	40
	Project Report - Phase II	After the completion of 14 Weeks	40
SEE	Viva voce	Semester End	20
Overall	CIE assessments will be taken for 80 marks and SEE for 20 marks		