Panipat Institute of Engineering & Technology

Department of Information Technology

Syllabus

of

B.Tech. 2nd Year IT

(w.e.f. session 2025-26 onwards)

ANNEXURE-II

Third Semester

PANIPAT INSTITUTE OF ENGINEERING AND TECHNOLOGY										
An Autonomous Institute affiliated to Kurukshetra University, Kurukshetra										
B.Tech. Information Technology Semester - III										
Scheme of Studies & Examinations (w.e.f. Session 2025-26)										
Course Code	**Subject Name**	**Period (s)**			**Hours/ Week**	**Credit (s)**	**Continuous Internal Evaluation (CIE)**	**Semester End Examination (SEE)**	**Total Mark (CIE +SEE)**	**Duration of Exam**
		L	**T**	**P**						
Engineering Sciences Courses										
BT-ECE-221A	Digital Systems Design	3	0	0	3	3	40	60	100	3 hrs
BT-ECE-273A	Digital Systems Design Lab	0	0	2	2	1	50	50	100	3 hrs
Basic Sciences Courses										
ASH-MAT-211A	Discrete Structures	3	1	0	4	4	40	60	100	3 hrs
Mandatory Non-Credit Course										
BT-CE-102A	Environmental Studies	3	0	0	3	0	40	60	100	3 hrs
Professional Core Courses										
BT-IT-203A	Data Structures	3	0	0	3	3	40	60	100	3 hrs
BT-IT-205A	Object Oriented Programming	3	0	0	3	3	40	60	100	3 hrs
BT-IT-271A	Data Structures Lab	0	0	2	2	1	50	50	100	3 hrs
BT-IT-273A	Object Oriented Programming Lab	0	0	2	2	1	50	50	100	3 hrs
BT-IT-275A	Web Development Lab	0	0	2	2	1	50	50	100	3 hrs
BT-IT-291A	Summer Internship I	0	0	0	0	4	100	-	100	3 hrs
Total		**15**	**0**	**08**	**23**	**21**	**500**	**500**	**1000**	
Honors/ Minor Degree Courses										
BT-IT-251A	R programming	3	0	2	5	4	40	60	100	3 hrs
BT-IT-253A	Web Programming with Python and Java Script									
Total		**18**	**0**	**10**	**28**	**25**	**540**	**560**	**1100**	

B. Tech. Information Technology (Hons.): Approved in 2nd meeting of Academic Council held on 24.06.2025.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
ASH-MAT-211A	Discrete Structures	3	1	0	4
		CIE	SEE		Total
		40	60		100

Course Outcomes	
CO1	*Apply* the fundamental principles of set theory and propositional logic to solve problems using Venn diagrams, truth tables, and mathematical induction.
CO2	*Analyze* various types of relations and functions, *evaluate* their properties, and *solve* combinatorial problems using generating functions, recurrence relations.
CO3	*Demonstrate* understanding of graph structures and *apply* traversal algorithms to solve problems involving paths.
CO4	*Analyze* algebraic structures such as groups, monoids, and rings, and *evaluate* their properties.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO12	PSO 1	PSO 2
CO 1	**3**	**2**	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**3**	**2**
CO 2	**3**	**2**	**3**	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**3**	**3**
CO 3	**3**	**2**	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**3**	**3**
CO 4	**3**	**2**	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**3**	**2**

COURSE CIRRICULAM

Course Outline

Unit Number: 1	**Set Theory & Logic Fundamentals**	**Contact Hours: 10**
Basics of Sets, Venn Diagrams, Operations on Sets, Laws of Set Theory, Power Sets, Principle of Inclusion-Exclusion and its properties, Cartesian Products, Equivalence classes, Partition of Sets, Principle of Mathematical Induction Logic: Propositions, Logical Operations, Truth Tables, Logical Equivalence and Implication, conditional and biconditional statements Laws of Logic, Normal Forms (CNF, DNF), Predicates and Quantifiers.		
Unit Number: 2	**Relations and Functions**	**Contact Hours: 12**
Relations, Diagraphs. Paths in Relations, Properties of Relations, Equivalence Relation, Partial Order Relations, Lattice, Transitive Closure and Warshall's Algorithm. Functions: Definitions of function and its correlation with relation and its Types (Injective, Surjective, Bijective). Composition, Identity and Inverse Functions. Combinatorics: Permutations, Combinations, Pigeonhole Principle. Generating Functions and Recurrence Relations		
Unit Number: 3	**Graph Theory**	**Contact Hours: 10**
Basic terminology, types of graphs, subgraphs, representation of graphs, graph transversal algorithm: BFS and DFS, Euleraian and Hamiltonion paths, Graph colouring and planner		
Unit Number: 4	**Algebraic Structures**	**Contact Hours: 12**
Algebraic structures with one binary operation -semi groups, monoids and groups, Product and quotient of algebraic structures, Isomorphism, homomorphism, automorphism, Cyclic groups, Normal sub group, Ring homomorphism and Isomorphism.		

ANNEXURE-II

Textbook/s:

1. C.L. Liu, Elements of Discrete Mathematics, McGraw Hill, Reprinted 2000 (originally 1985).

2. B. Kolman, R.C. Busby, S. Ross, Discrete Mathematical Structures, PHI Pvt. Ltd.

Reference Books:

1. Tremblay J.P. and Manohar R., Discrete Mathematical Structures with Applications to Computer Science, McGraw Hill Book Company, 1975, International Edition 1987.

2. Ralph P. Grimaldi, Discrete and Combinatorial Mathematics, Addison-Wesley Publishing Company, Reprinted 1985.

3. Kenneth H. Rosen, Discrete Mathematics and Its Applications, McGraw Hill Book Company, 1999. (Recommended Sections: 7.1 to 7.5)

4. Satinder Bal Gupta, Discrete Mathematics and Structures, Laxmi Publications, Reprint 2013.

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

BT-ECE-221A			Digital Systems Design			BRANCH: CSE, IT	
L	**T**	**P**	**Credits**	**Semester End Examination (SEE)**	**Continuous Internal Evaluation (CIE)**	**Total Marks (CIE+SEE)**	**Time**
3	**0**	**0**	**3**	**60**	**40**	**100**	**3 Hr.**

Pre-requisites: Basics of Electronics Engineering

Course Outcomes	
CO1	Summarize the number systems and understand the basic characteristics of logic families.
CO2	Illustrate the various Boolean postulates and Minimization techniques.
CO3	Design and analysis of different type of combinational circuits.
CO4	Design and analysis of synchronous and asynchronous sequential circuits using flip flops.
CO5	Illustrate the working of various counters in digital system design.
CO6	Illustrate the working of different A/D and D/A converters.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)													
COs	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO1 0**	**PO1 1**	**PO1 2**	**PSO 1**	**PSO 2**
CO1	3	–	–	–	–	–	–	–	–	–	–	1	1	–
CO2	3	3	–	–	–	–	–	–	–	–	–	1	1	–
CO3	3	–	3	–	–	–	–	–	–	–	–	1	1	–
CO4	3	3	3	2	–	1	–	–	–	–	–	1	1	–
CO5	3	–	–	–	–	–	–	—	–	–	–	1	1	–
CO6	3	–	3	2	–	2	–	–	–	–	–	1	1	–

Unit-1 Fundamentals of Digital Systems **Contact Hours: 12**

Fundamentals of Digital Systems: Logic Gates, Universal Gates, Number systems: binary, signed binary, octal, hexadecimal number, binary arithmetic, one's and two's complements arithmetic, **Minimization Techniques:** Boolean Algebra, Standard representation of logic functions: SOP and POS forms, Simplification of switching functions using K-map and Quine-McCluskey tabular methods.

Unit-2 Combinational Circuits **Contact Hours: 10**

Combinational Digital Circuits: Design procedure: Half adder, Full Adder, Half subtractor, Full subtractor, Parallel binary adder, parallel binary Subtractor, BCD adder, Multiplexer and De-multiplexer, decoder, encoder, parity checker, parity generators, Magnitude Comparator.

Unit-3 Sequential Circuits **Contact Hours: 10**

ANNEXURE-II

Sequential circuits: A 1-bit memory, Latch, the clocked SR flip flop, J- K, T and D types flip flops, Interconversion of flipflops, applications of flip flops, shift registers, Bi-directional Shift Register, Synchronous and Asynchronous counter, Mod -N counter.

Unit-4 Converters and Memory Devices **Contact Hours: 12**

A/D and D/A Converters: Digital to analog converters: weighted resistor/converter, R-2R Ladder D/A converter, specifications for D/A converters, analog to digital converters: quantization and encoding, A/D converter, successive approximation A/D converter, Dual slope A/D converter specifications for A/D converters.

Classification of memories: ROM, RAM, PROM, EPROM, EEPROM, EAPROM, Programmable Logic Devices - Programmable Logic Array (PLA), Programmable Array Logic (PAL), Implementation of PLA, PAL using ROM.

Text Books:

1. Donald P. Leach and Albert Paul Malvino, Digital Principles and Applications, 8th Edition, TMH, 2003.M.
2. Morris Mano, Digital Design, 3rd Edition, Prentice Hall of India Pvt. Ltd., 2003 / Pearson Education (Singapore) Pvt. Ltd., New Delhi, 2003.
3. A. A. Kumar: Fundamentals of digital circuits, Prentice Hall of India.

Reference Books:

1. A.K. Maini, Digital Electronics, Wiley India
2. John F. Wakerly, Digital Design, Fourth Edition, Pearson/PHI, 2006
3. John. M Yarbrough, Digital Logic Applications and Design, Thomson Learning, 2002.
4. S. Salivahanan and S. Arivazhagan, Digital Circuits and Design, 3rd Edition., Vikas Publishing House Pvt. Ltd, New Delhi, 2006.

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

B. Tech. Information Technology (Hons.): Approved in 2nd meeting of Academic Council held on 24.06.2025.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-203A	Data Structures	3	0	0	3
		CIE	SEE		Total
		40	60		100

Course Outcomes	
CO1	Explain foundational concepts of data structures and abstract data types (ADTs), analyze algorithm complexity, and differentiate between linear and non-linear structures.
CO2	Apply appropriate searching and sorting algorithms to solve computational problems using arrays.
CO3	Implement stack and queue data structures and apply them to expression conversion and evaluation problems
CO4	Implement various types of linked lists and apply them in polynomial representation and manipulation.
CO5	Construct and traverse tree and graph structures, and apply them in expression evaluation and path-finding problems.
CO6	Assess the impact of data structure and algorithm design choices on memory utilization and computational efficiency.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PS O2
CO1	3	3	3	2	-	-	-	-	-	-	-	2	2	3
CO2	3	3	3	2	-	-	-	-	-	-	-	2	2	3
CO3	3	3	3	2	-	-	-	1	1	1	-	2	2	3
CO4	3	3	3	2	-	-	-	1	1	1	-	2	2	3
CO5	3	3	3	2	-	-	-	1	1	1	-	2	2	3
CO6	3	3	3	2	-	-	-	-	1	1	-	2	2	3

Unit 1	Introduction to Data Structures	Contact Hours: 12
Introduction to Data Structures: Introduction to Data structures & (ADTs) abstract data types, Real life applications, Categories of Data Structure: Linear and nonlinear, Introduction to algorithm complexity: time and space complexity.		

ANNEXURE-II

Arrays: Array Representation: (Storage structure), Basic Operations: traversal, insertion, deletion, search, update. Sparse Matrix and its representations. **Searching techniques:** linear search, binary search. **Sorting techniques:** selection sort, insertion sort, bubble sort, merge sort, quick sort, radix sort.		
Unit 2	**Stacks and Queues**	**Contact Hours: 10**
Stacks: Sequential implementation of stacks, representation using arrays, Basic operations: push, pop, peek. Stack Applications: Polish-notations, Expression evaluation, Evaluation of postfix expression, Infix to Postfix conversion, Memory Management. **Queues:** Definition, sequential implementation of linear queues, representation using arrays, basic operations: enqueue and dequeue; applications of queues. Circular queues: implementation using arrays, advantages over linear queues, and applications. Priority queues and their applications.		
Unit 3	**Linked Lists**	**Contact Hours: 12**
Linked Lists: Need for dynamic data structures, comparison between arrays and dynamic implementation using linked lists, linked list representation, types of linked lists, basic operations: search, insertion, and deletion. **Singly Linked Lists:** Insertion and deletion at front, end, between nodes, and after a given node. **Circular Linked Lists**: Introduction, advantages, insertion and deletion at front, end, between nodes, and after a given node.**Doubly Linked Lists:** Introduction, merits and demerits over singly linked lists, insertion and deletion at front, end, between nodes, and after a given node. **Polynomial Representation:** Representation of polynomials using arrays and linked lists.		
Unit 4	**Trees & Graphs**	**Contact Hours: 10**
Trees: Definition, basic terminology, binary tree, primitive operations on binary trees, binary tree traversals: pre-order, in-order, and post-order. Representation of binary trees in memory: array and linked list. Expression trees: representation of infix, postfix, and prefix expressions using trees. Binary Search Trees (BST): definition, insertion, deletion, and search operations. **Graphs:** Definition of undirected and directed graphs and networks, basic terminology, representation of graphs using adjacency matrix and adjacency list. Graph traversals: breadth-first search (BFS) and depth-first search (DFS). Minimum spanning trees. Hashing functions.		

Reference Books:

1. T. H. Cormen, C. E. Leiserson, R L Rivest and C Stein, “Introduction to Algorithms” 3rd Edition, MIT press, 2009.
2. R.S. Salaria, “Data Structures”, Khanna Publishing House, 2021.
3. Seymour Lipschutz, “Data Structures”, McGraw Hill Education; 1st edition, 2014.
4. R.B. Patel, “Expert Data Structures with C”, Khanna Book Publishing Company, 2020.
5. M. A. Weiss, Data Structures and Problem-Solving Using Java, Addison-Wesley, 1997.
6. M. Tannenbaum, Y Langsam and M J Augenstein, Data Structures Using C++, Prentice Hall India, 1996.
7. A. H. Aho, J. E. Hopcroft and J. Ullman, Data Structures and Algorithms, Addison-Wesley, 1987.

Online Learning Resources:

1. https://www.mygreatlearning.com/blog/data-structures-using-c/
2. https://www.scaler.com/topics/c-data-structures/
3. https://herovired.com/learning-hub/blogs/sorting-in-data-structure/
4. https://www.simplilearn.com/tutorials/data-structure-tutorial/trees-in-data-structure
5. NPTEL video series, Data-structures and Algorithms, Instructor: Naveen Garg

B. Tech. Information Technology (Hons.): Approved in 2nd meeting of Academic Council held on 24.06.2025.

ANNEXURE-II

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-205A	Object Oriented Programming	3	0	0	3
		CIE	SEE		Total
		40	60		100

Course Outcomes	
CO1	Explain fundamental concepts of Object-Oriented Programming (OOP), including classes, objects, encapsulation, inheritance, polymorphism, and abstraction.
CO2	Implement Java programs using control statements, arrays, strings, constructors, and methods.
CO3	Apply inheritance, method/constructor overloading, interfaces, and abstract classes to build extensible and maintainable Java applications.
CO4	Apply exception handling mechanisms to develop robust Java programs.
CO5	Develop and manage Java packages and perform file I/O operations using Java streams and file handling classes.
CO6	Utilize the Java Collection Framework and multithreading to develop efficient, concurrent Java applications.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PS O2
CO1	3	-	-	-	-	-	-	-	1	1	-	1	1	2
CO2	3	3	2	-	-	-	-	-	1	1	-	1	1	2
CO3	3	3	-	-	-	-	-	-	1	1	-	1	1	2
CO4	3	3	-	-	-	-	-	-	1	1	-	1	1	2
CO5	3	3	2	-	-	1	-	-	1	1	-	1	1	2
CO6	3	3	-	-	-	1	-	-	1	1	-	1	1	2

Course Outline

Unit 1	Introduction	Contact Hours: 10
Introduction to OOPs: What is OOP, Difference b/w Object Oriented, Procedural and Structural Programming language. Advantages and Features of Object-Oriented Language, OOPs Paradigm **Introduction to Java:** Introduction to Java, History of Java, JVM, The overview of Java's architecture and the architecture of the Java Virtual Machine (JVM). JRE, Java Environment, Java Source File Structure and Compilation.		

ANNEXURE-II

Programming Structures in Java: Defining Classes in Java, Accessing Classes, Constructors, Methods, Access Specifiers, Static Members, Final Members, Comments, Data types, Variables, Operators, Control Flow Statement, Arrays & String, Encapsulation.		
Unit 2	**Interface and Classes**	**Contact Hours: 12**
Extending Class and Inheritance: Using Existing Classes, Class Inheritance, Choosing Base Class, Access Attributes, Polymorphism, Multiple Levels of Inheritance, Abstraction through Abstract Classes, Using Final Modifier, The Universal Super Class-Object Class. **Polymorphism**: Method Overloading, Constructor Overloading, Dynamic Method Lookup **Interfaces:** Defining Interfaces, Abstract Method Declarations, Implementing Interfaces, Extending Interfaces, Interface References, Constants in Interfaces **Fundamental Classes:** Overview of the java. lang Package, The Object Class, The Wrapper Classes, The String Class, The StringBuilder and the String Buffer Classes.		
Unit 3	**Packages and Exception Handling**	**Contact Hours: 12**
Packages: Defining Packages, Using Packages, Compiling Code into Packages, Running Code from Packages. Scope Rules, Accessibility Modifiers, **Exception Handling:** The try Block, the catch Block, the finally Block, the throw Statement, the throws Clause, Checked and Unchecked Exceptions, Defining New Exceptions. **Input /Output Basics**: I/O Basic, Byte and Character Structure, I/O Classes, Reading Console Input, Writing to Console Output, Reading and Writing on Files, Random Access Files, Storing and Retrieving Objects from File. Stream Benefits.		
Unit 4	**Java Collection Framework and Multithreading**	**Contact Hours: 08**
Java Collections Framework: Collection in Java, Collection Framework in Java, Hierarchy of Collection Framework, Iterator Interface, Collection Interface, List Interface, ArrayList, LinkedList, Vector, Stack, Queue Interface, Set Interface, HashSet, LinkedHashSet, SortedSet Interface, TreeSet, Map Interface, HashMap Class, LinkedHashMap Class, TreeMap Class, Hashtable Class, Sorting, Comparable Interface, Comparator Interface, Properties Class in Java		

Text Book:

1. Cay Horstmann, Core Java, Volume-I Fundamentals, Pearson Education, 2024

Reference Books:

1. E. Balaguruswamy, Programming with Java: A Primer,7th Ed,2023

2. Kathy Sierra & Bert Bates, Head First Java, O'REILLY, 2nd Ed., 2005.

Online Learning Resources:

1. https://onlinecourses.nptel.ac.in/noc21_cs56

2. https://onlinecourses.nptel.ac.in/noc25_cs34

3. https://www.geeksforgCayeeks.org/java/

4. https://horstmann.com/corejava/livelessons/index.html#(1)

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required

to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-CE-102A	**Environmental Studies**	3	0	0	-
		CIE	SEE		Total
		40	60		100

Course Outcomes

CO1	Develop concepts of basic environmental factors.
CO2	Introduce to the students the basic understanding of ecosystem and its structural and functional aspects and vast biodiversity.
CO3	Outline aspects of environmental issues.
CO4	Understand the knowledge of energy resources and their environmental implications.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	-	-	-	-	2	3	-	-	-	-	-	**2**	**3**	-
CO2	2	-	-	-	-	-	3	-	-	-	-	-	**2**	**3**	-
CO3	2	-	-	-	-	3	3	2	-	-	-	-	**1**	**3**	**2**
CO4	3	-	-	-	-	2	3	-	-	-	-	2	**2**	**3**	**2**

COURSE CIRRICULAM

Course Outline

Unit Number: 1	The multidisciplinary nature of environmental studies	Contact Hours:10
Introduction to Environment: Definition, Scope, and importance of environmental studies; need for public awareness. Environmental Pollution: Definition, Cause and effects of: Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Role of an individual in prevention of pollution, Pollution case studies		
Unit Number: 2	**Natural Resources**	**Contact Hours:10**
Water resources: over-utilization, floods, drought, dams-benefits and problems; Mineral resources: Use and exploitation, environmental effects; Food resources : changes caused by modern agriculture, fertilizer-pesticide problems, water logging, Energy resources : Growing energy needs, renewable and non-renewable energy sources; Land resources : Land as a resource, land degradation, man induced landslides, soil erosion and desertification.		
Unit Number: 3	**Ecosystems and Biodiversity**	**Contact Hours:10**

ANNEXURE-II

Concept of an ecosystem, Structure and function, Energy flow, Ecological succession, ecological pyramids. Concept of Biodiversity, definition and types, Hot-spots of biodiversity; Threats to biodiversity, Endangered and endemic species of India, Conservation of biodiversity.		
Unit Number: 4	**Social Issues and the Environment**	**Contact Hours:8**
Water conservation, rain water harvesting, Environmental ethics: Issues and possible solutions. Climate change, global warming, acid rain, ozone layer depletion, Public awareness. Population growth, variation among nations, Family Welfare Programme.Human Population and the Environment - Population growth, Population explosion, Women and Child Welfare.		

Field Work - Visit to a local area to document environmental assets—river/forest/grassland/hill/ mountain. Visit to a local polluted site—Urban/Rural/Industrial/Agricultural. Study of common plants, insects, birds. Study of simple ecosystems—pond, river, hill slopes, etc (Field work equal to **5 lecture hours**)

Reference Books

- A Textbook of Environmental Studies by Asthana D.K. and Asthana Meera
- Fundamental Concepts in Environmental Studies by Mishra D.D.
- Environmental Studies by S.C Sharma M.P Poonia
- Textbook of Environmental Studies for Undergraduate by ErachBharucha
- Environmental Studies: Third Edition by R. Rajagopalan

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-271A	Data Structures Lab	0	0	2	1
		CIE	SEE		Total
		50	50		100

Course Outcomes	
CO1	Implement and manipulate arrays to perform insertion, deletion, and traversal operations.
CO2	Apply linear and binary search algorithms to locate elements in arrays.
CO3	Implement sorting algorithms (Bubble, Insertion, Selection) and analyze their time complexity.
CO4	Design and simulate stacks and queues using arrays, including overflow and underflow checks.
CO5	Construct singly linked lists and perform node insertion and deletion operations.
CO6	Solve problems using algorithm design techniques and appropriate data structures.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PS O2
CO1	3	2	2	-	1	-	-	-	1	1	-	2	2	3
CO2	3	2	2	-	1	-	-	-	1	1	-	2	2	3
CO3	3	2	3	-	1	-	-	-	1	1	-	2	2	3
CO4	3	2	3	-	1	-	-	-	1	1	-	2	2	3
CO5	3	2	3	-	1	-	-	-	1	1	-	2	2	3
CO6	3	2	3	-	1	-	-	1	1	1	-	2	2	3

Week No.	Program Description	Mapping with BL	Mapping with CO
Week 1	Traverse and Display Elements of an Array Read, print, and iterate through array elements.	2	CO1
Week 2	Find Maximum and Minimum Elements in an Array Implement algorithms to identify max/min values	3	CO1
Week 3	Replacing elements at Beginning, End, and Specified Positions in an Array	3	CO1
Week 4	Delete Elements from an Array by Position and Value Remove elements by index or value, update array size.	3	CO1
Week 5	Perform Linear Search on an Array Search for a key and return its index/position.	3	CO2
Week 6:	Implement Binary Search on a Sorted Array Apply iterative/recursive methods for sorted arrays.	4	CO2
Week 7	Sort an Array Using Bubble Sort Demonstrate pairwise swapping and complexity analysis.	4	CO3

ANNEXURE-II

Week 8	Sort an Array Using Insertion Sort Simulate "inserting" elements into sorted subarrays.	4	CO3
Week 9	Sort an Array Using Selection Sort Select and swap smallest/largest elements iteratively.	4	CO3
Week 10	Implement a Stack Using Arrays (Push/Pop Operations) Simulate LIFO behavior with overflow/underflow checks.	6	CO4
Week 11	Implement a Queue Using Arrays (Enqueue/Dequeue Operations) Simulate FIFO behavior with circular queue logic.	6	CO4
Week 12	Create a Singly Linked List and Insert Nodes at the End Dynamically allocate nodes and update pointers.	6	CO5
Week 13	Implement a Doubly Linked List Create a doubly linked list and implement operations to insert and delete nodes at the beginning, end, and a given position. Update both prev and next pointers dynamically.	6	CO5
Week 14	Binary Tree Traversals (Inorder, Preorder, Postorder) Construct a binary tree using dynamic memory allocation. Implement recursive functions to perform inorder, preorder, and postorder traversals.	3	CO6
Week 15	Graph Representation using Adjacency Matrix and List Design and implement a graph data structure. Accept a set of vertices and edges from the user and represent them using an adjacency matrix and adjacency list.	6	CO6

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-273A	Object Oriented Programming Lab	0	0	2	1
		CIE	SEE		Total
		50	50		100

Course Outcomes	
CO1	Apply basic object-oriented concepts such as classes, objects, methods, and constructors in Java programs.
CO2	Implement inheritance and interfaces to achieve code reusability and polymorphism.
CO3	Apply interfaces, constants, and interface references to develop modular and flexible Java programs.
CO4	Perform string manipulations and use wrapper classes effectively in Java applications.
CO5	Use packages, access modifiers, control structures, and arithmetic operations to structure Java programs.
CO6	Design and implement Java applications using custom exceptions and Java collections to manage data.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2
CO1	3	2	3	-	1	-	-	-	1	1	-	1	3	2
CO2	3	2	3	-	1	-	-	-	1	1	-	1	3	2
CO3	3	2	3	-	1	-	-	-	1	1	-	1	3	2
CO4	3	2	3	-	1	-	-	-	1	1	-	1	3	2
CO5	3	2	3	-	1	-	-	1	1	1	-	1	3	2
CO6	3	2	3	-	1	-	-	1	1	1	-	1	3	2

List of Experiments

Week No.	Program Description	Mapping with BL	Mapping with CO
Week 1:	Write and execute a basic program using the Java.	BL2	CO1
Week 2:	Develop a Student Information System in Java using Class and Object.	BL3	CO1
Week 3:	Create a Simple Calculator in Java using the concept of Method Overloading	BL3	CO1
Week 4	WAP for Grocery Billing System using Java Operators, Expressions, and Control Flow Statements	BL4	CO5
Week 5	Implement an Electricity Bill Calculator in Java using Inheritance.	BL4	CO2

B. Tech. Information Technology (Hons.): Approved in 2nd meeting of Academic Council held on 24.06.2025.

ANNEXURE-II

Week 6	Design a Shape Area Calculator in Java using Interfaces. (for any 2 shapes)	BL4	CO2
Week 7	Develop a Vehicle Management System in Java by Defining and Implementing Interfaces.	BL4	CO2
Week 8	Build a Bank Interest Calculator in Java using Constants in Interfaces and Interface References.	BL4	CO3
Week 9	Perform String Manipulation in Java using The String Class, StringBuilder, and String Buffer.	BL4	CO4
Week 10	Demonstrate Wrapper Class Concepts in Java using Wrapper Classes and Autoboxing/Unboxing.	BL4	CO4
Week 11	Create and Use a Java Package by Defining, Using, and Compiling Packages.	BL4	CO5
Week 12	Illustrate Access Modifiers in Java using Scope Rules, Accessibility Modifiers, and Other Modifiers for Members.	BL4	CO5
Week 13	Develop a Custom Exception Handling Mechanism in Java.	BL4	CO6
Week 14	Manage a Student List System in Java using List Interface (ArrayList, LinkedList, Vector, Stack).	BL4	CO6

Suggested Resources:

1. https://onlinecourses.nptel.ac.in/noc21_cs56

2. https://onlinecourses.nptel.ac.in/noc25_cs34

3. https://www.geeksforgeeks.org/java/

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-275A	Web Development Lab	0	0	2	1
		CIE	SEE		Total
		50	50		100

Course Outcomes	
CO1	Design and develop static web pages using HTML5 and CSS.
CO2	Develop responsive web pages using modern CSS features such as grid and flexbox.
CO3	Integrate PHP with MySQL for database operations.
CO4	Build web project with complete frontend and backend technologies.
CO5	Demonstrate communication skills through presentation and prepare well-formatted lab reports.
CO6	Follow professional ethics in web development including data privacy, security and code documentation.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PS O2
CO1	3	1	3	-	2	-	-	1	-	-	-	1	3	3
CO2	3	1	3	-	2	-	-	1	-	-	-	1	3	3
CO3	3	1	3	-	2	-	-	1	-	-	-	1	3	3
CO4	3	1	3	-	2	-	-	1	1	1	-	1	3	3
CO5	3	-	3	-	2	-	-	1	1	1	-	1	3	-
CO6	3	-	3	-	2	-	-	1	1	1	-	1	3	-

List of Experiments

Week No.	Program Description	Mapping with BL	Mapping with CO
Week 1:	Create a new document that takes the format of a business letter. Combine <P>and tags to properly separate the different parts of the documents. Such as the address, greeting, content and signature. Create a seven-item ordered list using Roman numerals. After the fifth item, increase the next list value by 5. Beginning with an ordered list, create a list that nests both an unordered list and a definition list.	3	CO1
Week 2:	Design a personal web page using HTML5 elements including images, lists, hyperlinks, and semantic tags.	3	CO1
Week 3:	Design a product showcase page for an e-commerce website using HTML tables, semantic layout, and multimedia tags.	3	CO1

ANNEXURE-II

Week 4:	Create a college admission form using HTML form elements such as input, text area, radio buttons, checkboxes, and dropdown lists.	3	CO1
Week 5:	Create frame with anchor tag. Create links in HTML with the graphics embedding.	3	CO1
Week 6:	Apply internal and external CSS to style an HTML page with custom fonts, background colors, and margins.	3	CO1
Week 7:	Create a responsive navigation bar using Flexbox and hover effects.	4	CO2
Week 8:	Design a photo gallery webpage using CSS Grid with hover zoom effects and transitions.	4	CO2
Week 9:	Create a web page with a styled form layout using CSS box model, padding, borders, and input styling.	4	CO2
Week 10:	Write a PHP program to accept two numbers from a user and perform addition, subtraction, multiplication, and division.	3	CO3
Week 11:	Create a PHP-based feedback form and display the submitted data on the same page.	3	CO3
Week 12:	Develop a login page using PHP and implement session handling for user login/logout.	4	CO3
Week 13:	Create a student registration form in PHP and store the details in a MySQL database	4	CO3
Week 14:	Develop a PHP function to calculate the factorial of a number.	3	CO3
Week 15:	Create a mini project by using front end and backend technologies.	4	CO4

Suggested Resources:

1. Web Application Development – *SWAYAM (Govt. of India)*: https://swayam.gov.in/nd2_cec20_cs09/preview
2. PHP and MySQL – *SWAYAM / Spoken Tutorial Project, IIT Bombay* https://swayam.gov.in/nd2_aic20_sp32/preview
3. W3Schools https://www.w3schools.com/
4. Tutorials Point – Web Developmenthttps://www.tutorialspoint.com/web_development_tutorials.htm
5. Coursera (Free to Audit) https://www.coursera.org/learn/html-css-javascript-for-web-developers
6. GeeksforGeeks – Web Development https://www.geeksforgeeks.org/web-development/

ANNEXURE-II

BT-ECE-273A			Digital System Design Lab				
L	**T**	**P**	**Credits**	**Semester End Examination (SEE)**	**Continuous Internal Evaluation (CIE)**	**Total Marks (CIE+SEE)**	**Time**
0	**0**	**2**	**1**	**50**	**50**	**100**	**3 Hr.**
Pre-requisites: Basic Electronics Engineering							
Course Outcomes							
CO1	Identify various digital ICs and understand their operation						
CO2	Design basic combinational Circuits and verify their functionalities.						
CO3	Apply the design procedures to design basic sequential circuits using Flip-Flops.						
CO4	Understand the concept of encoders and decoders and verify their operation.						

List of Experiments

Week 1. Familiarization with Digital Trainer Kit and associated equipment.

Week 2. Study of TTL gates AND, OR, NOT, NAND, NOR, EX-OR, EX-NOR.

Week 3. Design and realize a given function using K-Maps and verify its performance.

Week 4. To verify the operation of Multiplexer and De-multiplexer.

Week 5. To verify the operation of Comparator.

Week 6. To verify the truth table of S-R, J-K, T, D Flip-flops.

Week 7. To verify the operation of Bi-directional shift register.

Week 8. To design and verify the operation of 3-bit asynchronous counter.

Week 9. To design and verify the operation of asynchronous Up/down counter.

Week 10. To design and verify the operation of asynchronous Decade counter.

Week 11. Study of Encoder and Decoder.

Week 12. Study of BCD to 7 segment Decoder

Note: At least ten (10) experiments from the above list are mandatory to perform for the students.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-291A	Summer Internship -I	0	0	0	4
		CIE	SEE		Total
		100	-		100

Course Outcomes	
CO1	Experience the practical working environment of industry and learn to work in teams either as team leader or member.
CO2	Apply the domain knowledge to design and development of solutions in a global, economic, environmental and societal real-world problems using modern tools.
CO3	Analyze various career opportunities and decide career goals.
CO4	Prepare professional work reports using ethical practices and demonstrate strong communication skills via presentations.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO2
CO1	2	-	-	-	-	-	-	-	2	-	1	1	1	2
CO2	3	-	3	2	3	1	1	-	-	-	-	-	3	2
CO3	-	1	-	-	-	-	-	-	-	-	1	2	-	-
CO4	-	-	-	-	-	-	-	2	-	3	-	-	-	-

Course Description: This course enables students to learn technologies on industrial level. The student will be working closely with the technical team. This course enhances student's ability to think out of the box and suggest new ways of implementing ideas in a better manner and should be able to brainstorm and come up with innovative ideas.

Course Assessment: The students will be evaluated based on 4 weeks of work at industry site after second semester on the basis of following parameters:

Rubric	Parameters	Weightage	Assessment Marks
R1	Objective of Training	20%	20
R2	Domain Knowledge	20%	20
R3	Practical Implementation	20%	20
R4	Q&A during Presentation	20%	20
R5	Training Report	20%	20
Total		**100%**	**100**

ANNEXURE-II

The detailed assessment through each rubric is mentioned below:

Rubric	Parameter	Level of Achievement			
		Excellent (17-20)	**Good (13-16)**	**Average (9-12)**	**Poor (0-8)**
R1	Objective of training	Objective of training is clearly and well defined.	Objective of training is defined with good justifications.	Objective of training is defined with little justifications.	Objective of training is unclear.
R2	Domain Knowledge	Extensive knowledge of technology implemented	Fair knowledge of technology implemented	Lacks sufficient knowledge of technology implemented	No knowledge of technology implemented
R3	Practical Implementation	Practical Implementation is completed in very systematic manner.	Practical Implementation is completed in appropriate manner.	Practical Implementation is completed but not systematically.	Practical Implementation is not completed.
R4	Q&A during Presentation	Answers effectively in a satisfied manner to queries by the examiner.	Answers appropriately to queries by the examiner.	Non satisfactory answers to the queries by the examiner.	Does not answer to queries by the examiner.
R5	Training Report	Report as per specified format and completed.	Report completed with very few contents not as per format.	Report completed but formatting not done properly	Report not prepared as per format.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-251A	R Programming	3	0	2	4
		CIE	SEE		Total
		40	60		100

Course Outcomes	
CO1	**Perform** scalar and vector calculations, power operations, and integer and modulo division efficiently, **manipulate** key data structures such as vectors, matrices, lists, and factors to **analyze** and **apply** their relevance in data science, and **implement** built-in and custom functions in R to **optimize** computational tasks and **automate** processes.
CO2	**Apply** fundamental programming concepts, including conditionals, loops, sequences, and user-defined functions, to **enhance** code efficiency and reusability in R.
CO3	**Implement** various file input/output operations, import and export different data formats, and handle missing values effectively for reliable data processing.
CO4	**Apply** data manipulation techniques using dplyr and tidyr, **reshape** and **aggregate** datasets efficiently, and **demonstrate** introductory machine learning applications focused on data preparation.
CO5	**Visualize** data using base graphics, multi-panel plots, and advanced plotting techniques, and **optimize** presentations by saving and exporting visualizations in multiple formats.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO2
CO1	3	2	2	2	3	-	-	-	1	1	1	1	2	3
CO2	3	3	2	2	3	-	-	-	1	1	1	1	2	3
CO3	3	2	3	2	3	-	-	-	1	1	1	1	2	3
CO4	3	3	3	2	3	-	-	-	1	1	-	1	2	3
CO5	3	2	3	2	3				1	1	1	1	2	3

ANNEXURE-II

Unit 1	Introduction to R Programming	Contact Hours: 10
Why R? History and Applications in Data Science & Machine Learning. Installing R, RStudio, and necessary packages. Introduction to the RStudio interface, command line, data editor, **Basic Syntax**: Variables, data types, Arithmetic, relational, logical operators, vectorized operations. scalar and vector calculations, power operations, integer and modulo division, **Data Structures:** Vectors-Creating, indexing, and manipulating vectors. Matrices- Creating matrices, matrix operations (addition, multiplication, access). Lists and Factors- Creating lists, indexing, operations on factors, **Built-in Functions**: Using functions in R, writing basic functions.		
Unit 2	**Control Structures and Functions**	**Contact Hours: 12**
Conditional- If-else, nested if-else, ifelse() function, switch (). **Loops**- for loops, while loops, repeat loops. **Sequences**-Creating sequences using seq () and rep () functions **User defined functions**-writing and using user defined functions for reusable code. **String Handling and Manipulation:** String Basics- Manipulation and formatting with print(), format(), **String Operations**: Concatenation using paste(), string splitting, substitution, and replacement, **Regular Expressions**: Introduction to regular expressions for text processing. **Handling Special Characters**: Operations on alphabets, dates, and special characters.		
Unit 3	**Data Frames and Data Manipulation**	**Contact Hours: 13**
Introduction to Data Frames: Creating, indexing, and manipulating data frames, **Merging Data**: Combining and merging multiple data frames, **Data Wrangling**: Introduction to the dplyr package for filtering, selecting, and transforming data, **Factors in Data Frames**: Working with categorical data. **File I/O and Importing Data**: **Importing Data**: Reading CSV, Excel, and other formats using readr, readxl, **Exporting Data**: Writing data frames to CSV, Excel, **Handling Missing Data**: Identifying and handling missing values, **Data Cleaning**: Techniques for data cleaning, transformation, and handling outliers. **Advanced Data Manipulation: Using dplyr**-Advanced filtering, grouping, summarizing, and aggregating data, **Data Reshaping**- Using tidyr for reshaping and tidying data, **Merging Data Frames**: Using merge(), join operations, **Handling Large Datasets**: Techniques for working with large data efficiently.		
Unit 4	**Data Visualization**	**Contact Hours: 10**
Introduction, Importance, **Base Graphics in R**: Basic plots such as scatter plots, histograms, bar plots, scatter plots, box plots, histograms. **Multi-panel Plots**: Combining multiple plots using facet_wrap() and facet_grid(), **Saving and Exporting Plots**: Saving visualizations as images and PDFs.**Sequences, Sorting, and More Advanced Functions**: Sorting vectors and data frames using order (), sort (). **Applying Functions**: Using apply (), lapply (), and sapply () for efficient data manipulation, **Working with Factors**: Exploring the use of factors and their applications in data analysis.		

ANNEXURE-II

Data Frames – Combining and Importing: Data Frames: Further exploration of data frames: combining, adding, and removing columns/rows, **Data Import/Export**: Handling CSV, Excel, JSON, and other file formats using R packages (readr, readxl, jsonlite), **Data Transformation**: Efficiently transforming data using dplyr and tidyr (reshaping, pivoting), **Introduction to ML Applications**: Basic introduction to the use of R for machine learning (focus on data preparation).

List of Experiments:

1. Download and install R-Programming environment and install basic packages using install. Packages () command in R.
2. Learn all the basics of R-Programming (Data types,Variables , Operators etc.)
3. Implement R-Loops with different examples.
4. Learn the basics of functions in R and implement with examples.
5. Implement data frames in R. Write a program to join columns and rows in a data frame using c bind () and r bind () in R.
6. Implement different String Manipulation functions in R.
7. Implement different data structures in R (Vectors, Lists, Data Frames)
8. Write a program to read acsv file and analyze the data in the file in R
9. Create pie charts and bar charts using R.
10. Create a data set and do statistical analysis on the data using R.
11. Write R program to find Correlation and Covariance
12. Write R program for Regression Modeling
13. Write R program to build classification model using KNN algorithm
14. Write R program to build clustering model using K-mean algorithm

Reference Books:

1. Peng, R. D. (2016). R programming for data science (pp. 86-181). Victoria, BC,Canada: Leanpub.
2. Boehmke, B. & Brandon, G. (2020). Handson Machine Learning with R, CRCPress.
3. Horton, N.J. & Kleinman, K. (2015) Using R & R Studio for DataManagement, StatisticalAnalysis , and Graphics, CRC Press.

Online Learning Resources:

1. NPTEL video series, Data Science with R by Prof Salabh.
2. NPTEL video series, Introduction to R by Prof Salabh.

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

B. Tech. Information Technology (Hons.): Approved in 2nd meeting of Academic Council held on 24.06.2025.

ANNEXURE-II

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-253A	Web Programming with Python & Javascript	3	0	2	4
		CIE	SEE		Total
		40	60		100

Course Outcomes	
CO1	**Understand** fundamental web development principles, including web standards, accessibility, and client-server architecture, and **apply** responsive design techniques using modern CSS frameworks like Flexbox, Grid, and Bootstrap.
CO2	**Utilize** version control systems such as Git and GitHub to **collaborate** effectively on projects, **manage** source code, and **deploy** web applications efficiently.
CO3	**Develop** dynamic web applications using Python frameworks (Flask/Django), **implement** routing, HTTP methods, and template rendering, and **integrate** databases using PostgreSQL, SQLite, and ORM technologies.
CO4	**Apply** JavaScript programming concepts for **DOM manipulation**, **event handling**, and **API integration**, while **designing** modern front-end interfaces using React.js and Vue.js.
CO5	**Implement** AJAX, Fetch API, and WebSockets to **enable** real-time data communication and **create** interactive single-page applications with seamless user experiences.
CO6	**Deploy** web applications using platforms like Netlify, Heroku, and AWS, **containerize** environments using Docker, **automate** CI/CD workflows with Jenkins and Maven, and **optimize** performance through load balancing and caching strategies.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO2
CO1	3	2	3	2	3	1	-	-	1	1	-	1	3	3
CO2	3	3	3	2	3	1	-	-	1	1	1	1	3	3
CO3	3	3	3	3	2	-	-	-	-	1	1	1	3	3
CO4	3	3	3	2	3	1	-	-	1	1	1	1	3	3
CO5	3	3	3	3	3	-	-	-	1	1	1	1	3	3
CO6	3	3	3	3	3	-	-	-	1	1	1	1	3	3

ANNEXURE-II

Unit 1	Foundations of Web Development	Contact Hours: 10
Introduction to Web Development: Web standards and accessibility, Understanding client-server architecture **Responsive Web Design:** CSS fundamentals (Selectors, Box model), CSS Grid, Flexbox, Bootstrap, Media queries for different screen sizes **Version Control & Collaboration:** Overview of version control systems – role of version control systems – Types of control systems and their supporting tools – Overview of Git ((commits, branches), GitHub workflow (pull requests, collaboration), Using Git for team projects) – Overview of Source code and Version Control hosts – Deploy the files to GitHub. **Setting Up a Web Development Environment:** Choosing the right IDE (VS Code, PyCharm), Browser dev tools for debugging, Package management (npm, pip). Introduction to Web Applications, Scrapping the Web		
Unit 2	**Backend Development with Python**	**Contact Hours: 12**
Python for Web Development: Flask vs. Django: Choosing the right framework, Setting up a Flask/Django project, Handling requests and responses **Routing and Views:** URL patterns and dynamic routing, Handling HTTP methods (GET, POST, PUT, DELETE), Template rendering with Jinja2 (Flask) / Django Templates **Databases and ORM:** SQL basics and database design, Using PostgreSQL and SQLite, Django ORM vs. SQL Alchemy		
Unit 3	**Frontend Development & JavaScript**	**Contact Hours: 13**
Introduction to JavaScript, Events, Variables, querySelector, DOM Manipulation: JavaScript Console, Arrow Functions, TODO List; Intervals, Local Storage, APIs: JavaScript Objects, Currency Exchange. Introduction to User Interfaces, Single Page Applications, scroll: Infinite Scroll; Animation, React: Addition, DOM manipulation and event handling, Java Script Units and imports. Introduction to React.js and Vue.js, Component-based Architecture, Using hooks and lifecycle methods AJAX and Fetch API for dynamic content, Ajax and Fetch API for dynamic content, Web Sockets for real-time applications		
Unit 4	**Deployment & Full-Stack Project**	**Contact Hours: 10**
Web Application Deployment: Deployment platforms (Netlify, Heroku, AWS), Docker for containerization, CI/CD automation (Importance of continuous Integration, Overview and Features of Jenkins, set up Jenkins, Overview and Features of Maven, - Setup Maven), Load balancing and caching strategies, writing unit and integration tests, Debugging and profiling		

ANNEXURE-II

tools **Final Project: Stack Web Application:** Idea selection and wireframing, Backend and frontend integration, Deployment and presentation

List of Experiments

1. Perform the following actions using Git / GiTHub
   a. Create a new repository in GitHub
   b. Clone the repository
   c. Git branches
   d. Check the status of a Git Project
   e. Use commit, view to the Gitt
2. Perform the following actions using Git / GiTHub
   a. Feature branches and collaborations
   b. Merge a branch
   c. Pull Request
   d. Retrieval
   e. Fixing merge conflicts
3. Create a personalized job board by filtering jobs based on keywords, location, or salary. You can scrape job postings from platforms like Linkedin, Indeed or Naukri. Use of selenium (for dynamic content) and BeautifulSoup is recommended for the above exercise.
4. The following experiments are based on FLASK
   A. A. Use Flask's Flask-RESTful extension to create a simple REST API to do (CRUD OPERATIONS)
   B. Implement user authentication using Flask-Login. Add features like registration, login, logout, and password hashing
   C. Connect Flask to a database (e.g., SQLite or PostgreSQL) using SQLAlchemy. Build a small blog application with models for users, posts, and comments.
   D. Build a real-time chat app using Flask-SocketIO. Implement rooms and user-specific messaging.
5. Develop a microservice with Flask that communicates with other services via REST APIs. Experiment with Docker to containerize the service.
6. The following experiments are based Django
   A. Build a basic e-commerce site with product listings, a shopping cart, and checkout functionality.
   B. Create a custom user model to extend Django's default authentication system. Add fields like profile picture, bio, and social media links.
   C. Develop a REST API for a blog or social media app. Add features like pagination, filtering, and authentication (JWT or OAuth)
   D. Use Django signals to trigger actions (e.g., sending a welcome email when a user registers). Experiment with custom signals for specific use cases.

ANNEXURE-II

The following experiment needs to be implemented in JavaScript

7. Write a program to handle user interactions and manipulate the DOM dynamically. Use the following steps
   a. Create an HTML page with buttons and input fields.
   b. Use `querySelector` to select elements.
   c. Add event listeners (`click`, `keyup`, etc.) to modify the page dynamically
   d. Display real-time changes using JavaScript Console logs.
8. Implement a simple TODO list that persists data using Local Storage. Use the following steps
   a. Create an input field and a button to add tasks.
   b. Store tasks in Local Storage using localStorage.setItem().
   c. Retrieve tasks using localStorage.getItem() and display them.
   d. Allow users to delete tasks and update Local Storage accordingly.
9. Implement infinite scrolling and basic animations in a webpage. Use the following tasks
   a. Create a long list of items in HTML.
   b. Use JavaScript to detect when the user reaches the bottom of the page.
   c. Dynamically load more content using fetch ().
   d. Apply CSS animations to elements appearing on scroll.
10. Build a simple React app using components and hooks. Implement following tasks
    a. Set up a React project using create-react-app.
    b. Create functional components with useState and useEffect.
    c. Implement event handling within components.
    d. Display dynamic content based on user interactions.
11. Fetch data from an API and display it dynamically. Implement following tasks
    a. Use fetch () to retrieve data from a public API (e.g., currency exchange rates).
    b. Parse the JSON response and display it on the webpage.
    c. Implement error handling for failed requests.
    d. Refresh data periodically using setInterval ().
12. Implement real-time communication using WebSockets completing following task
    a. Set up a WebSocket server (Node.js with ws package).
    b. Create a client-side WebSocket connection.
    c. Send and receive messages dynamically.

d. Display real-time updates in a chat-like interface.

13. Containerize a simple web application using Docker.
14. Automate the build and deployment process using Jenkins and Maven. Execute the following steps:
    a. Install Jenkins and set up a new pipeline.
    b. Install the Maven plugin in Jenkins.
    c. Create a simple Java application with a pom.xml file for Maven.
    d. Configure Jenkins to pull code from a Git repository.
    e. Set up a Jenkins job to build the project using Maven (mvn clean install).
    f. Automate deployment using Docker containers.
    g. Verify the successful execution of the pipeline.
15. Complete a project of your choice and make a presentation in the class.

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Fourth Semester

PANIPAT INSTITUTE OF ENGINEERING AND TECHNOLOGY										
An Autonomous Institute affiliated to Kurukshetra University, Kurukshetra										
B.Tech. Information Technology Semester - IV										
Scheme of Studies & Examinations (w.e.f. Session 2025-26)										
Course Code	**Subject Name**	**Period (s)**			**Hours/Week**	**Credit (s)**	**Continuous Internal Evaluation (CIE)**	**Semester End Examination (SEE)**	**Total Marks (CIE+ SEE)**	**Duration of Exam**
		L	**T**	**P**						
Engineering Sciences Courses										
BT-EC-202A	Microprocessors Based System Design	3	0	0	3	3	40	60	100	3 hrs
BT-EC-272A	Microprocessors Based System Design Lab	0	0	2	2	1	50	50	100	3 hrs
Mandatory Non-Credit Course										
ASH-HUM-210A	Indian Knowledge System: Concepts in Engineering	2	0	0	2	2	40	60	100	3 hrs
Professional Core Courses										
BT-IT-202A	Operating Systems	3	0	0	3	3	40	60	100	3 hrs
BT-IT-204A	Design & Analysis of Algorithms	3	0	0	3	3	40	60	100	3 hrs
BT-IT-206A	Database Management Systems	3	0	0	3	3	40	60	100	3 hrs
BT-IT-208A	Computer Organization and Architecture	3	0	0	3	3	40	60	100	3 hrs
BT-IT-272A	Operating Systems Lab	0	0	2	2	1	50	50	100	3 hrs
BT-IT-274A	Design & Analysis of Algorithms Lab	0	0	2	2	1	50	50	100	3 hrs
BT-IT-276A	Database Management Systems Lab	0	0	2	2	1	50	50	100	3 hrs
Total		**17**	**0**	**8**	**25**	**21**	**440**	**560**	**1000**	
Honors Degree Courses										
BT-IT-252A	Foundations of Data Analytics, Machine Learning and Statistics	3	0	2	5	4	40	60	100	3 hrs
BT-IT-254A	Server-Side Programming									
Total		**20**	**0**	**10**	**30**	**25**	**480**	**620**	**1100**	

Minors in Cloud Technology										
BT-IT-256A	Linux & Cloud Technology	4	0	0	4	4	40	60	100	3 hrs
BT-IT-278A	Linux & Cloud Technology Lab	0	0	2	2	1	50	50	100	3 hrs

B. Tech. Information Technology (Hons.): Approved in 2nd meeting of Academic Council held on 24.06.2025.

ANNEXURE-II

BT-ECE-202A			Microprocessors Based System Design				
L	**T**	**P**	**Credits**	**Semester End Examination (SEE)**	**Continuous Internal Evaluation (CIE)**	**Total Marks (CIE+SEE)**	**Time**
3	**0**	**0**	**3**	**60**	**40**	**100**	**3 Hr.**

Pre-requisites: Basics of digital electronics and computer architecture.

Course Outcomes

CO1	To understand the detailed architecture of microprocessor 8086
CO2	To implement the interfacing of memories to 8086 Microprocessor and study of interrupt
CO3	To classify and explain the instruction set and programming concepts in assembly language.
CO4	To apply the programming tools to Interface peripherals with microprocessor.

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2
CO1	3	3	-	-	-	-	-	-	-	-	-	-	2	-
CO2	3	3	2	3	-	-	-	-	-	-	-	1	2	-
CO3	3	2	-	-	2	-	-	-	2	-	-	-	3	-
CO4	3	2	2	3	2	-	-	-	2	-	-	1	3	-

Unit-1 8086 CPU ARCHITECTURE **Contact Hours: 11**

Introduction to basic computer architecture, Functional units, Basic operational concepts, Harvard architecture, Von-Neumann Architecture, 8086 Block diagram; description of data registers, address registers; pointer and index registers, PSW, Queue, BIU and EU. 8086 Pin diagram descriptions. Generating 8086 CLK and reset signals using 8284. WAIT state generation. Microprocessor BUS types and buffering techniques, 8086 minimum mode and maximum mode CPU module.

Unit-2 MAIN MEMORY SYSTEM DESIGN **Contact Hours: 06**

Memory devices, 8086 CPU Read/Write timing diagrams in minimum mode and maximum mode. Units of memory capacity, concept of memory banking, Address decoding techniques. Interfacing different configurations of SRAMS, ROMS/PROMS with 8086 based CPU module. Interfacing and refreshing DRAMS.

Interrupts: 8086 Interrupt mechanism; interrupt types and interrupt vector table. Applications of interrupts

Unit-3 8086 INSTRUCTION SET AND PROGRAMMING TECHNIQUES **Contact Hours: 07**

Instruction formats, addressing modes, Data transfer instructions, string instructions, logical instructions,

arithmetic instructions, transfer of control instructions; process control instructions; Assembler directives. Writing assembly Language programs for logical processing, arithmetic processing, timing delays; loops, data conversions

Unit-4 BASIC I/O INTERFACE **Contact Hours: 12**

I/O Intel's 8255 - description and interfacing with 8086. Interfacing of 0808/0809 ADCs and 0800 DACs, Generation of different waveforms. Interfacing of stepper motor, Interfacing LED displays, Interfacing Keyboards, optical encoder with 8086. Universal synchronous asynchronous receiver transmitter- 8251: descripting and interfacing with 8086. Intel's 8259. DMA operation, Intel's 8237.

Text Books:

1. D.V. Hall, Microprocessors and Interfacing, McGraw Hill 2nd ed.
2. Barry B. Brey, "The Intel Microprocessor8086/8088, 80186", Pearson Education, Eighth Edition, 2009
3. Morris Mano, "Computer System Architecture", PHI.

Reference Books:

1. Liu, Gibson, "Microcomputer Systems: The 8086/88 Family", 2nd Edition, PHI,2005
2. Kenneth Ayala, "The 8086 Microprocessor: Programming & Interfacing the PC", Cengage Learning, Indian Edition, 2008
3. Kip Irvine, "Assembly language for IBM PC", PHI, 2nd Edition, 1993
4. Peter Abel, "Assembly language programming", Pearson Edu,5th Edition,2002

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-202A	Operating Systems	3	0	0	3
		CIE	SEE		Total
		40	60		100

Course Outcomes

CO1	**Understand** operating system architecture, evolution, and **implement** process management techniques, including scheduling algorithms and threading models, to optimize system efficiency.
CO2	**Implement** synchronization mechanisms such as semaphores and monitors, **design** solutions for concurrency control, and **develop** strategies for deadlock prevention and recovery.
CO3	**Describe and Implement** memory management concepts, including paging, segmentation, and virtual memory, and **enhance** disk performance through optimized scheduling algorithms.
CO4	**Develop** structured file systems using efficient allocation methods, directory structures, and security protocols while **integrating** data protection mechanisms.
CO5	**Evaluate** security threats, authentication methods, and intrusion detection techniques to **reinforce** operating system security and resilience.
CO6	**Utilize** system calls for file operations, process control, and access management to **demonstrate** real-world OS functionalities in practical implementations.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO1	3	-	-	-	-	-	-	-	-	-	-		2	-
CO2	3	3	-	-	-	-	-	-	-	-	-	1	2	-
CO3	3	3	2	-	-	-	-	-	-	-	-	1	2	-
CO4	3	3	-	-	-	1	-	-	-	1	-	-	2	-
CO5	3	3	2	-	-	-	-	1	-	-	-	1	2	-
CO6	3	3	-	-	-	-	-	-	1	-	-	1	2	-

Unit 1	Introduction to Operating System	Contact Hours: 12
Operating system: Organization, abstraction provided by OS, features and roles, OS evolution, Operating system architecture, OS examples; Process management: Process control block, system calls and interrupts, context switching, schedular and dispatcher, process states and life cycle, CPU scheduling algorithms: FCFS, SJF, STRF, priority, round robin, multilevel queue and feedback scheduling. Threads: multithreading, kernel vs. user level threads, process vs. thread; Case Study: fork, wait, exec, exit, kill, getpid		
Unit 2	**Deadlocks & Synchronization**	**Contact Hours: 10**

ANNEXURE-II

Synchronization: Principle of Concurrency, Critical Section problem, Classical two process and n process solution, Semaphores, Hardware Synchronization, Monitor, Classical problems of synchronization, Inter Process Communication. Deadlock: Conditions, Modelling, detection, recovery, avoidance and deadlock prevention. Case Study: signals, pipe		
Unit 3	**Memory Management & Disk Management**	**Contact Hours: 10**
Memory Management: background, logical vs. physical address space, contiguous memory allocation, paging, Implementation issues in paging such as page tables, inverted page tables, segmentation, segmentation with paging. Virtual Memory: background, demand paging, concept of page replacement, page replacement algorithms, allocation of frames, thrashing. Disk Management: disk structure, disk scheduling (FCFS, SSTF, SCAN, C-SCAN), disk reliability, disk Performance parameters		
Unit 4	**File Systems & Security**	**Contact Hours: 10**
File Systems: File Type, attributes, access and security, file operations, file organization and access methods, allocation methods, directory structure, directory operations, Implementation of directories, free- space management, logical file system, physical file system Protection and Security: Goals of protection and security, security attacks, authentication, program threats, system threats, Securing system and facilities, Intrusion Detection: Auditing and logging, Tripwire, System call monitoring; Case Study: open, close, read, write, lseek, stat, sync, mkdir, rmdir, link, unlink, mount, umount users + Security: chown, chmod, getuid, setuid		

Textbook/s:

1. Operating System Concepts – Silberschatz, Galvin, Gagne
2. Modern Operating Systems – Andrew S. Tanenbaum, Herbert Bos

Reference Books:

1. Peterson, J.L. & Silberschatz, A.: Operating System Concepts, Addison, Wesley-Reading.
2. Brinch, Hansen: Operating System Principles, Prentice Hall of India.
3. Haberman, A.N.: Introduction to Operating System Design Galgotia Publication, New Delhi.
4. Tanenbaum, A.S.: Operating Systems.
5. Hansen, P.B.: Architecture of Concurrent Programs, PHI.
6. Shaw, A.C.: Logic Design of Operating Systems, PHI.

Online Learning Resources:

1. https://nptel.ac.in/courses/106/108/106108101/
2. https://www.coursera.org/learn/os-power-user

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required

to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-204A	Design & Analysis of Algorithms	3	0	0	3
		CIE	SEE		Total
		40	60		100

Course Outcomes	
CO1	Define and articulate the fundamental principles of algorithms, including their characteristics, performance metrics, and complexity measures. Apply asymptotic notations to rigorously analyze both time and space complexities.
CO2	Formulate recurrence relations for recursive algorithms and employ mathematical techniques to solve them, thereby deriving precise time complexity bounds.
CO3	Design, implement, and critically analyze advanced data structures and graph representations. Utilize graph traversal algorithms to address complex algorithmic challenges effectively.
CO4	Develop efficient algorithmic solutions based on the divide-and-conquer paradigm, emphasizing systematic problem decomposition and solution synthesis.
CO5	Construct algorithmic solutions for optimization problems using greedy and dynamic programming strategies, and evaluate the trade-offs in efficiency and performance.
CO6	Devise solutions for combinatorial and optimization problems through backtracking and branch-and-bound methods, and assess their performance, correctness, and computational feasibility.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO2
CO1	3	3	-	-	-	-	-	-	-	-	-	1	3	3
CO2	3	3	-	3	-	-	-	-	-	-	-	1	3	-
CO3	3	3	3	-	-	-	-	-	-	-	-	1	3	3
CO4	3	3	3	3	-	1	-	-	-	-	-	1	3	3
CO5	3	3	3	3	-	-	-	-	-	-	-	1	3	3
CO6	3	3	3	3	-	-	-	-	1	-	-	1	3	3

Unit 1	Introduction to Algorithm Analysis	Contact Hours: 08
Overview of Algorithms- Characteristics of Algorithms, Criteria for Analyzing Algorithms; Complexity Measures- Time and Space Complexity, Best, Worst, and Average-Case Complexities; Asymptotic Notations & Their Properties- O, Ω, Θ, o, ω, comparing growth rates of standard functions (e.g., polynomials, exponentials, logarithms); Recurrence Relations & Analysis of Recursive Algorithms- Formulating recurrence equations, Methods to solve recurrences-Substitution, Recursion Tree, Master's Theorem.		
Unit 2	**Advanced Data Structures & Divide-and-Conquer**	**Contact Hours: 12**
Advanced Data Structures: Self-Balancing Trees: AVL Trees, Red-Black Trees, B-Trees, B+ Trees- Insertion and Deletion (including rotations/balancing steps); Heaps and Heap Sort; Disjoint Set (Union-		

ANNEXURE-II

Find); Graph Traversals: Fundamental Graph Representations (Adjacency list, Adjacency matrix), breadth-first search (BFS) and depth-first search (DFS), Topological Sorting. Divide-and-Conquer Paradigm: Binary Search, Merge Sort, Quick Sort, Strassen's Algorithm for Matrix Multiplication.		
Unit 3	**Greedy Strategy & Dynamic Programming**	**Contact Hours: 12**
Greedy Algorithms: Greedy Paradigm, Fractional Knapsack, Interval Scheduling / Job Scheduling with Deadlines, Minimum cost spanning trees, Huffman Coding, Single-Source Shortest Path (for Nonnegative Edges)- Dijkstra's Algorithm; Handling Negative Edge Weights-Bellman-Ford Algorithm (Single-Source Shortest Path with negative edges, no negative cycle). Dynamic Programming: Introduction: Memoization vs. Tabulation, Distinguishing DP from Greedy & Divide-and-Conquer, Classic DP Problems: 0–1 Knapsack, Matrix Chain Multiplication, All-Pairs Shortest Path: Floyd–Warshall Algorithm, Longest Common Subsequence (LCS), Traveling Salesman Problem (TSP).		
Unit 4	**Backtracking & Branch-and-Bound**	**Contact Hours: 10**
Backtracking: Overview and State-Space Trees, N–Queen Problem, Subset/Knapsack Variants (Backtracking approach), Graph problems: Hamiltonian Cycle, Graph Coloring. Branch-and-Bound: LC (Least-Cost) Searching and Bounding Functions, FIFO Branch-and-Bound vs. LC Branch-and-Bound, Applications: 0–1 Knapsack Problem (optimization via bounding), Traveling Salesman Problem (TSP).		

Text Book:

1. Coremann, Leiserson, Rivest, Stein, Introduction to Algorithms, PHI

Reference Books:

1. S. Sridhar , "Design and Analysis of Algorithms", Oxford University Press, 2014.
2. Aho, Hopcroft, and Ullman, "The Design and Analysis of Computer Algorithms", Addison Wesley.
3. M. Tenenbaum, Augestien, "Data Structures using C", Third Edition, Pearson Education, 2007
4. Michael T. Goodrich, Roberto Tamassia, "Algorithm Design", First Edition, Wiley, 2006.

Online Learning Resources:

1. https://www.geeksforgeeks.org/analysis-of-algorithms/
1. https://www.tutorialspoint.com/design_and_analysis_of_algorithms/index.htm
2. https://www.youtube.com/playlist?list=PLEBuowGoCtr0nMAlx8obo8x_Pgle1HvSv
3. https://www.youtube.com/playlist?list=PLxCzCOWd7aiHcmS4i14bI0VrMbZTUvlTa
4. https://onlinecourses.nptel.ac.in/noc25_cs33/preview
5. https://onlinecourses.nptel.ac.in/noc25_cs23/preview

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

B. Tech. Information Technology (Hons.): Approved in 2nd meeting of Academic Council held on 24.06.2025.

ANNEXURE-II

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-206A	DATABASE MANAGEMENT SYSTEMS	3	0	0	3
		CIE	SEE		Total
		40	60		100

Course Outcomes

CO1	Understand the evolution and architectural design of database management systems. Construct conceptual models by developing ER/EER diagrams using StarUML to demonstrate a comprehensive understanding of database architectures.
CO2	Design and develop relational databases using SQL and PostgreSQL. This includes defining schemas, writing complex queries, and performing complete CRUD operations to manage data effectively.
CO3	Identify and enforce functional dependencies and apply normalization techniques to optimize database designs. Evaluate various normalization forms to ensure data consistency and eliminate redundancy in schema development.
CO4	Design and implement robust transaction management strategies by employing ACID properties and serializability concepts. Develop solutions to handle transactions effectively, ensuring consistency and reliability in database operations.
CO5	Examine and implement concurrency control mechanisms—such as lock-based, timestamp-based, and optimistic protocols—to manage multiple simultaneous transactions. Develop strategies for detecting, preventing, and resolving deadlocks, ensuring smooth operations.
CO6	Analyze strategies for managing unstructured data and implement NoSQL solutions using MongoDB. Utilize information retrieval techniques to query, rank, and manage unstructured text data efficiently.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2
CO1	3	2	-	-	-	-	-	-	-	-	-	1	2	-
CO2	3	2	-	-	2	-	-	-	1	-	-	1	2	3
CO3	3	2	-	-	-	-	-	-	-	-	-	1	2	-
CO4	3	2	-	-	-	1	-	-	-	1	-	1	2	-
CO5	3	-	-	-	-	-	-	1	-	-	-	1	2	-
CO6	3	-	-	-	2	-	-	-	-	-	-	1	2	3

Unit 1	Introduction to Database	Contact Hours: 8
Data, Database, Database management system, Historical background from file system to database management system, Data models, Schemas, Instances, Database Users, DBMS architecture, ER Model and EER Model along with its implementation in StarUML.		
Unit 2	**Query Languages and Database Design**	**Contact Hours: 15**

ANNEXURE-II

Query Languages: Keys, Integrity Constraint, Relational Algebra: Selection, Projection, Union, Intersection, Set Difference, Cartesian Product, Join, Division. Relational calculus: Domain relational calculus and Tuple relational calculus, SQL introduction, Query writing in SQL using DDL, DML, DCL and TCL operators., Postgre introduction, Query writing of CRUD operations in PostgreSql. Database Design: Functional Dependencies, Dependency preservation and Lossless and Lossy decomposition, Database Anomalies, Normalization and its types.		
Unit 3	**Transaction Processing and Deadlock Handling**	**Contact Hours: 11**
Transaction Processing: Transaction Concept, State diagram, ACID properties, Serializability: Conflict and View Serializability, Need for Concurrency, Lock based protocols, Timestamp-based protocols, Validation-based protocols, Shadow Paging. Deadlock: Deadlock, Deadlock handling and Starvation.		
Unit 4	**Unstructured Data Management and NO SQL Databases**	**Contact Hours: 11**
Unstructured Data management: Unstructured text, Information retrieval systems, document retrieval and ranking. NoSQL Database: Introduction to MongoDB, Data Types in MongoDB, CRUD operations, Comparison and Logical Operators, Querying Arrays.		

Textbook/s:

1. Korth, Silberschatz, "Database System Concepts", 4th, 5th or 6th Ed.Publisher
2. Steve Bobrowski, "Oracle & Architecture", TMH, 2000
3. MongoDB Fundamentals: A Hands-on Guide to Using MongoDB and Atlas in the Real World Book by Juned Ahsan, Liviu Nedov, and Michael Harrison.Publisher
4. Beginning Databases with PostgreSQL, 2nd Edition Author: Neil Matthew, Richard Stone

Reference Books:

1. Date C. J., "An Introduction to Database Systems", 7th Ed., Narosa Publishing, 2004
2. Elmsari and Navathe, "Fundamentals of Database Systems", 4th Ed., A. Wesley, 2004
3. Ullman J. D., "Principles of Database Systems", 2nd Ed., Galgotia Publications, 1999.

Online Learning Resources:

1. https://www.youtube.com/watch?v=rbwXdTsCk2c&list=PLyvBGMFYV3auVdxQ1-88ivNFpmUEy-U3M
2. https://www.youtube.com/watch?v=sSmz9DQsvDM
3. https://ocw.mit.edu/courses/6-830-database-systems-fall-2010/

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

B. Tech. Information Technology (Hons.): Approved in 2nd meeting of Academic Council held on 24.06.2025.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-208A	Computer Organization and Architecture	3	0	0	3
		CIE	SEE		Total
		40	60		100

Course Outcomes	
CO1	Define and evaluate the Von Neumann architecture and classify computer systems according to Flynn's taxonomy. Critically analyze the layered structure of computer systems—including the interrelations among digital logic, microarchitecture, Instruction Set Architecture (ISA), and high-level language abstractions—to reveal their design rationale.
CO2	Identify and describe the core components of a computer system, including the CPU, memory, and I/O subsystems. Construct a detailed illustration of the instruction cycle and design a logic model for an accumulator-based execution system that integrates instruction timing and control mechanisms.
CO3	Apply the Register Transfer Language (RTL) to model and demonstrate register transfers and micro-operations. Design and implement a microprogrammed control unit that incorporates control memory organization, address sequencing, and microprogram sequencing to achieve effective control flow.
CO4	Evaluate and compare diverse CPU architectures—including accumulator, register, stack, and memory/register models—and different addressing modes. Analyze the design trade-offs and performance implications of RISC versus CISC architectures to determine their suitability for various applications.
CO5	Describe the structure and organization of the memory hierarchy, differentiating among various memory types such as semiconductor RAM/ROM, cache, auxiliary memory, and others. Critically analyze key memory management mechanisms including caching policies, virtual memory strategies, Direct Memory Access (DMA), and I/O Processor (IOP) approaches
CO6	Assess system performance parameters with an emphasis on pipelining, pipeline hazards, and memory strategies. Recommend and design performance optimizations by applying principles like instruction-level parallelism and Amdahl's Law to enhance overall system efficiency.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PS O2
CO1	3	3	-	-	-	-	-	-	-	1	-	1	2	-
CO2	3	-	2	-	-	-	-	-	-	1	-	1	2	-
CO3	3	-	-	-	2	-	-	-	-	1	-	1	2	-
CO4	3	3	-	-	-	-	-	-	-	1	-	1	2	-
CO5	3	3	-	1	-	-	-	-	-	1	-	1	2	-
CO6	3	2	2	-	-	-	-	-	-	1	-	1	3	-

Unit 1	Basic Computer Organizations and Design	Contact Hours: 10

General System Architecture: Von-Neumann Model, Store program control concept, Flynn's classification of computers (SISD, MIMD); Multi level viewpoint of a machine: digital logic, micro-architecture, ISA, operating systems, high level language

Structured organization: CPU, caches, main memory, secondary memory units & I/O, Instruction Codes, Computer registers, Computer Instructions, Timing and Control, Instruction Cycle, Type of Instructions:

Memory reference instructions, Register reference instructions, Input output instructions Design of accumulator logic,Register Transfer Language (RTL),		
Unit 2	**Register Transfer and Microoperations**	**Contact Hours: 10**
Register transfer, Bus and Memory Transfers, Arithmetic Microoperations, Logic Microoperations, Shift Micro Operations, Arithmetic Logic Shift Unit, Microprogrammed Control: Control memory; address sequencing, microprogram sequencer, Amdahl's Law Design of Control Unit		
Unit 3	**Central Processing Unit:**	**Contact Hours: 11**
CPU Architecture types (accumulator, register, stack, memory/register), Instruction formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Program Interrupt, RISC, CISC, Pipelining: pipeline hazards, Design issues of pipeline architecture, Instruction level parallelism and advanced issues.		
Unit 4	**Memory Organization**	**Contact Hours: 11**
Memory hierarchy, Main memory (Semiconductor RAM & ROM organization, memory expansion, Static & dynamic memory types); Cache memory replacement policy, Associative & direct mapped cache organizations. Auxiliary Memory, Associative Memory, Cache memory, Virtual Memory, Interrupt Driven I/O,Direct Memory Access (DMA), Input-Output Processor (IOP).		

Reference Books:

1. Wakerly John, Digital Design: Principles and Practice (5 ed.), Prentice Hall, 2017. ISBN 9780134460093.

2. David A. Patterson, John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, 5th Edition - 2014, Morgan Kaufmann Publishers.

3. Andrew S. Tanenbaum, Structured Computer Organization, 6th Edition - 2013, Pearson Education.

4. John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, 6th Edition - 2017, Morgan Kaufmann Publishers.

Online Learning Resources:

1. NPTEL, Computer Organization and Architecture, [Online Course], Indian Institute of Technology (IIT), Available at: https://nptel.ac.in/courses/106105163
2. MIT OpenCourseWare, Computation Structures (6.004), [Online Course], Massachusetts Institute of Technology, Available at: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-004-computation-structures-spring-2017/
3. Coursera, Computer Architecture by Princeton University, [Online Course], Available at: https://www.coursera.org/learn/comparch
4. YouTube - Gate Smashers, Computer Organization and Architecture Lectures, [Online Videos], Available at: https://www.youtube.com/c/GateSmasher
5. GeeksforGeeks, Computer Organization and Architecture Tutorials, [Online Resource], Available at: https://www.geeksforgeeks.org/computer-organization-and-architecture-tutorials/

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required

to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
ASH-HUM-210A	INDIAN KNOWLEDGE SYSTEM: CONCEPTS IN ENGINEERING	2	-	-	2
		CIE	SEE		Total
		40	60		100

Course Outcomes: Students will be able to

CO1	Analyse the foundational elements of Indian Knowledge Systems and the contributions of ancient Indian universities to global knowledge development.
CO2	Understand the significance of traditional knowledge protection systems and the strategies for safeguarding indigenous intellectual property in the global economy.
CO3	Demonstrate how ancient Indian contributions to science, astronomy, and mathematics established foundational concepts that influenced global scientific development.
CO4	Analyse the technological innovations and architectural achievements of ancient Indian civilisations

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	-	-	-	-	3	-	2	-	-	-	3
CO2	-	-	-	-	-	1	-	3	-	**1**	-	**3**
CO3	**1**	2	-	2	-	2	-	-	-	-	-	3
CO4	**2**	-	1	-	-	1	2	**-**	-	-	-	3
Average	0.75	0.5	0.25	0.5	-	**1.75**	0.5	1.25	-	0.25	-	3

COURSE CURRICULUM

Course Outline	
Unit-1: Introduction to the Indian Knowledge System	**(Contact Hours: 6)**
Indian Knowledge System: The Vedas, The Vedangs, Literature (Smriti, Puran, Itihas, Upnishad), Main Schools of Philosophy (6 + 3), Indian Knowledge System vs Western Knowledge System; Ancient Education System: Major contributions of ancient Indian universities like Nalanda, Takshashila, Vikramshila, Odantapuri, Nagarjuna, etc., to the knowledge system; Development of Indian Education system from ancient to modern	
Unit-2: Protection of Traditional Knowledge & Intellectual Property	**(Contact Hours: 6)**
Systems of traditional knowledge protection, The need for protecting traditional knowledge, Significance of TK Protection, the value of TK in the global economy, Role of Government to harness TK, Legal concepts for the protection of traditional knowledge, Patents and traditional knowledge, Strategies to increase protection of traditional knowledge	
Unit-3: Indian Knowledge in Science, Astronomy and Mathematics	**(Contact Hours: 10)**
Science: Concepts of Matter, Life and Universe, Gravity, Sage Agastya's Model of Battery, Velocity of Light; Vedic Cosmology, Astronomy: Bhāratīya Kāla-gaṇanā, Indian Calender system, The motion of the Sun and Moon. Mathematics: Numbers, fractions and geometry in the Vedas. Decimal nomenclature of numbers in the Vedas. Zero and Infinity. Simple constructions from Sulba-sutras, Important texts of Indian mathematics. A brief introduction to the development of algebra, trigonometry and calculus.	
Unit-4: Indian Knowledge in Engineering, Technology, and Architecture	**(Contact Hours: 8)**
Pre-Harappan and Sindhu Valley Civilization, Laboratory and Apparatus, Juices, Dyes, Paints and	

ANNEXURE-II

Cements, Glass and Pottery, Metallurgy, Engineering Science and Technology in the Vedic Age and Post-Vedic Records, Iron Pillar of Delhi, Rakhigarhi, Mehrgarh, Sindhu Valley Civilization, Marine Technology and Bet–Dwārkā

Textbooks:

[1] A. Jha, Traditional Knowledge System in India. New Delhi: Atlantic Publishers, 2002.
[2] K. Kapoor and M. Danino, Knowledge Traditions and Practices of India. New Delhi: Central Board of Secondary Education (CBSE), 2012.
[3] P. Abhang, P. Moghe, P. Holay, and S. Kulkarni, Rediscovering Indian Knowledge System. Pune: Sankalp Publication, 2022.
[4] S. Raha et al., History of Science in India, vol. 1, part I, part II, vol. VIII. Kolkata: National Academy of Sciences, India and The Ramakrishna Mission Institute of Culture, 2014.
[5] M. Danino, Indian Knowledge Systems: Insights for the Contemporary World. Shimla: Indian Institute of Advanced Study, 2021.
[6] T. S. Bhanu Murthy, Ancient Indian Mathematics: Its Contribution to Modern Age. New Delhi: New Age International Publishers, 2018.

Reference Books:

[1] P. Kohle et al., Eds., Pride of India: A Glimpse of India's Scientific Heritage. Delhi: Samskrita Bharati, 2006.
[2] K. D. Verma, Vedic Physics. Delhi: Motilal Banarsidass Publishers, 2012.
[3] S. Soni, India's Glorious Scientific Tradition. New Delhi: Ocean Books Pvt. Ltd., 2010.
[4] A. L. Basham, The Wonder That Was India. London: Sidgwick & Jackson, 1954.
[5] V. K. Gupta, Traditional Knowledge Digital Library: A Tool to Prevent Bio-piracy. New Delhi: Council of Scientific and Industrial Research, 2011.
[6] D. F. Robinson et al., Protecting Traditional Knowledge: The WIPO Intergovernmental Committee on Intellectual Property and Genetic Resources, Traditional Knowledge and Folklore. Geneva: World Intellectual Property Organization (WIPO), 2017.
[7] S. Mohan, Indian Science Through the Ages. New Delhi: Publications Division, Ministry of Information and Broadcasting, Government of India, 2015.
[8] D. Chattopadhyaya, Science and Civilization in Ancient India. Calcutta: Firma KLM, 1977.

Online Learning Resources/URLs:

1. https://www.iksindia.org/ebook.php
2. https://www.brhat.in/openlibrary > Choose option IKS
3. https://onlinecourses.swayam2.ac.in/imb23_mg53/preview
4. https://onlinecourses.swayam2.ac.in/ntr25_ed18/preview
5. https://iksindia.org/course-list.php

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

B. Tech. Information Technology (Hons.): Approved in 2nd meeting of Academic Council held on 24.06.2025.

ANNEXURE-II

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-272A	Operating Systems Lab	0	0	2	1
		CIE	SEE		Total
		50	50		100

Course Outcomes	
CO1	Demonstrate proficiency in Unix/Linux environments, including general-purpose commands, shell scripting (Bash, Bourne, C shell), and text editing using the **vi** editor.
CO2	Implement and analyze process creation, management, and CPU scheduling algorithms, demonstrating the ability to simulate an operating system environment.
CO3	Develop multithreaded applications, applying synchronization and deadlock handling techniques to ensure efficient execution
CO4	Implement virtual memory management, device management, and disk scheduling algorithms, demonstrating their role in enhancing the system performance.
CO5	Design and Implement secure file systems, demonstrating file operations, directory management, and access control mechanisms.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PS O2
CO1	3	3	-	-	2	-	-	-	-	1	-	-	-	3
CO2	3	3	-	-	2	-	-	1	1	1	-	-	-	3
CO3	3	3	2	-	2	-	-	-	1	1	-	-	-	3
CO4	3	3	2	-	2	-	-	-	1	1	-	-	-	3
CO5	3	3	2	-	2	1	1	1	-	1	-	-	1	3

List of Experiments

Week No.	Program Description	Mapping with BL	Mapping with CO
Week 1:	Installation of Linux operating system.	3	CO1
Week 2:	Execution of simple C and C++ programs like addition, subtraction, multiplication etc. using CC and GCC compiler with their options.	3	CO1
Week 3:	Implement a program that simulates system calls for basic operations such as process creation, file manipulation, and device management. Demonstrate how system calls interact with the operating system components and services.	3	CO2
Week 4	Implement the following CPU scheduling algorithms: (a) Round Robin (b) SJF (c) FCFS (d) Priority	2	CO2

ANNEXURE-II

Week 5:	Create threads for multiple tasks to run concurrently, improving efficiency.	3	CO3
Week 6:	Implement to illustrate two processes communicating using shared memory.	2	CO3
Week 7:	Implement producer and consumer problem using semaphores	2	CO3
Week 8:	Implement Banker's Algorithm for Deadlock Avoidance	2	CO3
Week 9:	Implement the following: Multiprogramming with a fixed number of tasks (MFT) Multiprogramming with a variable number of tasks (MVT)	2	CO4
Week 10:	Implementation of worst, best & first fit for contiguous allocation assuming randomly generated free space list.	3	CO4
Week 11:	Implement the following page replacement algorithms: a) FIFO b) LRU c) LFU	2	CO4
Week 12:	Implement a file system simulator to demonstrate different file access methods (sequential, direct, indexed). Design a directory structure and simulate file operations such as creation, deletion, reading, and writing.	2	CO5
Week 13:	Develop a program to simulate file system implementation techniques, including different file allocation methods (contiguous, linked, indexed) and free-space management techniques. Implement a basic file system and directory structure.	3	CO5
Week 14:	a) Study of Unix/Linux general purpose utility command list: man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout,shutdown. b) Study of Bash shell, Bourne shell and C shell in Unix/Linux operating system. c) Study of Unix/Linux file system (tree structure)	3	CO1

Suggested Resources:

1. John Goerzen, Linux Programming Bible -Wiley Dream Tech India (P) Ltd.
2. Richard L. Peterson, Complete Reference, Red Hat Linux—TMH
3. Patrick Volker Ding, Kevin Richard, Eric Foster-Johnson, Linux Configuration & Installation BPB publication.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-274A	Design & Analysis of Algorithms Lab	0	0	2	1
		CIE	SEE		Total
		50	50		100

Course Outcomes	
CO1	Analyze and compare the time complexity of various sorting algorithms through practical implementation, demonstrating efficiency of algorithm selection.
CO2	Implement and evaluate advanced data structures such as AVL trees and BST to optimize search and retrieval operations
CO3	Apply graph traversal and shortest path algorithms for problem-solving in both the directed and undirected graphs.
CO4	Design and implement divide-and-conquer algorithms, showcasing their effectiveness for solving computational problems.
CO5	Apply greedy algorithms and dynamic programming techniques to solve optimization problems and evaluate their efficiency and applicability.
CO6	Solve combinatorial problems using backtracking and branch & bound approaches.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PS O2
CO1	3	3	3	-	-	-	-	-	-	-	-	1	2	-
CO2	3	3	3	-	2	-	-	-	1	-	-	1	2	-
CO3	3	3	-	3	-	-	-	-	1	-	-	1	2	-
CO4	3	3	3	-	3	-	-	-	1	-	-	1	2	-
CO5	3	3	3	3	-	1	-	-	1	-	-	1	2	-
CO6	3	3	3	3	-	-	-	1	1	-	-	1	2	-

List of Experiments

Week No.	Program Description	Mapping with BL	Mapping with CO
Week 1:	Implement Quick Sort to arrange a given set of elements in order and measure the time taken for sorting. Conduct multiple experiments by varying n, the number of elements in the list. The dataset can be sourced from a file or generated using a random number generator. Solve the problem using both the simple method and the divide-and-conquer approach, then analyze the results to determine the efficiency of each method.	BL4	CO1, CO4
Week 2:	Implement the Merge Sort algorithm to arrange a given set of elements in order and measure the time taken for sorting. Conduct multiple experiments by varying n, the number of elements in the list. The dataset can be sourced from a file or	BL4	CO1, CO4

ANNEXURE-II

	generated using a random number generator. Solve the problem using Merge Sort, then analyze the results to determine its efficiency.		
Week 3:	Implement insertion and deletion operations on an AVL Tree (or Red-Black Tree), demonstrate rotations, and compare the height/operations count with a simple BST.	BL4	CO2
Week 4:	Implement the Topological ordering of vertices in each digraph.	BL3	CO3
Week 5:	Print all the nodes reachable from a given starting node in a digraph using BFS and DFS methods.	BL3	CO3
Week 6:	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.	BL3	CO5
Week 7:	Find the shortest paths in each directed weighted graph (with possible negative edges) using Bellman-Ford.	BL3	CO5
Week 8:	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's and Prim's algorithms.	BL3	CO5
Week 9:	Find a subset of a given set S = {s1, s2,….., sN} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1,2,6} and {1,8}. A suitable message is to be displayed if the given problem instance doesn't have a solution.	BL3	CO5
Week 10:	Implement 0/1 Knapsack problem using Dynamic Programming.	BL3	CO5
Week 11:	Implement and compare Branch and Bound and Dynamic Programming for solving the Traveling Salesperson Problem (TSP) to determine the optimal solution and evaluate computational efficiency.	BL4	CO5, CO6
Week 12:	Implement N-Queen's problem using Back Tracking.	BL3	CO6

Suggested Resources:

1. T. Cormen, C. Lieserson, R. Rivest, C. Stein, "Introductions to Algorithms", Third Edition, PrenticeHall/India, 2009.
2. Sartaj Sahni, "Data Structures, Algorithms and Applications in C++", Universities Press Pvt. Ltd., 2012.
3. S. Sridhar, "Design and Analysis of Algorithms", Oxford University Press, 2014.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-276A	Database Management Systems Lab	0	0	2	1
		CIE	SEE		Total
		50	50		100

Course Outcomes	
CO1	Design and implement **structured databases** for diverse applications (retail, library, student admissions, and sales), applying **SQL operations**, enforcing **data integrity constraints**, and utilizing **query optimization techniques** such as **ORDER BY, GROUP BY, HAVING, and pattern matching (LIKE)** to ensure efficient data retrieval and management.
CO2	Develop SQL queries for **advanced data retrieval, modification, integrity constraints, joins, views, and pattern matching**, ensuring efficient database management.
CO3	Implement **stored procedures, triggers, and views** to enhance data handling and automation in relational databases.
CO4	Optimize query performance using indexing, transactions, and user access management.
CO5	Explore NoSQL concepts and perform CRUD operations in MongoDB, demonstrating adaptability to modern database technologies.
CO6	Apply database management techniques to **real-world applications**, showcasing practical knowledge and problem-solving skills.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO 2
CO1	3	3	3	-	3	-	-	-	1	-	-	1	-	-
CO2	3	3	-	-	3	-	-	-	-	-	-	-	-	-
CO3	3	3	-	-	3	-	-	-	-	-	-	1	-	-
CO4	-	3	3	-	3	1	-	-	-	-	1	-	-	-
CO5	3	3	-	-	3	-	-	-	-	1	-	1	-	-
CO6	-	-	3	-	-	1	-	-	1	1	1	-	-	1

List of Experiments

Week No.	Program Description	Mapping with BL	Mapping with CO
Week 1:	Create a database for a retail store to manage customer details. Perform insertion, deletion, and modification operations. Retrieve customer records in ascending order using ORDER BY.	BL3	CO1
Week 2:	For a library database, write SQL queries using SELECT, WHERE, ORDER BY, and GROUP BY to list books by genre and filter books published after 2015.	BL3	CO2
Week 3:	Design a student admission table and enforce PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE, CHECK, and DEFAULT constraints to ensure data integrity.	BL3	CO1
Week 4:	In a sales database, calculate average monthly sales per branch using arithmetic and relational operators. Use GROUP BY, HAVING, and pattern matching with LIKE to filter high-performing branches.	BL3	CO2
Week 5:	Create a view to display employees working on more than one project using GROUP BY and HAVING. Create another view	BL3	CO3

ANNEXURE-II

	showing employees not assigned to any project using NOT IN or LEFT JOIN.		
Week 6:	In a student-course database, use NATURAL JOIN, EQUI JOIN, and OUTER JOIN to display student names, course names, and enrollment status.	BL3	CO2
Week 7:	Write a stored procedure to compute a telephone bill. Use conditional logic to apply different billing rates based on usage slabs.	BL3	CO3
Week 8:	Develop a procedure to assign incentives to employees working on all available projects. Display a message using control structures if no such employee exists.	BL3	CO3
Week 9:	Compute income tax for employees using predefined salary slabs. Write a stored procedure that calculates tax and inserts the results into a separate table.	BL3	CO3
Week 10:	Implement BEFORE and AFTER triggers on an employee table to log changes made during insert, delete, and update operations into an audit table.	BL3	CO3
Week 11:	Simulate a booking system and demonstrate use of COMMIT, ROLLBACK, and SAVEPOINT to manage transaction failures and recovery in Postgre.	BL3	CO4
Week 12:	Create indexes on frequently queried fields (e.g., employee ID, department) and analyze query performance with and without indexing using EXPLAIN in postgre.	BL4	CO4
Week 13:	Assign and revoke access rights to different user roles in a payroll system using GRANT and REVOKE statements. Demonstrate role-based access control in postgre.	BL3	CO4
Week 14:	In MongoDB, create a collection for products and perform insertMany (), updateOne (), deleteMany (), and find () queries to manage and retrieve product details.	BL3	CO5
Week 15:	Develop a real-world project using the above studied databases.	BL6	CO6

Suggested Resources:

1. Learning MySQL: Get a Handle on Your Data by Vinicius M Grippa (Author), Sergey Kuzmichev (Author)
2. PostgreSQL: Up and Running a Practical Guide to the Advanced Open-Source Database by Regina O. Obe, Leo S. Hsu · 2017.
3. MongoDB Complete Guide, by Manu Sharma, BpB Publications

ANNEXURE-II

BT-ECE-272A			Microprocessors Based System Design Lab				
L	**T**	**P**	**Credits**	**Semester End Examination (SEE)**	**Continuous Internal Evaluation (CIE)**	**Total Marks (CIE+SEE)**	**Time**
0	**0**	**2**	**1**	**50**	**50**	**100**	**3 Hr.**

Pre-requisites: Basics of digital electronics and computer architecture.

Course Outcomes

CO1	Implement Fundamental Arithmetic and Logical Operations in Assembly Language
CO2	Apply Memory Manipulation Techniques in Assembly Language
CO3	Implement String and Data Processing Operations in Assembly Language
CO4	Develop Algorithmic Solutions for Problem Solving in Assembly Language

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO2
CO1	3	-	-	-	-	-	-	1	1	-	-	-	-	1
CO2	3	3	3	2	-	-	-	1	1	-	-	1	1	2
CO3	3	3	-	2	3	-	-	1	1	-	-	-	3	3
CO4	3	3	3	2	3	-	-	1	1	-	-	1	3	3

List of Experiments:

1. Write an assembly language program to add/sub two 8-bit, 16-bit and 32-bit numbers.
2. Write an assembly language program to multiply two 16 bit unsigned/signed numbers.
3. Write an assembly language program to divide two unsigned/signed numbers (32/16, 16/8, 16/16,8/8).
4. Write an assembly language program to find 1's and 2's complement of a number.
5. Write an assembly language program to evaluate the following equation:

$$x = \frac{(6 \times 8) + 9 - 7}{5}$$

6. Write an assembly Language program to multiply 2, 8-bit numbers using repetitive addition. Assume the multiplicand is saved at location DS:0050 and the multiplier is available at location DS:0060.
7. Write an assembly language program to save 55H on 10 memory locations staring from DS:200H onwards.
8. Write an ALP to copy a chunk of 5 bytes of data starting from memory location DS:0100H to memory location DS:0110H
9. Write an ALP to exchange two chunks of data, 8 bytes each. First data begins from memory location DS:0100H and the second data begins from memory location DS:0120H.
10. Write an assembly language program to add 10 Byte of data available in the memory from

location DS:0050H onwards and save the result in memory location DS:0020H.

11. Write an assembly language program to convert the following two unpacked BCD numbers into a packed BCD number:
    a. 05H
    b. 08H
12. Write an assembly language program to search the presence of a given byte in a list of 20 byte available in the memory starting from the memory location DS:0100H.
13. Write an assembly language program to count the number of 1's in a given byte. Save the count in DL register.
14. Write an assembly language program to check the parity of a given byte. If the parity is odd save 0FH in DL register. If the parity is even saved 0F0H in the DL register.
15. Write an assembly language program to find smallest /largest number from array of n numbers.
16. Write an assembly language program to arrange numbers in array in ascending / descending order.
17. Write an assembly language program to convert Hex to Decimal, Decimal to Hex.
18. Write an assembly language program to compare two strings using string instructions / without using string instructions.
19. Write an assembly language program to display string in reverse order, string length, concatenation of two strings.
20. Write an assembly language program to find factorial of a number.
21. Write an assembly language program to transfer the data between 8086 Microprocessor and Peripheral Device Using 8255 Programmable Peripheral Interface.

Note: Students have to perform at least 12 experiments from the given list of practical.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-252A	**Foundations of Data Analytics, Machine Learning and Statistics**	**3**	**0**	**2**	**4**
		CIE	**SEE**		**Total**
		40	**60**		**100**

Course Outcomes	
CO1	**Analyze** the evolution of traditional software to data-centric systems, **compare** data science with data analytics and machine learning, and **evaluate** their applications, components, and lifecycle in various domains.
CO2	**Apply** statistical concepts including population sampling, parameter estimation, correlation, and regression models to **interpret** data patterns and **support** decision-making in data-driven environments.
CO3	**Demonstrate** hypothesis testing techniques, including significance testing for population means, variances, and proportions, and **implement** ANOVA for comparative statistical analysis.
CO4	**Utilize** statistical tools and methodologies to **conduct** case studies in business data analytics and healthcare machine learning applications, ensuring effective real-world data analysis.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO1	PO2	PO3	PO 4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO1	PSO2
CO1	3	2	-	-	-	-	-	-	-	1	-	1	-	3
CO2	3	2	3	3	3	-	-	-	-	1	1	1	1	3
CO3	3	2	3	3	3	-	-	-	-	1	1	1	1	3
CO4	3	2	3	3	3	1	1	1	-	1	1	1	1	3

Unit 1	**Introduction**	**Contact Hours: 11**
Evolution from traditional software to data centric system, Definition of Data science, applications, components, lifecycle. Data Analytics- Definition, types. Data science v/s Data Analytics. Machine Learning- Definition, comparison with traditional programming, categories- supervised, unsupervised, reinforced, examples. Interrelation among data science, data analytics and machine learning. Importance of statistics.		
Unit 2	**Statistics**	**Contact Hours: 12**

ANNEXURE-II

Introduction, Populations and Samples, Census and Sampling Method, Parameter and Statistic, Independent and Dependent Variables. Mathematical expectations and their properties. Correlation & Regression-Partial Correlation, Multiple Regression Model, Non-linear Regression, Polynomial Regression Models		
Unit 3	**Decision Analysis**	**Contact Hours: 12**
Introduction to Null and Alternative Hypothesis, Type I and Type II Error, The Procedure of Hypothesis Testing; Hypothesis Testing of a Population Mean: Sample, a proportion (One Sample), Population Variance, Population Mean: Two Independent Samples, Dependent Samples (Paired Samples), Two Population Proportion, Two Population Variances; Analysis of Variance (ANOVA).		
Unit 4	**Case Study**	**Contact Hours: 06**
Case Study using statistical tools: Data Analytics in Business, Machine Learning in Healthcare		

List of Experiments:

1. Import a real-world dataset (e.g., CSV) and perform basic cleaning.
2. Calculate mean, median, mode, standard deviation, and variance.
3. Create bar charts, histograms, boxplots, scatter plots.
4. Explore dataset features and visualize relationships.
5. Perform descriptive and diagnostic analytics on e-commerce dataset.
6. Demonstrate use of probability distributions (normal, binomial).
7. Apply t-test or chi-square test to test assumptions.
8. Perform simple linear regression to predict outcome.
9. Use decision tree or logistic regression on a labeled dataset.
10. Apply K-means clustering to segment data.

Reference Books:

1. Probability for Statistics and Machine Learning: Anirban Das Gupta – 2011.
2. Fundamental Mathematical Statistics: S.C Gupta and V.K Kapoor, Sultan Chand & Sons
3. An Introduction to Statistics with Python with Applications in Life Sciences by Thomas Haslwanter, 2016.
4. Jain V.K., “Data Sciences”, Khanna Publishing House, Delhi
5. Applied statistical methods, Irving W. Burr, Academic press.
6. Probability, Statistics and Random process, Dr. K. Murugesan & P. Gurusamy by Anuradha Agencies, Deepthi publications.

ANNEXURE-II

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-254A	**Server-Side Programming**	**3**	**0**	**2**	**4**
		CIE	**SEE**		**Total**
		40	**60**		**100**

Course Outcomes	
CO1	**Analyze** client-server architecture, HTTP/HTTPS protocols, and request/response mechanisms to **implement** secure and efficient server-side programming in web development.
CO2	**Develop** dynamic web applications using Java Servlets and JSP, **manage** sessions with cookies, and **configure** Apache Tomcat for optimized web deployment.
CO3	**Utilize** Spring MVC for **building** scalable web applications, **apply** dependency injection for modular architecture, and **perform** transaction management and CRUD operations using Spring DAO.
CO4	**Integrate** Hibernate ORM for seamless database interaction, **design** efficient data models using Hibernate annotations and HQL queries, and **execute** CRUD operations with performance optimization.
CO5	**Implement** Java Server Faces (JSF) to **create** interactive user interfaces, **manage** request processing lifecycle, and **enhance** UI components using PrimeFaces for improved accessibility.
CO6	**Deploy** full-stack applications by **designing** backend and frontend integration, **utilizing** debugging, profiling, caching strategies, and **optimizing** performance through CI/CD automation.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)														
	PO1	**PO2**	**PO3**	**PO 4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	3	3	3	2	-	-	-	-	1	1	1	1	3	3
CO2	3	3	3	2	-	-	-	-	1	1	1	1	3	3
CO3	3	3	3	2	3	-	-	-	1	1	1	1	3	3
CO4	3	3	3	2	3	-	-	-	1	1	1	1	3	3
CO5	3	3	3	2	3	-	-	-	1	1	1	1	3	3
CO6	3	3	3	2	3	-	-	-	1	1	1	1	3	3

ANNEXURE-II

Unit 1	**Introduction**	**Contact Hours: 11**
Overview of client-server architecture, Role of server-side programming in web development, HTTP/HTTPS, request/response cycle, status codes. Introduction to various tools & techniques used in server-side programming. Java Servlets: Servlets, Servlet Architecture, Servlet Life Cycle, Generation of dynamic content with Servlet , Session Handling, Understanding Cookies, Installing and Configuring Apache Tomcat Web Server, Creating a Download Manager in Java		
Unit 2	**Java Server Pages**	**Contact Hours: 12**
Introduction to JSP , Comparison with Servlet, JSP Architecture, JSP: Life Cycle, Scripting Elements, Directives, Action Tags, Implicit, Objects, Expression Language(EL), JSP Standard Tag Libraries(JSTL), Custom Tag, Session Management, Exception Handling, CRUD, Application.		
Unit 3	**Java Web Frameworks**	**Contact Hours: 12**
Spring MVC, Spring: Introduction, Architecture, Spring MVC Module, Life Cycle of Bean Factory, Explore: Constructor Injection, Dependency Injection, Inner Beans, Aliases in Bean, Bean Scopes, Spring Annotations, Spring AOP Module, Spring DAO, Database Transaction Management, CRUD, Operation using DAO and Spring API.		
Unit 4	**Hibernate**	**Contact Hours: 10**
Introduction to Hibernate, Exploring Architecture of Hibernate, Object Relation Mapping (ORM) with Hibernate, Hibernate Annotation, Hibernate Query Language (HQL), CRUD Operation using Hibernate API. Java Server Faces: Features of JSF, JSF: Architecture, Request processing Life cycle, Elements, Expression Language (EL), Standard Component, Facelets Tag, Convertor Tag, Validation Tag, Database Access, PrimeFaces		

List of Experiments:

1. Create and deploy a basic Java Servlet.
2. Understand the phases of a servlet's life cycle: initialization, service, and destruction. Deploy and interact with the servlet to trigger its methods. View logs to analyze how the life cycle methods are executed. Discuss findings on servlet instantiation and destruction processes.
3. Implement session tracking and cookie-based authentication in servlets. Observe cookie storage in the browser and analyze session behavior upon closing and reopening the browser.
4. Develop a servlet-based file download manager. Optimize file streaming using buffered output. Test with various file types and measure download efficiency.

B. Tech. Information Technology (Hons.): Approved in 2nd meeting of Academic Council held on 24.06.2025.

ANNEXURE-II

5. Understand the life cycle of JSP pages and utilize implicit objects. Deploy the page and trigger lifecycle events through multiple user interactions. Analyze logs to understand JSP execution and object handling.
6. Implement session tracking mechanisms and exception handling in JSP. Introduce intentional errors and implement JSP exception handling using <%@ page errorPage="error.jsp" %> and isErrorPage. Evaluate the session persistence behavior and analyze how JSP handles errors gracefully.
7. Develop a complete CRUD (Create, Read, Update, Delete) application using JSP, JSTL, and custom tags.
8. Implement Hibernate to map Java objects to database tables using annotations.
9. Write a program to illustrate the request-processing flow in Spring MVC. Use DispatcherServlet to route requests and analyze the response flow. Deploy and test the web application while logging the request-response cycle.
10. Investigate how Spring manages beans and injection mechanisms.
11. Use Spring AOP to apply logging and security aspects dynamically. Apply method interception to log user activities and secure sensitive operations.
12. Develop a complete CRUD operation using DAO and Spring API. Deploy the application and test database interactions for consistency and efficiency.
13. Implement transaction management and exception handling in Spring applications.
14. Illustrate how JSF handles UI components and form submissions.

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-258A	Linux & Cloud Technology	4	0	0	4
		CIE	SEE		Total
		40	60		100

Course Outcomes

CO1	Identify and use Linux utilities to create and manage simple file processing operations, organize directory structures with appropriate security, and develop shell scripts to perform more complex tasks.
CO2	Effectively use the Linux system to accomplish typical personal, office, technical, and software development tasks.
CO3	Monitor system performance and network activities. Effectively use software development tools including libraries, preprocessors, compilers, linkers, and make files.
CO4	Comprehend technical documentation, prepare simple readable user documentation and adhere to style guidelines.
CO5	Analyze the use of cloud in different editors in Linux environment.
CO6	Introduce the domain of cloud.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2
CO1	2	3	-	3	-	-	-	-	-	-	-	-	1	1
CO2	3	-	1	-	2	-	-	-	-	-	-	-	1	1
CO3	-	2	3	1	-	-	-	-	-	-	-	-	1	2
CO4	-	2	1	3	-	-	-	-	-	-	-	-	1	2
CO5	2	1	-	2	-	-	-	-	-	-	-	-	1	2
CO6	2	-	2	-	1	-	-	-	-	-	-	-	1	3

Unit 1	**Introduction to Unix**	**Contact Hours: 13**
Introduction to Unix: History, Unix vs Linux, Basic Terminologies (kernel, shell, terminal), Features, Unix Architecture, File System Structure, Types of Users, File Types, Internal vs External Commands, Basic UNIX Commands: ls, cd, pwd, cp, mv, rm, man, mkdir, rmdir, touch, cat, more, less		
Unit 2	**Shell Programming**	**Contact Hours: 17**
Shells and its types: Bourne, Korn, Bash, C shell, tcsh – Introduction and Differences, Shell Interpretive Cycle, Command Structure, Variables, Environment Variables, Wildcards, Metacharacters, Escaping & Quoting, Command Substitution, Conditional Statements in Shell (if, case), Exit Status, Looping Constructs: for, while, until, and their use in scripts.		
Unit 3	**Graphical Interface**	**Contact Hours: 13**
Session Management, Basic Operations- Login, Logout, Startup Files, Multitasking & Background Jobs (&, jobs, fg, bg), Graphical Desktop: GNOME, KDE, XFCE – Overview and Use-Cases. Filters and Piping: Introduction, Simple filters: pr, head, tail, cut, paste, sort, nl, tr		

ANNEXURE-II

Unit 4	Introduction to Cloud	Contact Hours: 12
Introduction, Characteristics, Applications, Deployment Models (Public, private and hybrid cloud). Access of private and public cloud, Introduction to EC2, Key Pair Generation, Launch and Access Linux Instances.		

Textbook/s:

1. Unix and shell programming by Richard F. Gilberg and Behrouz A. forouzan

Reference Books:

1. Unix Shell programming by Stephen G. Kochan and Patric Wood
2. Sumitabha Das, "Unix Concepts and Applications", Tata McGraw Hill.

NOTE: 1. For the semester examination, nine questions are to be set by the examiner. Question no. 1, containing 5-7 short answer type questions, will be compulsory & based on the entire syllabus. Rest of the eight questions is to be set by setting two questions from each of the four units of the syllabus. The candidates will be required to attempt five questions in all, selecting one from each unit AND Question no. 1. All questions will carry equal marks.

2. The students will be allowed to use non-programmable scientific calculator. However, sharing / ex-change of calculator or any other items are prohibited in the examinations. No programmable calculators, mobile phones or other electrical/ electronic items are allowed in the examination.

ANNEXURE-II

Course Code	Course Title	L	T	P	Credits
BT-IT-278A	Linux & Cloud Technology Lab	0	0	2	1
		CIE	SEE		Total
		50	50		100

Course Outcomes	
CO1	Identify and use Linux utilities to create and manage simple file processing operations, organize directory structures with appropriate security, and develop shell scripts to perform more complex tasks.
CO2	Effectively use the Linux system to accomplish typical personal, office, technical, and software development tasks.
CO3	Monitor system performance and network activities. Effectively use software development tools including libraries, preprocessors, compilers, linkers, and make files
CO4	Comprehend technical documentation, prepare simple readable user documentation and adhere to style guidelines.
CO5	Analyze the use of different editors in the Linux environment.
CO6	Explore the application domain of Linux environment in AWS cloud.

Course Outcomes (CO) to Programme Outcomes (PO) mapping (scale 1: low, 2: Medium, 3: High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PS O2
CO1	2	3	-	3	-	-	-	-	1	1	-	-	1	1
CO2	3	-	1	-	2	-	-	-	1	1	-	-	1	1
CO3	-	2	3	1	-	-	-	-	1	1	-	-	1	2
CO4	-	2	1	3	-	-	-	-	1	1	-	-	1	2
CO5	2	1	-	2	-	-	-	-	1	1	-	-	1	2
CO6	2	-	2	-	1	-	-	-	1	1	-	-	1	3

List of Experiments

Week No.	Program Description	Mapping with BL	Mapping with CO
Week 1:	Setup and configure a Linux Environment using VirtualBox, WSL, or Live USB	3	CO1, CO2
Week 2:	Install Linux Operating System with Proper Partitioning and Boot Loader Configuration	4	CO1, CO2
Week 3:	Install and work with csh and tcsh Shells on Linux	3	CO1, CO2
Week 4	Demonstrate Shell Interpretive Cycle, Wildcards, Meta Characters, Escaping, and Quoting	3	CO1
Week 5:	Manage User Sessions, Groups, and File Permissions	4	CO1
Week 6:	Install and Configure Graphical Desktop Environment (GNOME/Xfce)	3	CO2
Week 7:	Implement Command Piping with Filters (pr, cut, sort, etc.)	4	CO1, CO2

ANNEXURE-II

Week 8:	Create Shell Scripts using Conditional Statements	3	CO1
Week 9:	Write Shell Scripts using Looping Constructs	3	CO1
Week 10:	Demonstrate File Redirection and Use of Commands like grep, awk, sed	3	CO1, CO2
Week 11:	Manage Software Packages using apt, yum, snap, and Source Compilation	4	CO2, CO3
Week 12:	Launch and configure a Linux EC2 Instance on AWS and Access via SSH	4	CO6
Week 13:	Deploy a Linux-based Text Editor on EC2	3	CO5, CO6
Week 14:	Host a Static Web Page using Apache or Nginx on EC2	3	CO6
Week 15:	Mini Project – Automate a Task or Configure a Network Service like FTP/SSH	6	CO1, CO3, CO6