Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech (CSE/ IT) - All Programs

SEMESTER- III (2025-29)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTCS306M	DCC	Fundamentals of Data Structures	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

COURSE OBJECTIVES:

The student will have ability to:

1. To understand efficient storage mechanisms of data for easy access.
2. To design and implementation of various basic and advanced data structures.
3. To introduce various techniques for representation of the data in the real world.
4. To develop applications using data structures.
5. To understand the concept of protection and management of data.

COURSE OUTCOMES:

Upon completion of the subject, students will be able to:

1. Get a good understanding of applications of Data Structures.
2. Develop applications using data structures.
3. Handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
4. Decide the appropriate data type and data structure for a given problem.
5. Select the best algorithm to solve a problem by considering various problem characteristics, such as the data size, the type of operations, etc.

SYLLABUS

UNIT I — 10 HOURS

Introduction: Overview of Data structures, Types of data structures, Primitive and Non-Primitive data structures and Operations, Introduction to Algorithms & complexity notations. Characteristics of Array, One-Dimensional Array, Operation with Array, Two-Dimensional Arrays, Three or Multi-Dimensional Arrays, Sparse matrix, Drawbacks of linear arrays. Strings, Array of Structures, Pointer and one-dimensional Arrays, Pointers and Two-Dimensional Arrays, Pointers and Strings, Pointer and Structure.

UNIT II — 9 HOURS

Linked List: Linked List as an ADT, Linked List Vs. Arrays, Dynamic Memory Allocation & De-allocation for a Linked List, Types of Linked List: Circular & Doubly Linked List. Linked. **List operations:** All possible insertions and deletion operations on all types of Linked list Reverse a Single Linked List; Divide a singly linked list into two equal halves, Application of Linked List.

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTCS306M	DCC	Fundamentals of Data Structures	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

UNIT III — 8 HOURS

Stack: The Stack as an ADT, Stack operation, Array Representation of Stack, Link Representation of Stack, Application of stack – Recursion, Polish Notation. Types of Recursions, problem based on Recursion: Tower of Hanoi. **The Queue**: The Queue as an ADT, Queue operation, Array Representation of Queue, Linked Representation of Queue, Types of Queue: Circular Queue & Dequeue, Introduction of Priority Queue, Application of Queues.

UNIT IV — 10 HOURS

Tree: Definitions and Concepts of Binary trees, Types of Binary Tree, Representation of Binary tree: Array & Linked List. General tree, forest, Expression Tree. Forest and general tree to binary tree conversion. Binary Search Tree Creation, Operations on Binary Search Trees: insertion, deletion & search an element, Traversals on Binary SEARCH TREE and algorithms. Height balanced Tree: AVL, B-Tree, 2-3 Tree, B+Tree: Creation, Insertion & Deletion. **Graph:** Definitions and Concepts Graph Representations: Adjacency MATRIX, Incidence matrix, Graph TRAVERSAL (DFS & BFS), Spanning Tree and Minimum Cost Spanning Tree: Prim's & Kruskal's Algorithm.

UNIT V — 8 HOURS

Sorting: Sorting Concept and types of Sorting, Stable & Unstable sorting. Concept of Insertion Sort, Selection sort, Bubble sort, Quick Sort, Merge Sort, Heap & Heap Sort, Shell Sort & Radix sort. Algorithms and performance of Insertion, selection, bubble, Quick sort & Merge sort.

TEXTBOOKS:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.
2. M. A. Weiss, *Data Structures and Algorithm Analysis in C++*, 4th ed. Upper Saddle River, NJ, USA: Pearson Education, 2014.

REFERENCE:

1. S. Sahni, *Data Structures, Algorithms, and Applications in C++*, 2nd ed. New Delhi, India: Universities Press, 2005.
2. M. T. Goodrich, R. Tamassia, and D. M. Mount, *Data Structures and Algorithms in C++*, 2nd ed. Hoboken, NJ, USA: Wiley, 2011.
3. R. K. Shukla, *Data Structures Using C & C++*. New Delhi, India: Wiley India, 2016.

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTCS306M	DCC	Fundamentals of Data Structures	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

4. E. Balagurusamy, *Data Structures Using C*. New Delhi, India: Tata McGraw-Hill, 2017.
5. P. S. Deshpande and O. G. Kakde, *C & Data Structures*. Boston, MA, USA: Charles River Media, 2015.
6. G. Pai, *Data Structures*. New Delhi, India: Tata McGraw-Hill, 2015.

LIST OF PRACTICALS

1. To develop a program to find an average of an array using AVG function.
2. To implement a program that can insert, delete and edit an element in array.
3. To implement an algorithm for insert and delete operations of circular queue and implement the same using array.
4. Write a menu driven program to implement the push, pop and display option of the stack with the help of static memory allocation.
5. Write a menu driven program to implement the push, pop and display option of the stack with the help of dynamic memory allocation.
6. Write a menu driven program to implementing the various operations on a linear queue with the help of static memory allocation.
7. Write a menu driven program to implementing the various operations on a linear queue with the help of dynamic memory allocation.
8. Write a menu driven program to implement various operations on a linear linked list.
9. Write a menu driven program to implement various operations on a circular linked list
10. Write a program for implementation of Bubble sort
11. Write a program for Insertion sort
12. Write a program for Merge Sort
13. Write a program to implement Heap sort
14. Write a program to implement Quick sort
15. Write a program to Construct a Binary Search Tree and perform deletion, inorder traversal on it
16. Write a program to develop an algorithm for binary tree operations and implement the same.
17. Write a program to design an algorithm for sequential search, implement and test it.
18. Write a program to develop an algorithm for binary search and perform the same.

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech (CSE/ IT) - All Programs

SEMESTER- III (2025-29)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTCS307M	DCC	Principles of Operating Systems	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

COURSE OBJECTIVES:

The student will have ability to:

1. Learn the fundamentals of Operating Systems.
2. Study the mechanisms of Operating System to handle processes and threads and their communication.
3. Gain knowledge of process management concepts that includes architecture, Mutual exclusion algorithms, deadlock detection and recovery algorithms.
4. Learn the mechanisms involved in memory management in Operating System.
5. Understand the components and management aspects of disc scheduling

COURSE OUTCOMES:

Upon completion of the subject, students will be able to:

1. Describe the structure of Operating System in detail.
2. Design and Implement Process management Techniques in Operating System.
3. Calculate CPU Scheduling criteria.
4. Understand The Memory Management of Operating System.
5. Elaborate Disc Scheduling.

SYLLABUS

UNIT I — 10 HOURS

Introduction to Operating System: Introduction and Need of operating system, Layered Architecture/Logical Structure of Operating system, Type of OS (Multiprogramming, Timesharing, Real-Time, Networked, Distributed, Clustered, Handheld), Operating system as Resource Manager and Virtual Machine, System Calls/Monitor Calls, Firmware- BIOS, Boot Strap Loader. Threads- processes versus threads, threading, concepts, models, kernel & user level threads, thread usage, benefits.

UNIT II — 9 HOURS

Process Management: Process Model, Creation, Termination, States & Transitions, Context Switching, Process Control Block, CPU and I/O bound, CPU scheduler- short, medium, long-term, dispatcher, Scheduling: - preemptive and non-preemptive, Static and Dynamic Priority. Criteria/Goals/Performance Metrics, scheduling algorithms- FCFS, SJFS, shortest remaining time, round robin, Priority scheduling, multilevel queue scheduling, multilevel feedback queue scheduling.

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech (CSE/ IT) - All Programs

SEMESTER- III (2025-29)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTCS307M	DCC	Principles of Operating Systems	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

UNIT III — 8 HOURS

Inter process Communication: Introduction to Message Passing, Race Condition, Critical Section Problem, Peterson's Solution, Semaphore, Classical Problems of Synchronization Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem. **Deadlock-** System model, Resource types, Deadlock Problem, Deadlock Characterization, Methods for Deadlock Handling, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock Detection, Recovery from Deadlock.

UNIT IV — 9 HOURS

Memory Management: concepts, functions, logical and physical address space, address binding, degree of multiprogramming, swapping, static & dynamic loading- creating a load module, loading, static & dynamic linking, memory allocation schemes- first fit, next fit, best fit, worst fit and quick fit.

Virtual Memory- concept, virtual address space, paging scheme, pure segmentation and segmentation with paging scheme hardware support and implementation details, memory fragmentation, demand paging, working set model, page fault frequency, thrashing, page replacement algorithms- optimal, FIFO, LRU, Bleady's anomaly; TLB (translation look aside buffer).

UNIT V — 9 HOURS

File Management: Concepts, Naming, Attributes, Operations, Types, Structure, File Organization & Access (Sequential, Direct ,Index Sequential) Methods, Memory Mapped Files, Directory Structures One Level, Two Level, Hierarchical/Tree, Acyclic Graph, General Graph, File System Mounting, File Sharing, Path Name, Directory Operations, Overview Of File System in Linux & Windows.

Input/output Subsystems- Concepts, Functions/Goals, Input/ Output devices- Block and Character, Spooling, Disk Structure & Operation, Disk Attachment, Disk Storage Capacity, Disk Scheduling Algorithm- FCFS, SSTF, Scan Scheduling, C-Scan Schedule.

TEXTBOOKS:

1. Silberschatz, *Operating System Concepts*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2018.
2. S. Tanenbaum, *Modern Operating Systems*, 4th ed. Harlow, U.K.: Pearson Education, 2014.

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech (CSE/ IT) - All Programs

SEMESTER- III (2025-29)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTCS307M	DCC	Principles of Operating Systems	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

REFERENCE:

1. A. S. Godbole, Operating Systems, 3rd ed. New Delhi, India: Tata McGraw-Hill Education, 2017.
2. W. Stallings, Operating Systems: Internals and Design Principles, 8th ed. Harlow, U.K.: Pearson Education, 2014.
3. V. Shukla, Operating Systems, 3rd ed. Indore, India: Kataria & Sons, 2013.
4. M. Singhal and N. G. Shivaratri, Advanced Concepts in Operating Systems. New Delhi, India: Tata McGraw-Hill Education, 2017.

LIST OF PRACTICALS

1. Implement and update the BIOS settings of your PC.
2. Suppose there are five printers connected to a system. Each printing process requires a different amount of time to complete. The CPU allocates a fixed time slice to each process. After the time slice expires, the next process enters the CPU. If a process is not completed within its time slice, it is re-entered into the queue based on First-Come, First-Served (FCFS) scheduling in a rotational manner. Apply scheduling to this scenario.
3. Implement Non-Preemptive Priority CPU Scheduling.
4. Implement Non-Preemptive Shortest Job First (SJF) CPU Scheduling.
5. Suppose there are five different resources connected to a system (e.g., three printers and two scanners), each requiring different times to complete their tasks. Identify which scheduling algorithm provides the best CPU performance.
6. Implement scheduling where the CPU executes the process that arrives first (FCFS Scheduling).
7. Implement Round-Robin CPU Scheduling.
8. Write a program to implement Semaphore.
9. Consider a situation where five faculty members are seated around a table with four ballpoint pens placed on it. At any time, only one pen can be picked up by one faculty member for writing. What happens if all attempt to pick up a pen simultaneously? Find the solution.
10. Consider a dentist's clinic with one dentist and one treatment chair, along with *n* waiting chairs for patients:
    - If no patient is present, the dentist sleeps in the chair.
    - When a patient arrives, they wake up the dentist.
    - If multiple patients arrive while the dentist is treating one, the remaining patients wait if chairs are available; otherwise, they leave if no chairs are free. Find the solution for this scenario.
11. Write a program to implement Memory Management Algorithms such as *First Fit*, *Best Fit*, and *Worst Fit*.
12. Demonstrate Virtual Memory Techniques such as *LRU* (Least Recently Used) and *FIFO* (First-In, First-Out).

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech (CSE/ IT) - All Programs

SEMESTER- III (2025-29)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTCS307M	DCC	Principles of Operating Systems	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

13. Implement Shortest Seek Time First (SSTF) Disk Scheduling Algorithm.
14. Implement SCAN Disk Scheduling Algorithm.
15. Implement Circular SCAN (C-SCAN) Disk Scheduling Algorithm.
16. Implement LOOK Disk Scheduling Algorithm.

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech (CSE/IT), CSE (ES, AI-IBM, DS, BDCE)

SEMESTER III (2025-2029)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTDSE323	DSE	Python Programming	60	20	20	30	20	2	0	2	3

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

COURSE OBJECTIVES:

The student will have ability:

1. Analyse fundamental Python programming structures, data types, and optimization techniques to construct computationally efficient code.
2. Formulate algorithmic solutions using dynamic programming, sliding windows, and modular syntax to solve complex problems on competitive programming platforms.
3. Apply vectorized data manipulation and statistical visualization principles using NumPy, Pandas, and Matplotlib to preprocess real-world datasets for downstream AI/ML workflows.

COURSE OUTCOMES:

Upon completion of the subject, students will be able to:

1. Implement Python core constructs, exception handling, and functional paradigms to build clean, error-resistant scripts.
2. Synthesize high-performance vectorized computation matrices and relational multi-index data frames for exploratory data analysis.
3. Construct production-grade statistical plots and clean, pre-processed data frames formatted directly for machine learning pipelines.

SYLLABUS

UNIT I — 6 HOURS

Python Basics & Control Structures: Environment & Setup: Installing Python and VS Code; running scripts via the terminal; using pip to install packages. **Variables & Core Data Types**: Basic tokens, keywords, and identifiers; working with Integers, Floats, Strings, and Booleans; explicit Type Casting. **Operators & Input/Output**: Arithmetic, Assignment, Comparison, and Logical operators; reading user input with input(); formatting output using f-strings. **Conditional Statements**: Single and multi-way branching layouts using if, elif, and else blocks; nested conditions. **Looping Mechanisms**: Fixed iterations using for loops and the range() generator; conditional tracking using while loops; loop break parameters using break and continue.

UNIT II — 6 HOURS

Modular Programming, Arrays & Hashing: Functions & Scope: Defining functions with def; executing the return statement; distinguishing between Local and Global variable scope. **Function Parameters:** Passing positional arguments, default keyword parameters, and handling variable-length inputs using *args and kwargs.

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTDSE323	DSE	Python Programming	60	20	20	30	20	2	0	2	3

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

Linear Data Configurations: Python Lists and Tuples; indexing, slicing syntax [start : stop : step], and list manipulation methods (append, insert, pop, remove). **Dictionaries & Sets:** Creating key-value pairs in Dictionaries; basic dictionary tracking methods (keys(), values(), items()); Set creation and unique element mathematical operations.

UNIT III — 6 HOURS

Object-Oriented Programming: Transitioning from procedural code to Object-Oriented Programming; defining a class and instantiating an object. **Attributes & Methods:** Understanding instance variables; creating class methods; utilizing the instance reference parameter (self). **Object Lifecycle:** Implementing the constructor framework using the __init__() method to initialize object state parameters. **Core OOP Pillars:** Implementing basic Single Inheritance to reuse code; understanding Encapsulation via public and private attribute prefixes. **Robust Programming:** Anticipating runtime crashes; implementing structured code protection using try, except, and finally blocks; catching standard errors (ValueError, IndexError, KeyError).

UNIT IV — 6 HOURS

NumPy: Array Memory Allocation Hierarchy: N-Dimensional array (ndarray) layout, contiguous block allocations, homogeneous data storage bounds vs standard Python lists, managing data types (dtype). **Vectorization & Matrix Operations:** Eliminating raw loops via universal functions (ufuncs), broadcasting rules across arrays of mismatched shapes, logical indexing masks, element filtering, sorting array axes. **Mathematical Dimension Manipulations:** Axis math definitions (axis=0 row vectors vs axis=1 column vectors), array reshape metrics, flattening operations, matrix splitting and stacking procedures (hstack, vstack). **Data Synthesis:** Generating pseudo-random numbers (np.random), managing execution continuity with seeds, generating linear and step distribution scales (linspace, arange).

UNIT V — 6 HOURS

Data Structural Analysis & Visualization: Pandas Architecture: Series and DataFrame data layouts, handling structural metadata labels, alignment mechanics during mathematical execution. **Data Ingestion & Cleaning:** Parsing tabular file inputs (pd.read_csv, pd.read_excel), locating structural missing entities (isna, isnull), managing missing rows/columns via drops (dropna) and imputations (fillna), removing duplicate indexes. **Transformation & Aggregation Engines:** Slicing indexes via positional labels (loc, iloc), running query filtering criteria, applying mapping alterations across axis elements, the GroupBy lifecycle split-apply-combine paradigm, joining relational data schemas (merge, concat). **Matplotlib Visual Analytics:** Figure vs Axes execution frames, styling line graphs,

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTDSE323	DSE	Python Programming	60	20	20	30	20	2	0	2	3

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

building data point scatter distributions, density tracking histograms, deploying multi-panel rendering grids via plt.subplots, formatting legends, ticks, and coordinate parameters for analysis pipelines.

TEXTBOOKS:

1. P. Deitel and H. Deitel, *Intro to Python for Computer Science and Data Science: Learning to Program with AI, Big Data and the Cloud*, 1st ed. (Global Edition). New York, NY, USA: Pearson, 2022.
2. W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.

REFERENCE:

1. J. Shovic and A. Simpson, *Python All-in-One For Dummies*, 3rd ed. New York, NY, USA: John Wiley & Sons (Apress Media Partner), 2024.
2. E. Matthes, *Python Crash Course: A Hands-On, Project-Based Introduction to Programming*, 3rd ed. San Francisco, CA, USA: No Starch Press, 2023.
3. J. VanderPlas, *Python Data Science Handbook: Essential Tools for Working with Data*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
4. A. Aziz and T. Lee, *Elements of Programming Interviews in Python: The Insiders' Guide*, 1st ed. CreateSpace Independent Publishing, 2017.
5. RHA OLE Platform: Red Hat Academy Online Learning Environment - Core Enterprise System Automation Engineering with Python tracks.
6. Python Institute Course Network: Python Institute Certified Associate in Python Programming (PCAP) core tracking models. [https://pythoninstitute.org/pcap]
7. MIT OCW Video Track: MIT 6.0001 Introduction to Computer Science and Programming in Python series hosted on the MIT OpenCourseWare channel. [https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/]

LIST OF PRACTICALS

1. Write a program to calculate employee net salary based on basic pay, variable tax rates, and allowance conditions.
2. Build a custom user authentication lookup utility using interactive while-loops, input checks, and break parameters.
3. Write a function that accepts variable-length numerical scores (*args) and calculates the optimized running average.

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech (CSE/IT), CSE (ES, AI-IBM, DS, BDCE)

SEMESTER III (2025-2029)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTDSE323	DSE	Python Programming	60	20	20	30	20	2	0	2	3

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

4. Implement a string formatting utility that takes a raw paragraph, filters out punctuation, and returns a unique word-frequency dictionary.
5. Create a class BankAccount with attributes for account number and balance, including methods for deposit() and withdraw().
6. Write a program that implements a class hierarchy where a subclass Manager inherits properties from a base class Employee.
7. Implement a file-parsing utility protected by try-except blocks to open a text file and trap potential file-not-found exceptions gracefully.
8. Create an array of 20 random floating-point configurations using NumPy and replace all elements greater than a threshold with a default boundary value.
9. Write a Pandas data cleaning script that inputs a dirty CSV dataset, drops rows with critical missing items, and groups records by a category index.
10. Construct a dual-subplot visualization using Matplotlib that tracks and compares line plots of monthly business costs versus revenue scales side-by-side.

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech CSE/IT - All Programs

SEMESTER-III (2025-2029)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTDSE322M	DSE	Principle of Programming Language	60	20	20	30	20	2	0	2	3

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

COURSE OBJECTIVES:

The student will have ability to:

1. Improve the background for choosing appropriate programming languages for certain classes of programming problems.
2. Understand principles to program in an imperative (or procedural), an object-oriented, a functional, and a logical programming language.
3. Understand the significance of implementation of a programming language in a compiler or interpreter
4. Increase the ability to learn new programming languages
5. Increase the capacity to express programming concepts and choose among alternative ways to express things.

COURSE OUTCOMES:

Upon completion of the subject, students will be able to:

1. Gain insight and develop understanding of the underlying principles and concepts of programming languages.
2. Competent with analyzing programming language design issues related to data types, expressions and control structures.
3. Describe the concept of sub-programming with the help of Functions. Also develop understanding with the parameter passing techniques and concept of function overloading.
4. Analyze various memory management techniques as well as apply various concepts of object-oriented programming.
5. Develop understanding with the exception handling concepts and gain knowledge of logical and functional programming.

SYLLABUS

UNIT I — 8 HOURS

Preliminary Concepts: Reasons for Studying, Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Programming Paradigms – Imperative, Object Oriented, Functional Programming, Logic Programming.

UNIT II — 8 HOURS

Data Types: Introduction, Primitive, Character, User Defined, Record, Union, Pointer and Reference Types, Design and Implementation Uses Related to these Types. Names, Variable, Concept of Binding

UNIT III — 8 HOURS

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTDSE322M	DSE	Principle of Programming Language	60	20	20	30	20	2	0	2	3

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

Expressions and Statements: Arithmetic Relational and Boolean Expressions, Short Circuit Evaluation Mixed Mode Assignment, Assignment Statements, Control Structures

UNIT IV 8 HOURS

Subprograms and Blocks: Fundamentals of Sub-Programs, Scope and Lifetime of Variable, Static and Dynamic Scope, Design Issues of Subprograms and Operations, Local Referencing Environments, Parameter Passing Methods, Overloaded Sub-Programs, Generic Sub-Programs.

UNIT V 8 HOURS

Abstract Data Types: Abstractions and Encapsulation, Introductions to Data Abstraction, Static and Stack Based Storage Management. Heap Based Storage Management. Garbage Collection. Object Oriented Programming in Smalltalk, C++, Java, C#, Php, Perl.

TEXTBOOKS:

1. R. W. Sebesta, *Concepts of Programming Languages*, 12th ed. Harlow, U.K.: Pearson Education, 2023.
2. D. A. Watt and W. Findlay, *Programming Language Design Concepts*. Hoboken, NJ, USA: Wiley, 2004.

REFERENCE:

1. K. C. Louden and K. A. Lambert, *Programming Languages: Principles and Practices*, 3rd ed. Boston, MA, USA: Cengage Learning, 2011.
2. M. Gabbrielli, S. Martini, and S. Giallorenzo, *Programming Languages: Principles and Paradigms*, 2nd ed. Cham, Switzerland: Springer, 2023.
3. P. Sestoft, *Programming Language Concepts*, 2nd ed. Cham, Switzerland: Springer, 2017.
4. A. B. Tucker and R. E. Noonan, *Programming Languages: Principles and Paradigms*, 2nd ed. New Delhi, India: Tata McGraw-Hill, 2006.
5. T. W. Pratt, M. V. Zelkowitz, and T. V. Gopal, *Programming Languages: Design and Implementation*, 4th ed. Upper Saddle River, NJ, USA: Pearson Education, 2000.

LIST OF PRACTICALS

1. Examine name and scope including static variables in C and call resolution in Java
2. Examine garbage and memory leak in C and develop a mechanism to avoid or detect memory leak
3. Examine assignment operation including assignment of arrays in Java, assignment of lists in Python, and assignment of structures in C
4. Examine goto statement including scope of goto in C, jump into or out of the block, and non-local goto
5. Examine callbacks including callbacks in C and interface and inner classes in Java

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech CSE/IT - All Programs

SEMESTER-III (2025-2029)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTDSE322M	DSE	Principle of Programming Language	60	20	20	30	20	2	0	2	3

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

6. Examine closure in Python and in C
7. Examine functions including variable number of arguments in C, Java, and Python
8. Examine functions including tail recursion, keyword parameters in Python, and stack smashing in C
9. Examine generics including lists (linked lists and array lists), sets (hash set, preset, link hash set), and map
10. Examine inheritance including override in Java, final in Java, multiple inheritance in Python, and downcasting in Java
11. Examine Java thread model, pthread, and Python including racing, synchronization, interthread communication, and thread local storage

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech (CSE/IT) – All Programs

SEMESTER III (2025-2020)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTDSE323	DSE	Go Lang	60	20	20	30	20	2	0	2	3

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

COURSE OBJECTIVES:

The student will have ability:

1. To introduce the design philosophy of Go: Simplicity, Safety, and Concurrency.
2. To master Go's type system, focusing on composition over class-based inheritance.
3. To develop highly concurrent systems using Go routines and Channels.
4. To build production-grade web services and high-performance APIs.
5. To understand the integration of Go with modern DevOps and Cloud tools.

COURSE OUTCOMES:

Upon completion of the subject, students will be able to:

1. Apply Go syntax and memory management concepts to solve computational problems.
2. Design modular software using Structs, Methods, and Implicit Interfaces.
3. Implement concurrent algorithms using Go routines and synchronization primitives (Channels, Wait Groups).
4. Build and test RESTful web services using Go's standard library and middleware logic.
5. Evaluate and optimize Go code performance for cloud-native software development.

SYLLABUS

UNIT I — 6 HOURS

The Go Environment & Foundations: History and Motivation; Installing Go; The go tool chain; Creating the first module; Compilation vs. Interpretation. Primitive Types, Variables, and Zero-values; Constants and iota; Scope rules. Control Structures: if, switch, and the versatile for loop; Logical operators and Bitwise operations.

UNIT II — 6 HOURS

Data Structures & Functional Logic: Arrays vs. Slices: Under-the-hood mechanics of slice headers, dynamic growth, and append logic. Maps (Hash tables) and String manipulation; Pointers: Stack vs. Heap allocation and Garbage Collection basics. Functions: Multiple returns, Variadic parameters, Anonymous functions, and Closures.

UNIT III — 6 HOURS

struts' & The Interface System: Struts': Initialization and Nesting; Methods: Value vs. Pointer receivers; The equivalent in Go. Interfaces: The "Implicit" implementation philosophy; Interface values and the Empty interface (interface {}). Composition: Embedding structs and interfaces to achieve reusability without class-based inheritance.

UNIT IV — 6 HOURS

Mastering Concurrency: Go routines: The M: N scheduler; Lightweight vs. OS threads; sync. Wait Group for

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTDSE323	DSE	Go Lang	60	20	20	30	20	2	0	2	3

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

task synchronization. Channels: Directional channels, Buffered vs. Buffered; The select statement for non-blocking I/O. Advanced Patterns: Worker Pools, Fan-in/Fan-out, Murexes (sync. Mutex), and Context cancellation for API timeouts.

UNIT V — 6 HOURS

Web Services & Production Readiness (9 hours): Error Handling: The Go idiom (error interface, Defer, Panic, Recover); Writing idiomatic Go code. Web Development: net/http package; JSON Marshaling/ Unmarshalling; Middleware design and Routing. Tooling & Testing: Unit testing with go test; Benchmarking; Compilation and Cross-platform builds.

TEXTBOOKS:

1. A. A. A. Donovan and B. W. Kernighan, *The Go Programming Language*. Boston, MA, USA: Addison-Wesley, 2015.
2. W. Kennedy, B. Ketelsen, and E. St. Martin, *Go in Action*. Shelter Island, NY, USA: Manning Publications, 2016.

REFERENCE:

1. K. Bodner, *Learning Go: An Idiomatic Approach to Real-World Go Programming*. Sebastopol, CA, USA: O'Reilly Media, 2021.
2. M. Ryer, *Go Programming Blueprints: Build real-world, high-performance, and robust Go applications*, 2nd ed. Birmingham, UK: Packt Publishing, 2016.
3. T. Tsoukalos, *Mastering Go: Create high-performance concurrent applications and micro services with Go*, 3rd ed. Birmingham, UK: Packt Publishing, 2021.

LIST OF PRACTICALS

1. Workspace setup and implementing a basic CLI tool for mathematical series generation.
2. Implement a "Student Record Search" using Slices and Binary Search logic.
3. Build a "Company Directory" using Maps with functions to add, delete, and list entries.
4. Implement a "Geometry Library" using Interfaces to calculate area/perimeter for different shapes.
5. Simulation of an "Order Processing System" using multiple Goroutines and Wait Groups.
6. Create a "Concurrent Web Scraper" that fetches multiple URLs simultaneously via Channels.
7. Build a basic REST API for a **"Sajag"** Cyber-Alert system where users can submit alerts via JSON.
8. Writing Unit Tests and Benchmarking performance for the API developed in Lab 7.

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech (CSE/ IT) - All Programs

SEMESTER- III (2025-29)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTCS402N	DCC	Software Engineering and Project Management	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

COURSE OBJECTIVES:

The student will have ability to:

1. Get knowledge of basic software engineering methods and practices.
2. Define software requirements and requirement engineering.
3. Apply approaches for various designs and their principles.
4. Explore testing in various domains.
5. Development of significant teamwork and project-based experience.

COURSE OUTCOMES:

Upon completion of the subject, students will be able to:

1. Compare various software process models and identify where these models are applicable.
2. Define and analyze software project management, the framework and the dimensions of software project management.
3. Comprehend System modeling using UML.
4. Identify software testing strategies by using testing tools.
5. Analyze software risks and risk management strategies.

SYLLABUS

UNIT I — 10 HOURS

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.

A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models.

Process models: The waterfall model, incremental process models, evolutionary process models,
the unified process. Agile development-Agile Process, Extreme Programming

UNIT II — 9 HOURS

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTCS402N	DCC	Software Engineering and Project Management	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

System models: Context models, behavioral models, data models, object models, structured methods.

UNIT III — 10 HOURS

Design Engineering: Design Process- Design concepts: Abstraction, Architecture, patterns, Separation of Concerns, Modularity, Information Hiding, Functional Independence, Refinement, Aspects, Refactoring, Object Oriented Design Concepts, Design Classes- Design Model: Data, Architectural, Interface, Component, Deployment Level Design Elements

Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, Use Case Diagrams, Class Diagrams, Interaction Diagrams, State chart Diagrams, Activity Diagrams, Package Diagrams, Component Diagrams, Deployment Diagrams

UNIT IV — 7 HOURS

Software Implementation: - Structured coding Techniques, Coding Styles, Standards and Guidelines, Documentation Guidelines-Modern Programming Language Features: Type Checking-User defined data types-Data Abstraction-Exception Handling, Concurrency Mechanism.

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, Object oriented software, Web Apps-validation testing, system testing, and the art of debugging.

UNIT V — 9 HOURS

Metrics for Process and Products: Software measurement, metrics for software quality. Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, ISO 9000 quality standards.

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

Maintenance: Software Maintenance, Software Supportability, Reengineering, Business Process Reengineering, Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering, Economics of Reengineering

TEXTBOOKS:

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech (CSE/ IT) - All Programs

SEMESTER- III (2025-29)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTCS402N	DCC	Software Engineering and Project Management	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

1. R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 6th ed. New York, NY, USA: McGraw-Hill International Edition, 2005.
2. P. Jalote, *An Integrated Approach to Software Engineering*. New Delhi, India: Narosa Publishing House, 2005.

REFERENCE:

1. R. Mall, *Fundamentals of Software Engineering*, 2nd ed. New Delhi, India: PHI Learning, 2004.
2. I. Sommerville, *Software Engineering*, 7th ed. Harlow, U.K.: Pearson Education, 2004.
3. G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*. Upper Saddle River, NJ, USA: Pearson Education, 1999.
4. R. C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*. Upper Saddle River, NJ, USA: Prentice Hall, 2025.
5. A. Hunt and D. Thomas, *The Pragmatic Programmer: Your Journey to Mastery*, 20th Anniversary ed. Boston, MA, USA: Addison-Wesley, 2019.
6. T. Winters, T. Manshreck, and H. Wright, *Software Engineering at Google: Lessons Learned from Programming Over Time*. Sebastopol, CA, USA: O'Reilly Media, 2020.

LIST OF PRACTICALS

1. Study and compare the SDLC models.
2. Prepare a SRS document in line with the IEEE recommended standards.
3. Study Requirement Engineering of project.
4. Study the UML drawing tools.
5. Draw the Entity relationship diagram of a project.
6. Draw the data flow diagrams at level 0 and level 1.
7. Draw use case diagram in UML.
8. Draw activity diagram in UML.
9. Draw class diagram in UML.
10. Draw the component diagram in UML.
11. Draw sequence diagram in UML.
12. Draw collaboration diagram in UML.
13. Use testing tools such as JUnit.

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech CSE/IT, CSE (ES, MA, AI-IBM, DS, GenAI, MLCC, ICS, BDCE)

SEMESTER-III (2025-2029)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTIT301N	DCC	Computer Networks	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

COURSE OBJECTIVES:

The student will have ability to:

1. Understand the general overview of the concepts and fundamentals of computer networks.
2. Understand the various components required to build different networks.
3. Familiarize the students with the standard models for the layered approach to communication between machines in a network and the protocols of the various layers.

COURSE OUTCOMES:

Upon completion of the subject, students will be able to:

1. Understanding basic computer network technology.
2. Understand the functions of each layer in the OSI and TCP/IP reference model.
3. Obtain the skills of subnetting and routing mechanisms.
4. Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.

SYLLABUS

UNIT I — 9 HOURS

Introduction: Importance of Computer Networks, Classifications & Types. Layered Architecture: Protocol hierarchy, Interfaces and Services, Connection Oriented & Connection less Services, ISO- OSI Reference Model, TCP/IP model overview, comparison of TCP/IP and ISO-OSI reference model.

UNIT II — 9 HOURS

Data Link Layer &MAC Sublayer: Need, Services Provided, Design issues, Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel. Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, MAC Addressing, Binary Exponential Back-off (BEB) Algorithm, Distributed Random Access Schemes/Contention Schemes: for Data Services (ALOHA and Slotted- ALOHA), CSMA, CSMA/CA, CSMA/CD.

UNIT III — 9 HOURS

Network Layer: Need, Services Provided, Design Issues, Routing Algorithms and types of Routing Algorithm, IPv4, IPv6, Classful and classless Addressing, Subnetting, Supernetting.

UNIT IV — 10 HOURS

Transport Layer: Need, Design Issues, Multiplexing and Demultiplexing, transport layer services, UDP,UDP Header Format, Principles of reliable data transfer, TCP, Connection Management, TCP Flow Control, TCP

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
BTIT301N	DCC	Computer Networks	60	20	20	30	20	3	0	2	4

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

Congestion Control, TCP Header Format, TCP Timer Management, SCTP.

UNIT V 8 HOURS

Session layer: Overview, Authentication, Session layer protocols.
Presentation layer: Overview, Data conversion, Encryption and Decryption, Presentation layer protocols (LPP, Telnet, X.25 packet Assembler/Disassembler).
Application Layer: Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, FTP.

TEXTBOOKS:

1. Andrew S Tanenbaum, *Computer Networks*, 6thEd., Pearson Education, 2016.
2. Behrouz A. Forouzan, *TCP/IP-Protocol suite*, 4th Ed., McGraw-Hill, 2010.

REFERENCE:

1. William Stallings, *Data and Computer Communication*, 10th Ed., Pearson, 2014.
2. Comer, *Internet working with TCP/IP Volume one*, Addison-Wesley, 2015.
3. W. Richard Stevens, *TCP/IP Illustrated, Volume 1*, 2nd Ed., Addison-Wesley Professional Computing Series.

LIST OF PRACTICALS

1. Demonstrate Different Types of Network Equipment's.
2. Color coding standard of CAT 5, 6, 7 and crimping of cable in RJ-45.
3. LAN installations and Configurations.
4. Experiment with basic Network configuration commands.
5. Write a program for error detection and correction technique.
6. Write a program for framing.
7. Write a program for routing algorithm.
8. Socket Programming.
9. Study about different network simulators.
10. Establish and simulate peer to peer network using packet tracer.
11. Simulate LAN using hub and switch and discuss pros and cons of hub.
12. Router configuration using packet tracer.

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech CSE (ES-RedHat)

SEMESTER-III (2025-2029)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
RH294	SEC	Red Hat Administration III	0	0	0	0	50	0	0	2	1

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

COURSE OBJECTIVES:

The student will have ability to:

1. Introduce the foundational architecture of Ansible, focusing on agentless configuration management, installation methods, inventory organization, and initial platform configuration.
2. Master the syntax, structure, and execution of Ansible Playbooks using advanced control structures like variables, conditional tasks, loops, and custom Jinja2 templates.
3. Develop scalable automation strategies through structured roles, content collections, and host system administration (users, storage, and networking) while applying industry-standard troubleshooting techniques.

COURSE OUTCOMES:

Upon completion of the subject, students will be able to:

1. Install Ansible, configure control nodes, and manage dynamic or static target host inventories using custom configurations.
2. Write, structure, and execute modular Ansible Playbooks that incorporate complex task logic, variables, secure secret protection, and error-handling conditions.
3. Design modular, reusable automation frameworks using Ansible Roles and Collections to efficiently automate routine enterprise system administration, security patching, and networking tasks.

SYLLABUS

UNIT I — 6 HOURS

Introduction: What is Ansible? Installing Ansible, Building an Ansible Inventory, Managing Ansible Configuration Files.

UNIT II — 6 HOURS

Playbooks: Writing and Running Playbooks. Implementing Multiple Plays. Managing Variables, Managing Secrets, Managing Facts.

UNIT III — 6 HOURS

Task Control: Writing Loops and Conditional Tasks, Implementing Handlers, Handling Task Failure.

File Deployment: Modifying and Copying Files to Hosts, Deploying Custom Files with Jinja2 Templates. Selecting Hosts with Host Patterns, Including and Importing Files.

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech CSE (ES-RedHat)

SEMESTER-III (2025-2029)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
RH294	SEC	Red Hat Administration III	0	0	0	0	50	0	0	2	1

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

UNIT IV — 6 HOURS

Roles: Describing Role Structure, Creating Roles, Deploy Roles from External Content Sources. Getting Roles and Modules from Content Collections, Reuse Content with System Roles

UNIT V — 6 HOURS

Management & Troubleshooting: Managing Users and Authentication, Managing Storage, Managing Network Configuration. Troubleshooting Playbooks, Troubleshooting Ansible Managed Hosts.

TEXTBOOKS:

1. M. Geerling, *Ansible for DevOps: Server and Configuration Management for Humans*, 2nd ed. St. Louis, MO, USA: Midwestern Mac Publishing, 2022
2. J. Freeman, *Ansible: Up and Running: Automating Configuration Management and Deployment the Easy Way*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.

REFERENCE:

1. L. Noire and B. Hudson, *Ansible Enterprise Automation Cookbook: Over 70 Recipes to Automate Infrastructure, Security, and Networking Pipelines*, 1st ed. Birmingham, UK: Packt Publishing, 2024.
2. R. de Jong, *Red Hat Ansible Automation Platform System Administration Guide*, New York, NY, USA: McGraw-Hill, 2023.
3. A. Patel, *Modern Infrastructure as Code: Principles of Ansible, Terraform, and GitOps*, San Francisco, CA, USA: Morgan Kaufmann, 2025.
4. Red Hat, Inc., "Ansible Documentation and Community Guides," 2026. [Online]. Available: https://docs.ansible.com/
5. RHA OLE Platform
6. Ansible Galaxy Community, "Ansible Galaxy Hub: Discover and Share Automation Roles and Collections," 2026. [Online]. Available: https://galaxy.ansible.com/

LIST OF PRACTICALS

1. Ansible Architecture and Control Node Setup: Install Ansible on a Linux-based control node. Configure a static hosts inventory file containing at least two managed target hosts (e.g., development and production nodes).

Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Shri Vaishnav Institute of Information Technology

Choice Based Credit System (CBCS) in the light of NEP-2020

B. Tech CSE (ES-RedHat)

SEMESTER-III (2025-2029)

COURSE CODE	CATEGORY	COURSE NAME	TEACHING & EVALUATION SCHEME					L	T	P	CREDITS
			THEORY			PRACTICAL					
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
RH294	SEC	Red Hat Administration III	0	0	0	0	50	0	0	2	1

Legends: **L** - Lecture; **T** - Tutorial/Teacher Guided Student Activity; **P** – Practical; **C** - Credit; ***Teacher Assessment** shall be based following components: Quiz/Assignment/ Project/Participation in Class, given that no component shall exceed more than 10 marks.

2. Configuration and Connectivity Testing: Create a customized ansible.cfg file in your project directory. Use the Ansible ad-hoc ping and setup modules to test SSH connectivity and pull hardware configurations from your managed hosts.
3. Multi-Play Blueprint Execution: Write a multi-play Ansible Playbook. The first play must install a basic web server (e.g., Apache/Nginx) on a web-host group, and the second play must install a database server on a db-host group.
4. Dynamic Data with Variables and Secrets: Create a playbook that uses host-specific variables to deploy software. Use ansible-vault to encrypt a file containing sensitive passwords (secrets), and execute the playbook by securely passing the vault password at runtime.
5. Conditional Logic, Loops, and Handlers: Write a playbook that uses a loop to create multiple system users simultaneously. Use a when conditional statement to run tasks only on specific operating system families, and implement a handler to restart a service only when its configuration changes.
6. Error Handling and Fault Tolerance: Build a playbook designed to manage a fragile task (e.g., downloading a remote asset). Implement block, rescue, and always sections to gracefully handle a simulated task failure without breaking the entire automation run.
7. Dynamic Configuration with Jinja2: Use a Jinja2 template (.j2) to generate a customized message-of-the-day (/etc/motd) file on target hosts. The template must dynamically read target system facts (like IP address, OS version, and total memory) and inject them into the deployed file.
8. Creating Custom Structural Roles: Organize your automation code by building a custom Ansible Role from scratch using the standard directory structure (tasks/, handlers/, vars/, defaults/, templates/). Use the role to automate a multi-step application deployment.
9. Galaxy and Collections Integration: Download and configure an official community role from Ansible Galaxy. Use an external System Role or Content Collection within a playbook to standardize system time configurations (NTP) across your entire infrastructure.
10. System Automation and Playbook Debugging: Write a comprehensive playbook that configures storage partitions, manages local user accounts, and adjusts network interfaces on target hosts. Intentionally introduce syntax and runtime errors, then utilize ansible-playbook --syntax-check, the debug module, and the -vvv (verbose) flag to systematically troubleshoot and fix the environment.