AC – 28/03/2025
Item No. – 6.9 (N)

As Per NEP 2020

University of Mumbai

Syllabus for Major Vertical – 1 & 4		
Name of the Programme – B.Sc. (Information Technology)		
Faulty of Science and Technology		
Board of Studies in Information Technology		
U.G. Second Year Programme	Exit Degree	U.G. Diploma in Information Technology
Semester		III & IV
From the Academic Year		2025-26

University of Mumbai

(As per NEP 2020)

Sr. No.	Heading	Particulars
1	**Title of program** **O:_____**	**B.Sc. (Information Technology)**
2	**Exit Degree**	**U.G. Diploma in Information Technology**
3	**Scheme of Examination** **R: _____**	NEP 40% Internal 60% External, Semester End Examination Individual Passing in Internal and External Examination
4	**Standards of Passing** **R: _____**	**40%**
5	**Credit Structure** **R. SU-510C** **R. SU-510D**	**Attached herewith**
6	**Semesters**	**Sem. III & IV**
7	**Program Academic Level**	**5.00**
8	**Pattern**	**Semester**
9	**Status**	**New**
10	**To be implemented from Academic Year**	**2025-26**

Sd/-	Sd/-	Sd/-
Sign of the BOS Chairman **Dr. Srivaramangai R** **BOS in Information Technology**	**Sign of the Offg. Associate Dean** **Dr. Madhav R. Rajwade** **Faculty of Science & Technology**	**Sign of the Offg. Dean** **Prof. Shivram S. Garje** **Faculty of Science & Technology**

Under Graduate Diploma in Information Technology

Credit Structure (Sem. III & IV)

(B. Sc.)- Major & Minor

	R. SU-510C									
Level	**Semester**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC,RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
	III	8 Python Programming DATA STRUCTURES Operating System Major Practical 3	…	4	2	VSC:2, APPLIED MATHEMATICS	AEC:2	FP: 2 CC:2	22	
5.0	**R. SU-510D**									**UG Diploma 88**
	IV	8 Core Java Software Engineering Computer Networks Major Practical 4	…	4	2	SEC:2 Computer Graphics OR Mobile Programming	AEC:2	CEP: 2 CC:2	22	
	Cum Cr.	28		10	12	6+6	8+4+2	8+2+2	88	
Exit option; Award of UG Diploma in Major and Minor with 88 credits and an additional 4 credits core NSQF course/ Internship OR Continue with Major and Minor										

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

Sem. - III

Vertical – 1
Major

Syllabus
B.Sc. (Information Technology)
(Sem.- III)

Title of Paper: Python Programming

Sr.No.	Heading	Particulars
1	**Description the course: Including but not limited to:**	**Introduction to Programming with Python** course is designed to help beginners learn Python, a versatile and beginner-friendly language known for its simplicity and readability. Python is an excellent choice for newcomers to programming due to its clear syntax and broad applications in fields like web development, data analysis, and artificial intelligence. In today's technology-driven world, programming skills are increasingly essential, and Python's popularity has surged due to its ease of use and extensive support community. Python is also a gateway language, allowing learners to transition easily into more advanced topics such as machine learning, data science, and web development. As an interpreted, high-level language, Python is particularly relevant across industries like technology, healthcare, finance, and academia, making Python proficiency a highly sought-after skill. The course focuses on core programming concepts like syntax, data structures, and control flow, ensuring that learners can write efficient and functional code. The course also encourages further learning, serving as a stepping stone for advanced Python courses or specialized areas like machine learning and web development. Python's beginner-friendly nature and expansive libraries make it an enjoyable language to learn, fostering both interest and engagement. By combining theory with hands-on projects, the course aims to spark curiosity and provide learners with tangible results from their efforts. As learners gain proficiency in Python, they will have the tools to tackle more complex programming challenges, making this course an invaluable starting point for anyone interested in programming or pursuing a career in tech. **Demand in the Industry:** Python's popularity in the industry is soaring. Professionals proficient in Python are in high demand across various sectors, including technology, finance, healthcare, and academia. Completion of this Course opens doors to entry-level positions in software development, quality assurance, data analysis, and scripting.
2	**Vertical:**	Major
3	**Type:**	Theory
4	**Credits:**	2 credits (1 credit = 15 Hours for Theory in a semester, Total 30 hours)
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50

7	**Course Objectives (CO):** CO 1.Master the core features of Python, including its execution model and a wide range of data types. CO 2. Develop proficiency in control flow by working with conditional statements, loops and other control structures. CO 3. Work efficiently with arrays, strings, and complex data structures, leveraging Python's capabilities for data manipulation. CO 4. Apply functions, modules, and string operations to solve real-world programming problems with flexibility and ease. CO 5. Manage file operations, utilize regular expressions, and handle date and time functions for comprehensive Python programming tasks.	
8	**Course Outcomes (OC):** OC 1. Demonstrate mastery of Python features to tackle a wide range of programming challenges. OC 2. Utilize control flow statements to ensure accurate and logical program execution. OC 3. Efficiently manipulate arrays, strings, and data structures to enhance. data handling and problem-solving. OC 4. Design modular, efficient programs by leveraging functions, modules, and string operations. OC 5. Manage file operations, employ regular expressions, and manipulate date and time data to improve program functionality and performance.	
9	**Module 1:** **Basic Elements of Python Programming:** Features of Python, Execution of a Python Program, Python Interpreter, Comments, IDLE, Data types, Dictionary, Sets, Mapping, Basic Elements of Python, Variables, Input Function, Output Statements, Command Line Arguments. Operators, Precedence of Operators, Associativity of Operators **Control Statements:** The if statement, The if … else Statement, The if … elif … else Statement, Loop Statement- while loop, for loop, Infinite loop, Nested loop, The else suite, break statement, continue statement, pass statement, assert statement, return statement. **Arrays:** Creating Arrays, Indexing and Slicing of Arrays, Basic Array Operations, Arrays Processing, Mathematical Operations on Array, Aliasing Arrays, Slicing and Indexing in NumPy Arrays, Basic slicing, Advanced Indexing, Dimensions and Attributes of an Array **Functions:** Function definition and call, Returning Results, Returning Multiple Values from a Function, Built-in Functions, Difference between a Function and a Method, Pass Value by Object Reference, Parameters and Arguments, Recursive Functions, Anonymous or Lambda Functions. Modules in Python. Strings: Creating Strings, Functions of Strings, Working with Strings, Formatting Strings, Finding the Number of Characters and Words, Inserting Substrings into a String.	15 Hrs

	Module 2:	
	List: Exploring List, Tuples and Dictionaries: Lists, List Functions and Methods, List Operations, List Slices, Nested Lists, Tuples, Functions in Tuple. **Working with Dictionaries:** Creating a Dictionary, Operators in Dictionary, Dictionary Methods, Using for Loop with Dictionaries, Operations on Dictionaries **Files in Python:** Opening and Closing a File, Working with Text Files, , Working with Binary Files, The 'with' statement, Pickle in Python, The seek() and tell() Methods, Random Accessing of Binary Files, Zipping and Unzipping Files, Working with Directories **Regular Expressions:** Introduction, Sequence Characters in Regular Expressions, Special Characters in Regular Expressions, Using Regular Expression on Files, Retrieving Information from an HTML File **Date And Time in Python:** Time, Date, Date and Time Now, combining date and times, formatting date and time, Finding and comparing dates, Sorting dates, Knowing the Time taken by a Program, Working with Calendar Module	15 Hrs
10	**Books and References:** **Textbooks** 1. Learning Python, Fourth Edition by Mark Lutz Copyright © 2009 Mark Lutz. Published by O'Reilly Media, Inc. 2. Python Basics: A Practical Introduction to Python 3 Revised and Updated 4th Edition David Amos, Dan Bader, Joanna Jablonski, Fletcher Heisler **Reference Books** 1. Let Us Python, Yashwant. B. Kanetkar, BPB Publication, 2019 2. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018 3. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	Format of Question Paper: External Examination (30 Marks)– 1 hr duration
14	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks) Or Q1: Attempt any three (out of five) from Module 1 (15 marks) Q2: Attempt any three (out of five) from Module 2 (15 marks)	

Title of Paper: DATA STRUCTURES

Sr.No.	Heading	Particulars
1	**Description the course: Including but Not limited to:**	Data Structures is a fundamental subject that focuses on the organization, storage, and manipulation of data. It provides the tools and techniques to efficiently manage and process data, forming the backbone of algorithms and software development.
2	**Vertical:**	Major
3	**Type:**	Theory
4	**Credits:**	2 credits (1 credit = 15 Hours for Theory in a semester, Total 30 hours)
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** 1. To understand the fundamental concepts of data structures and their applications. 2. To analyze the efficiency of algorithms and operations on data structures. 3. To provide practical exposure to implementing data structures in programming. 4. To understand the properties and applications of arrays, linked lists, stacks, and queues. 5. To translate data structure concepts into working code using a programming language. 6. To apply data structures to solve real-world problems like searching and sorting. 7. To grasp the structure and traversal methods of binary trees and binary search trees.	
8	**Course Outcomes (OC):** **Students will be able to:** OC 1. Demonstrate knowledge of core data structures and their operations OC 2. Analyze the time and space complexity of algorithms and choose the most efficient solution for a given problem. OC 3. Translate algorithmic solutions into correctly functioning code using their chosen programming language. OC 4. Implement and traverse binary trees and binary search trees, demonstrating their understanding of these structures.	
9	**Module 1:**	
	1. Introduction Basic terminology: data, information, data structure, abstract data type (ADT) Classification of data structures: linear, non-linear Algorithm analysis: time complexity, Big O notation **2. Arrays and Linked Lists** Array representation and operations (traversal, insertion, deletion, searching) Linked lists: singly linked lists (representation, insertion, deletion, traversal) Comparison of arrays and linked lists, advantages and disadvantages. **3. Stacks and Queues** Stack ADT: push, pop, peek operations Array implementation of stacks Applications of stacks: expression evaluation (infix to postfix conversion) Queue ADT: enqueue, dequeue, peek operations Array implementation of queues Applications of queues: basic scheduling scenarios **4. Recursion** Concept of recursion, base case, recursive step Examples: factorial, Fibonacci sequence	15 Hrs

	Module 2:	
	1.Trees Binary trees: representation, traversal (inorder, preorder, post order) Binary search trees: insertion, deletion, search Applications of trees: basic hierarchical data representation **2.Hashing** Hash functions and hash tables Collision handling: separate chaining Applications of hashing: dictionaries **3. Sorting and Searching** Sorting algorithms: bubble sort, insertion sort, selection sort Searching algorithms: linear search, binary search	**15 Hrs.**
10	**Books and References:** 1. Data Structures and Algorithms made Easy: Data Structures and Algorithmic Puzzles, Narasimha Karumanchi ,5th Edition 2017 2. A Simplified Approach to Data Structures, Lalit Goyal, Vishal Goyal, Pawan Kumar SPD,1st 2014 3. Problem Solving in Data Structures & Algorithms Using C by Hemant Jain ,1st Edition, BPB Publications, 2018 **4.** Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 4th Edition, MIT Press,2022	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 Hr. duration**
14	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks) Or Q1: Attempt any three (out of five) from Module 1 (15 marks) Q2: Attempt any three (out of five) from Module 2 (15 marks)	

Title of Paper: Operating System

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	Introduce operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file systems and protection) Introduce the issues to be considered in the design and development of operating system (memory, file and disk).
2	**Vertical :**	**Major**
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester, Total 30 hours)
5	**Hours Allotted :**	30
6	**Marks Allotted:**	30
7	**Course Objectives(CO):** 1. Understand basic knowledge of computer operating system 2. structures and functioning. 3. Understand the process management mechanism 4. CO 3. Ability to apply CPU scheduling algorithms to manage tasks. 5. CO 4. Discuss methods of prevention and recovery from system deadlock 6. CO 5. Understand the implementation of file systems and directories	
8	**Course Outcomes (OC):** 1. Outline the basic concept of operating systems 2. Analyze the working of operating system 3. Examine the working of various scheduling approaches 4. Apply the concepts of synchronization and deadlock 5. Apply the file access mechanisms	
9	**Modules:-** **Module 1:**	
	Operating System Overview: Basics of operating systems: Generations, Types, Structure, Services, System Calls, System Boot, System Programs, Protection and Security. **Process Management**: Process Concepts, Process States, Process Control Block, Scheduling-Criteria, Scheduling Algorithms and their Evaluation, Threads, Threading Issues. **Process Synchronization:** Background, Critical-Section Problem, Peterson's Solution. Synchronization Hardware, Semaphores, Classic Problems of Synchronization.	**15 Hrs**
	Module 2:	
	Memory Management: Main Memory, Swapping, Contiguous Memory Allocation, Paging, Structure of Page Table, Segmentation, Virtual Memory, Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing. **Deadlock:** System Model, Deadlock Characterization, Deadlock Prevention, Detection and Avoidance, Recovery from Deadlock. **File System Interface:** File Concept, Access Methods, Directory Structure, and File System Structure.	**15 Hrs**

10	**Books and Reference:** 1. Operating Systems – Internals and Design Principles William Stallings, Pearson 9th , 2009 2. Operating System Concepts, Abraham Silberschatz, Wiley, 8th Edition 3. Operating Systems, Godbole and Kahate, Godbole and Kahate, 3rd Edition.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks) Or Q1: Attempt any three (out of five) from Module 1 (15 marks) Q2: Attempt any three (out of five) from Module 2 (15 marks)	

Title of Paper: Major Practical 3

Sr.No.	Heading	Particulars
1	**Description the course: Including but not limited to:**	This course offers a comprehensive exploration of advanced Python programming concepts, designed to equip students with the tools to tackle real-world problems efficiently. It covers a range of topics, including text processing with regular expressions to identify patterns and extract meaningful data, as well as file handling techniques for both text and binary files. Students will also gain expertise in manipulating and comparing dates using Python's built-in date and time modules, along with performing calendar-based operations. The course emphasizes performance optimization by teaching students how to measure and improve program execution time. Additionally, students will learn how to extract structured data, such as hyperlinks from HTML files, and apply these techniques in practical scenarios. By the end of the course, students will be adept at solving complex problems, optimizing their Python solutions, and utilizing advanced programming concepts to handle diverse data processing tasks.
2	**Vertical:**	Major
3	**Type:**	Practical
4	**Credits:**	2 credits (30 Hours of Practical work in a semester)
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** 1. Understand fundamental programming concepts in Python, including input/output operations, conditional statements, and loops. 2. Understand and apply array operations, indexing, slicing, and mathematical functions using NumPy. 3. Develop problem-solving skills by using functions, recursive logic, lambda expressions, and modular programming. 4. Use data structures like lists and dictionaries and perform file operations. 5. Work with text processing, file handling, date manipulation, and performance analysis using advanced Python programming concepts 6. To provide hands-on experience in implementing fundamental data structures like arrays, linked lists, stacks, queues, trees, and graphs. 7. To develop skills in algorithm design and analysis for solving computational problems using data structures. 8. To enable students to choose appropriate data structures for different applications and justify their choices. 9. To enhance understanding of dynamic memory allocation and efficient data management techniques. 10. To equip students with the ability to debug and optimize code for data structure operations.	
8	**Course Outcomes (OC):** . OC 1. Apply Python programming concepts like input/output, conditional statements, and loops, to solve fundamental problems effectively. . OC 2. Demonstrate proficiency in performing basic operations, indexing, slicing, and analyzing attributes of arrays using NumPy. . OC 3. Apply functions, recursion, and lambda expressions to solve computational problems, and implement modular programming for reusable and efficient code design.	

	. OC 4. Implement lists and dictionaries, perform file operations, and use functions to solve real-world problems effectively. . OC 5. Process text, extract information, handle dates, and measure execution time for solving complex real-world problems. . OC 6. Ability to implement and manipulate basic and advanced data structures to solve real-world problems. . OC7 Proficiency in writing efficient algorithms using suitable data structures for operations like searching, sorting, and traversal. . OC8 Capability to analyze the time and space complexity of algorithms for various data structures. . OC9 Enhanced problem-solving skills by applying data structures in different domains such as databases, networks, and operating systems	
9	**Module 1** 1. Write programs for the following: a. Write a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old. b. Write a program to accept a number from the user and depending on whether the number is even or odd, print out an appropriate message to the user. c. Write a program to accept the SGPI from the user and print corresponding grade based on the following: d. SGPI Grade e. 9.00 – 10.00 O f. 8.00 – 8.99 A+ g. 7.00 – 7.99 A h. 6.00 – 6.99 B+ i. 5.50 – 5.99 B j. 5.00 – 5.49 C k. 4.00 – 4.99 P l. Below 4 F 2. Write programs for the following: a. d. Write a program to generate the Fibonacci series. b. e. Write a program to accept a number from the user display sum of its digits. 3. Write programs for the following: a. Write a program to perform basic operations, indexing and slicing on arrays. b. Write a program to implement mathematical functions on arrays. c. Write a program to perform array aliasing and copying. 4. Write programs for the following: a. Write a program to perform slicing, basic and advanced indexing on b. NumPy arrays. c. e.Write a program to analyze dimensions and attributes of arrays 5. Write programs for the following: a. Write a function to check the input value is Armstrong and also write the function for Palindrome. b. Write a recursive function to print the factorial for a given number. c. Write a lambda function that checks whether a given string starts with a specific character. 6. Write programs for the following: a. Write a program to compute number of characters and words in a string.	30 Hrs

	b. Create a file geometry.py to calculate base areas for shapes square and circle. In another file, write a function pointyShapeVolume(x, y, squareBase) that calculates the volume of a square pyramid if squareBase is True and of a right circular cone if squareBase is False. x is the length of an edge on a square if squareBase is True and the radius of a circle when squareBase is False. y is the height of the object. First use squareBase to distinguish the cases. Use the circleArea and squareArea from the geometry module to calculate the base areas. 7. Write programs for the following: a. Write a program that takes two lists and returns True if they have at least one common member. b. Write a Python script to sort (ascending and descending) a dictionary by value. 8. Write programs for the following: a. Write a program to accept and pass radius to a function that returns area and circumference (using tuple). b. Write a program to perform basic file operations on text files and binary files. c. Write a Python program to read last n lines of a file. 9. Write programs for the following: a. a.Write a program to count the occurrences of a specific word in a file using regular expressions. b. b.Write a program to extract all hyperlinks (<a href="...">) from an HTML file. 10. Write programs for the following: a. Write a program that compares two dates (in DD/MM/YYYY format) and prints which one is earlier. b. Write a program to measure program execution time. c. Write a program using the calendar module to print the weekday of the first day of a given month and year.	
	Module 2	30 Hrs
	1. Array Operations: Write a program to implement basic array operations: a. Insert an element at a specific position in an array. b. Delete an element from a specific position in an array. c. Search for an element in an array (linear search). 2. Linked List Manipulation: Write a program to: a. Create a singly linked list. b. Insert a node at the beginning, end, and at a given position in a linked list. c. Delete a node from a given position in a linked list. 3. Stack Application: Write a program to: a. Implement a stack using an array. b. Convert an infix expression to postfix notation using a stack. 4. Queue Application: Write a program to: a. Implement a queue using an array. b. Simulate a simple queuing system (e.g., customer service queue). 5.Binary Search Tree: Write a program to: a. Create a binary search tree. b. Insert nodes into a binary search tree. c. Search for a node in a binary search tree. 6. Tree Traversal: Write a program to: a. Implement pre-order, b. in-order,	

	c. Post-order traversal of a binary tree. 7.Hash Table: Write a program to: a. Implement a hash table with separate chaining for collision handling. b. Store and retrieve data from the hash table. 8.Sorting Algorithms: Write programs to implement and compare the following sorting algorithms: a. Bubble sort b. Insertion sort c. Selection sort 9.Searching Algorithms: Write programs to implement and compare: a. Linear search b. Binary search (on a sorted array) 10.Combined Application a. Design a simple program that uses multiple data structures .	
10	Text Books: 1. Learning Python, Fourth Edition by Mark Lutz Copyright © 2009 Mark Lutz. Published by O'Reilly Media, Inc. 2. Python Basics: A Practical Introduction to Python 3 Revised and Updated 4th Edition David Amos, Dan Bader, Joanna Jablonski, Fletcher Heisler 3. Data Structures and Algorithms made Easy: Data Structures and Algorithmic Puzzles, Narasimha Karumanchi ,5th Edition 2017	
11	Reference Books: 1. Let Us Python, Yashwant. B. Kanetkar, BPB Publication, 2019 2. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018 3. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017 4. A Simplified Approach to Data Structures, Lalit Goyal, Vishal Goyal, Pawan Kumar SPD,1st 2014 5. Problem Solving in Data Structures & Algorithms Using C by Hemant Jain ,1st Edition, BPB Publications, 2018 6. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 4th Edition, MIT Press,2022	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	