# NIRMALA MEMORIAL FOUNDATION COLLEGE OF COMMERCE AND SCIENCE (AUTONOMOUS)

*NAAC Reaccredited 'B++' Grade*

Affiliated to the

**UNIVERSITY OF MUMBAI**

**Program: Bachelor of Science (Information Technology)**

**A -U.G. Certificate in Information Technology 2025-26**

**B - U.G. Diploma in Information Technology 2026-27**

**C- Degree-B.Sc. (Information Technology) 2027-28**

**D - B.Sc. (Honours) in Information Technology 2028-29**

**E - B.Sc. (Honours with Research) in Information Technology 2027-28**

## S.Y.B.Sc. (Information Technology)

## Semester III and IV

**Choice Based Credit System (CBCS) with effect from the Academic year 2025-2026**

**Academic Council No:**

**Agenda No:**

**NIRMALA MEMORIAL FOUNDATION COLLEGE OF COMMERCE AND SCIENCE (AUTONOMOUS)**

**(As per NEP 2020)**

| Sr. No. | Heading | Particulars |
|---|---|---|
| 1 | Title of program<br>O: | B.Sc. (Information Technology) |
| 2 | Exit Degree | U.G. Diploma in Information Technology |
| 3 | Scheme of Examination R: | NEP<br>40% Internal<br>60% External, Semester End Examination Individual Passing in Internal and External Examination |
| 4 | Standards of Passing R: | 40% |
| 5 | Credit Structure<br>R. SU-510C<br>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 |

NIRMALA MEMORIAL FOUNDATION COLLEGE OF COMMERCE AND SCIENCE (AUTONOMOUS)


# Syllabus for
# Vertical – 1 & 4

| | | |
|---|---|---|
| **Name of the Programme – B.Sc. (Information Technology)** | | |
| **Faculty 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** |

# Preamble

**Introduction**

Information technology (IT) continues to be a dynamic and rapidly evolving field with high demand for skilled professionals. The demand for IT workers is driven by various factors, and the landscape may have evolved over a period of time. NEP envisages the multidisciplinary approach thus making IT much more applicable in all fields of life. This facilitates multi-institutional mobility of the students within India as well as abroad thus making the students attain different proficiency levels right from certificate to B.Sc Honours with Research. This new syllabus under NEP will thus enables the students for higher education, research and career in the field of IT

**Aims and Objectives**

The aims and objectives of a Bachelor of Science (B.Sc) program in Information Technology (IT) generally revolve around providing students with a comprehensive understanding of the principles, technologies, and applications within the field of information technology. The entire program collectively aim to produce graduates who are well-rounded IT professionals, capable of contributing to the design, development, and management of information technology systems in various industries. The specific details of the curriculum may vary among institutions offering B.Sc in Information Technology programs.

**Program Outcome**

- **Knowledge of Business Disciplines**

Develop a strong foundation in core areas of commerce such as accounting, economics, business law, management, and statistics, enabling students to understand and respond to contemporary business challenges.

- **Problem-Solving and Decision-Making Skills**

Apply logical reasoning and quantitative techniques to analyze business problems and make informed decisions in diverse organizational settings.

- **Communication and Interpersonal Skills**

Demonstrate proficiency in written, verbal, and digital communication, essential for effective collaboration and negotiation in business environments.

- **Ethical, Social, and Environmental Awareness**

Recognize ethical issues, social responsibilities, and environmental sustainability in business operations and decision-making.

- **Lifelong Learning and Adaptability**

Develop the ability to adapt to a dynamic global business environment by engaging in continuous learning, embracing technology, and cultivating personal and professional growth.

## Program Specific Outcome (PSO)

- **Technical Proficiency:**

Demonstrate a comprehensive understanding of fundamental concepts, principles, and technologies in information technology. Apply programming and software development skills to design and implement IT solutions.

- **System Thinking and Analysis:**

Apply system analysis and design methodologies to analyze and address complex problems. Design and develop IT systems that meet user requirements and organizational needs.

- **Database Management:**

Design, implement, and manage relational databases to store and retrieve information effectively. Demonstrate proficiency in using database management systems and querying languages.

- **Networking and Security:**

Understand and implement computer networks, protocols, and security measures.
Evaluate and implement security solutions to protect information systems.

- **Web Technologies:**

Develop web applications using a variety of technologies and programming languages.
Design and create user interfaces that adhere to web design principles.

- **Project Management:**

Apply project management principles to plan, execute, and deliver IT projects.
Demonstrate the ability to work effectively within project teams.

- **Emerging Technologies:**

Stay informed about and adapt to emerging technologies in the IT field.
Apply concepts of artificial intelligence, machine learning, cloud computing, and IoT to solve real-world problems.

- **Critical Thinking and Problem-Solving:**

Analyze and solve complex IT problems using critical thinking skills.
Apply problem-solving strategies to troubleshoot and resolve technical issues.

- **Communication Skills:**

Effectively communicate technical information to diverse audiences, both orally and in writing. Collaborate with team members and stakeholders to achieve common goals.

- **Ethics and Professionalism:**

Demonstrate ethical behavior and professionalism in all aspects of the IT profession.
Adhere to ethical standards and legal considerations related to information technology.

**NIRMALA MEMORIAL FOUNDATION COLLEGE OF COMMERCE AND SCIENCE (AUTONOMOUS)**

**(Credit Structure Semester III & IV)**

| | | | Semester- III | | Semester- IV | |
|---|---|---|---|---|---|---|
| **Verticals** | Type | | **Course** | **Credits** | **Course** | **Credits** |
| **Vertical 1** | **Major Subjects** | MJ1 | Python for AI | 2 | Core Java Techniques | 2 |
| | | MJ2 | Data Structures for Problem Solving | 2 | Software Engineering | 2 |
| | | MJ3 | Operating System | 2 | Computer Networks | 2 |
| | | MJ4 | Python for AI + Data Structures for Problem Solving Practical | 2 | Core Java Techniques + Computer Networks Practical | 2 |
| **Vertical 2 (ANY ONE BASED ON MAJOR)** | **Minor Subject** | MN 1 | DS_Python for Data Science | 2 | DS_Data Analysis with SAS/SPSS/R | 2 |
| | | MN 2 | DS_Scala for Data Science | 2 | Advance Python for Data Science | 2 |
| **Vertical 3** | **Open Elective** | OE1 | Financial Literacy | 2 | Management Skills | 2 |
| **Vertical 4 (ANY** | **SEC** | ** | *** | ** | Arduino Programmin | 2 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **ONE BASED ON MAJOR)** | | | | | g Practical | |
| | **VSC** | | Applied Mathematics | 2 | *** | ** |
| **Vertical 5** | **AEC** | | ह दिं ी भाषा : व्याव ारिक प्रयोग | 2 | Technical writing in English | 2 |
| **Vertical 6** | **CC** | | Sports -I | 2 | Sports -II | 2 |
| | **FP** | | Field Project | 2 | Community Engagement Project | 2 |
| | | | **Total Credits** | **22** | **Total Credits** | **22** |

**[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 ]**

NIRMALA MEMORIAL FOUNDATION COLLEGE OF COMMERCE AND SCIENCE (AUTONOMOUS)

# B.Sc. (Information Technology)

# Semester - III

# Vertical – 1
# Major

## Name of the Course: Python Programming for AI

| Sr.No. | Heading | Particulars |
|---|---|---|
| 1 | **Description the course :**<br>**Including but Not limited to:** | This course introduces Python, a simple yet powerful language widely used in web development, data analysis, and AI. Students will learn core programming concepts like syntax, control flow, and data structures through hands-on practice. Python's ease of use and high demand across industries make this course a valuable starting point for careers in tech, data, 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):**<br><br>**CO1.** Gain a solid understanding of Python's core features, execution model, and built-in data types.<br>**CO1.**Strengthen problem-solving skills through effective use of control structures like conditionals and loops.<br>**CO1.**Handle arrays, strings, and structured data efficiently using Python's rich data manipulation features.<br>**CO1.**Utilize functions, modules, and string processing techniques to design flexible and reusable code.<br>**CO1.**Perform file handling, apply regular expressions, and work with date and time modules to build complete Python solutions. | |
| 8 | **Course Outcomes (OC):**<br><br>**OC1.** Exhibit proficiency in Python's core features to solve diverse programming problems effectively.<br>**OC2**. Apply control flow constructs to develop logically correct and efficient programs.<br>**OC3**. Skilfully manipulate arrays, strings, and complex data structures for enhanced data processing.<br>**OC4**. Create modular and optimized programs using functions, modules, and advanced string handling. | |

| | | |
|---|---|---|
| | **OC5.** Perform file management, utilize regular expressions, and handle date/time operations to build robust applications. | |
| **9** | **Modules:-**<br>**Module 1:** | |
| | **Basic Elements of Python Programming:**<br>Features of Python, Execution of a Python Program, Python Interpreter, Comments, IDLE, Data types, Dictionary, Variables, Input Function, Output Statements, Operators, Precedence of Operators.<br>**Control Statements:**<br>Conditional statements (if, if-else, if-elif-else, nested if), Loops in python (while, for, nested loops), Loop manipulation statements (continue, pass, break, else)<br>**Arrays:**<br>Creating Arrays, Indexing and Slicing of Arrays, Basic Array Operations, Array Indexing in NumPy Arrays<br>**Functions:**<br>Function definition and call, Returning Results, Returning Multiple Values from a Function, Built-in Functions Pass Value by Object Reference, Parameters and Arguments, Recursive Functions, Anonymous or Lambda Functions, Modules in Python. **Strings:**<br>Creating Strings, Working with String functions, Formatting Strings | **15 Hrs** |
| | **Module 2:** | |
| | **List:**<br>Exploring List, Tuples and Dictionaries:<br>List Functions and Methods, List Operations, List Slices, Nested Lists, Tuples Functions, creating a Dictionary, Operators in Dictionary, Dictionary Methods, Using for Loop with Dictionaries, Operations on Dictionaries<br>**Files in Python:**<br>Opening and Closing a File, Working with Text Files and Binary Files, The 'with' statement, The seek() and tell() Methods, Random Accessing of Binary Files, Zipping and Unzipping Files.<br>**Regular Expressions:**<br>Introduction, Sequence Characters in Regular Expressions, Special Characters in Regular Expressions, Using Regular Expression on Files,<br>Retrieving Information from HTML File<br>**Date And Time in Python:**<br>Time, Date, Date and Time Now, combining date and times, | **15 Hrs** |

| | | |
|---|---|---|
| | formatting date and time, Working with Calendar Module<br><br>Introduction: What is Artificial Intelligence? Foundations of AI, History, the<br>state of art AI today. | |
| **10** | **Books and References:**<br>**Textbooks**<br>1. Learning Python, Fourth Edition by Mark Lutz Copyright © 2009 Mark Lutz. Published by O'Reilly Media, Inc.<br>2. Python Basics: A Practical Introduction to Python 3 Revised and Updated 4th Edition David Amos, Dan Bader, Joanna Jablonski, Fletcher Heisler<br>**Reference Books**<br>1. Let Us Python, Yashwant. B. Kanetkar, BPB Publication, 2019<br>2. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018<br>3. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017 | |
| **11** | **Internal Continuous Assessment: 40%** | **Semester End Examination: 60%** |
| **12** | **Continuous Evaluation through:**<br>Class test: 15 marks<br>Quizzes/ Presentations/<br>Assignments: 5 marks<br>Total: 20 marks | **Format of Question Paper: External Examination (30 Marks)– 1 hr duration** |
| **13** | **Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)**<br>Q1: Attempt any three (out of five) from Module 1 (15 marks)<br>Q2: Attempt any three (out of five) from Module 2 (15 marks) | |

## Name of the Course: Data Structures for Problem Solving

| Sr.No. | Heading | Particulars |
|---|---|---|
| 1 | **Description the course :**<br>**Including but Not limited to:** | Data Structures is a core subject that deals with the way data is organized, stored, and manipulated. It equips learners with methods and strategies to efficiently handle and process data, serving as a foundation for designing algorithms and building software applications. |
| 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 Hr |
| 6 | **Marks Allotted:** | 50 |
| 7 | **Course Objectives(CO):**<br><br>**CO1**.To develop the ability to evaluate the performance and efficiency of data structure operations and algorithms.<br>**CO2**.To offer hands-on experience in implementing common data structures through programming.<br>**CO3**.To explore the characteristics, use cases, and implementation of arrays, linked lists, stacks, and queues.<br>**CO4**.To enable students to convert theoretical data structure concepts into executable cod.<br>**CO5**.To apply suitable data structures for solving standard computational problems such as searching and sorting.<br>**CO6**.To understand the organization, properties, and traversal techniques of binary trees and binary search trees. | |
| 8 | **Course Outcomes (OC):**<br><br>**OC1.** Upon successful completion of this course, students will be able to:<br>**OC2**. Understand and apply fundamental data structures and their associated operations in problem-solving.<br>**OC3**. Analyze algorithms to assess their efficiency in terms of time and space | |

<table>
<tr><td></td><td colspan="2">complexity, and select appropriate solutions.<br>**OC4**. Implement algorithmic solutions effectively using a programming language of their choice.<br>**OC5**. Construct and traverse binary trees and binary search trees, demonstrating a clear conceptual and practical grasp of tree structures.</td></tr>
<tr><td rowspan="3">**9**</td><td colspan="2">**Modules:-**<br>**Module 1:**</td></tr>
<tr><td>**Introduction and Fundamentals**<br>Basic Terminology: Concepts of data, information, data structures, and Abstract Data Types (ADT)<br>Classification of Data Structures: Linear and Non-linear structures<br>**Algorithm Analysis**: Introduction to algorithm efficiency, time complexity, and Big O notation<br>Arrays and Linked Lists<br>Arrays: Representation, traversal, insertion, deletion, and searching techniques<br>Linked Lists: Singly linked list – structure, creation, insertion, deletion, and traversal<br>**Comparative Analysis**: Arrays vs. Linked Lists – use cases, advantages, and limitations Stacks, Queues, and Recursion<br>Stacks: Stack ADT and operations: Push, Pop, Peek<br>Array-based implementation Applications: Expression evaluation (postfix/prefix)<br>Queues: Queue ADT and operations: Enqueue, Dequeue, Peek<br>Array-based implementation Applications of queues in real-life scenarios Recursion: Introduction to recursion with basic examples (e.g., factorial, Fibonacci series) Understanding recursive logic and its relation to stack behaviour</td><td>**15 Hrs**</td></tr>
<tr><td colspan="2">**Module 2:**</td></tr>
</table>

| | | |
|---|---|---|
| | **Trees,Binary Trees: Representation of binary trees, Tree traversal techniques: Inorder, Preorder, and Postorder**<br>**Binary Search Trees (BST)**: Operations: Insertion, Deletion, and Search, Applications of trees in computing (e.g., hierarchical data representation, searching)<br>**Hashing**: Introduction to Hashing and Hash Functions, Implementation of Hash Tables, Collision Resolution Techniques: Separate Chaining, Applications of Hashing<br><br>**Sorting and Searching Algorithms :- Sorting Techniques**: Bubble Sort, Insertion Sort, Selection Sort **Searching Techniques**:Linear Search, Binary Search | **15 Hrs** |
| **10** | **Books and References:**<br>1. Data Structures and Algorithms made Easy: Data Structures and Algorithmic Puzzles,<br>Narasimha Karumanchi ,5th Edition 2017<br>2. A Simplified Approach to Data Structures, Lalit Goyal, Vishal Goyal, Pawan Kumar SPD,1st 2014<br>3. Problem Solving in Data Structures & Algorithms Using C by Hemant Jain ,1st Edition, BPB Publications, 2018<br>4. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest,<br>Clifford Stein, 4th Edition, MIT Press,2022 | |
| **11** | **Internal Continuous Assessment: 40%** | **Semester End Examination: 60%** |
| **12** | **Continuous Evaluation through:**<br>Class test: 15 marks<br>Quizzes/ Presentations/ Assignments: 5 marks<br>Total: 20 marks | **Format of Question Paper: External Examination (30 Marks)– 1 hr duration** |
| **13** | **Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)**<br>Q1: Attempt any three (out of five) from Module 1 (15 marks)<br>Q2: Attempt any three (out of five) from Module 2 (15 marks) | |

**NIRMALA MEMORIAL FOUNDATION COLLEGE OF COMMERCE AND SCIENCE (AUTONOMOUS)**

## Name of the Course: Operating System

| Sr.No. | Heading | Particulars |
|---|---|---|
| 1 | **Description the course :**<br>**Including but Not limited to:** | Introduce operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file systems and protection)<br>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 Hours |
| 6 | **Marks Allotted:** | 50 Marks |
| 7 | **Course Objectives(CO):**<br><br>**CO1**. Understand the fundamental concepts of operating systems, including their types, structure, and core functionalities.<br>**CO2**.Comprehend process management techniques including process scheduling, threading, and synchronization mechanisms.<br>**CO3.** Analyse memory management strategies, including paging, segmentation, and virtual memory concepts.<br>**CO4.** Understand the principles of deadlock, including its characterization, prevention, and recovery techniques.<br>**CO5.** Explore file system interfaces, including file organization, directory structures, and access methods.<br>**CO6.** Recognize the need for protection and security in operating systems. | |
| 8 | **Course Outcomes (OC):**<br><br>**OC1.** Explain the structure and functions of an operating system and how it acts as an interface between hardware and users.<br>**OC2.** Describe and analyse the different states of a process and demonstrate knowledge of process scheduling and multithreading.<br>**OC3.** Apply synchronization techniques to solve critical-section problems and avoid race conditions using semaphores and other methods.<br>**OC4.** Illustrate various memory management techniques, including paging and | |

| | | |
|---|---|---|
| | segmentation, and analyse page replacement algorithms.<br>**OC5.** Identify and handle deadlocks using appropriate algorithms for detection, prevention, and recovery.<br>**OC6.** Describe file system architecture, access methods, and implement directory structures.<br>**OC7.** Demonstrate an understanding of system protection and security mechanisms to ensure OS stability and integrity. | |
| **9** | **Modules:-**<br>**Module 1:** | |
| | Operating System Overview, Basics of operating systems, Generations, Types and Functions, Structure, Services, System Calls, System Boot, System Programs, Protection and Security, Process Management: Process Concepts, Process Creation and Deletion, Process Control Block, Scheduling-Criteria, Scheduling Algorithms and their Evaluation, Threads, Threading Issues. Introduction to process Synchronization, Inter process Communication, Critical-Section Problem, Peterson's Algorithm, Synchronization Hardware, Semaphores, Classic Problems of Synchronization. | **15 Hrs** |
| | **Module 2:** | |
| | 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, Banker's Algorithm, File System Interface, File Concept, Access Methods, Directory Structure, and File System Structure. | **15 Hrs** |
| **10** | **Books and References:**<br>1. Operating Systems – Internals and Design Principles William Stallings, Pearson 9th , 2009<br>2. Operating System Concepts, Abraham Silberschatz, Wiley, 8th Edition<br>3. Operating Systems, Godbole and Kahate, Godbole and Kahate, 3rd Edition. | |
| **11** | **Internal Continuous Assessment: 40%** | **Semester End Examination:** |

| | | |
|---|---|---|
| | | **60%** |
| **12** | **Continuous Evaluation through:**<br>Class test: 15 marks<br>Quizzes/ Presentations/ Assignments: 5 marks<br>Total: 20 marks | **Format of Question Paper: External Examination (30 Marks)– 1 hr duration** |
| **13** | **Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)**<br>Q1: Attempt any three (out of five) from Module 1 (15 marks)<br>Q2: Attempt any three (out of five) from Module 2 (15 marks) | |

## Name of the Course: Python for AI + Data Structures for Problem Solving Practical

| Sr.No. | Heading | Particulars |
|---|---|---|
| 1 | **Description the course : Including but Not limited to:** | **Python for AI** This course offers a comprehensive exploration of advanced Python programming concepts, designed to equip students with the tools to tackle real-world problems efficiently. Key topics include regular expressions, Python's built-in date and time modules, calendar-based operations, and file handling techniques for both text and binary files. Students will also learn methods for extracting and processing structured data.<br>**Data Structures for Problem Solving**<br>In a separate module, the course covers essentials of Data Structures. This includes the study and application of structures such as arrays, linked lists, stacks, queues, trees, and hash tables. Emphasis is placed on understanding their underlying logic, practical implementation, and performance analysis. This dual-language approach enhances students' programming versatility and problem-solving skills. |
| 2 | **Vertical :** | Major |
| 3 | **Type :** | Practical |
| 4 | **Credits :** | 2 credits |
| 5 | **Hours Allotted :** | 60 Hours |
| 6 | **Marks Allotted:** | 50 Marks |
| 7 | **Course Objectives(CO):**<br><br>**CO1.** Develop a solid understanding of core programming concepts in Python, including input/output operations, conditional statements, and loop constructs.<br>**CO2.** Apply array operations, indexing, slicing, and mathematical functions using the NumPy library for efficient numerical computing.<br>**CO3.** Strengthen problem-solving abilities through the use of functions, recursion, lambda expressions, and modular programming in Python.<br>**CO4.** Utilize built-in Python data structures such as lists and dictionaries, along with performing file operations for both text and binary files.<br>**CO5.** Work with text processing, date and time manipulation, and calendar-based operations in Python.<br>**CO6.** Gain hands-on experience in implementing fundamental data structures—such as arrays, linked lists, stacks, and queues—using C++.<br>**CO7.** Develop skills in designing and analyzing algorithms to solve computational | |

| | | |
|---|---|---|
| | problems using appropriate data structures.<br>**CO8.** Learn to select and justify the use of suitable data structures for various real-world applications.<br>**CO9.** Enhance understanding of dynamic memory allocation and efficient data management techniques in C++.<br>**CO10.** Build the ability to debug, test, and optimize code related to data structure operations for improved performance and reliability. | |
| **8** | **Course Outcomes (OC):**<br><br>**OC1.** Apply Python programming concepts such as input/output operations, conditional statements, and loops to solve fundamental problems.<br>**OC2.** Demonstrate proficiency in performing array operations, indexing, slicing, and analyzing attributes of arrays using NumPy.<br>**OC3.** Solve computational problems using functions, recursion, and lambda expressions, and implement modular programming for efficient and reusable code.<br>**OC4.** Implement real-world solutions using Python data structures like lists and dictionaries, along with file operations.<br>**OC5.** Process and analyze text data, extract relevant information, handle date and time operations, and measure program execution time.<br>**OC6.** Gain hands-on experience in implementing fundamental data structures including arrays, linked lists, stacks, queues, trees, and graphs using C++.<br>**OC7.** Design and analyze algorithms to solve computational problems using appropriate data structures.<br>**OC8.** Choose suitable data structures for different applications and justify their use based on problem requirements.<br>**OC9.** Understand and apply dynamic memory allocation and effective data management techniques in C++.<br>**OC10.** Debug and optimize data structure implementations for correctness and performance. | |
| **9** | **Modules:-**<br>**Module 1:** | |
| | **Practical 1:-**Write programs for the following:<br>a. Write a program that asks the user for their name and the age they will turn this year. Then, calculate and print the year they were born, addressing them by name.<br>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.<br>c. Write a program that accepts a signal strength value (in dBm) from the user and prints the corresponding signal quality based on the following scale:<br><br>**Signal Strength (dBm)** **Signal Quality**<br>-50 to 0 Excellent<br>-70 to -51 Good | **30 Hrs** |

| | | |
|---|---|---|
| | -85 to -71 Fair<br>-100 to -86 Poor<br>Below -100 No Signal<br><br>**Practical 2:-**Write programs for the following:<br>a. Write a program to print factorial of the number.<br>b. Write a program to generate the Fibonacci series.<br>c. Write a program to accept a number from the user display sum of its digits.<br>**Practical 3:-**Write programs for the following:<br>a. Write a program that accepts a list of daily temperatures (in °C) for a week from the user. Perform the following:<br>i. Print the full list of temperatures.<br>ii. Print the temperature of the first and last day.<br>iii. Print temperatures from day 3 to day 5.<br>iv. Calculate and print the average temperature for the week.<br>b. Write a program to implement mathematical functions on arrays.<br>c. Write a program to perform array aliasing and copying.<br>**Practical 4:-**Write programs for the following:<br>a. You are analyzing the daily temperatures (in °C) of a city for a week. Write a Python program using NumPy to do the following:<br>i. Store the temperatures in a NumPy array: [30, 32, 31, 35, 36, 34, 33].<br>ii. Print the temperature on the 4th day (using basic indexing).<br>iii. Print the temperatures from day 2 to day 5 (using slicing).<br>iv. Display the temperatures on alternate days (every second element).<br>v. Display all temperatures above 33°C (using boolean indexing).<br>b. Write a program to analyze dimensions and attributes of arrays<br>**Practical 5:-**Write programs for the following:<br>a. Write a function to check the input value is Armstrong and also write the function for Palindrome.<br>b. Write a recursive function to print the factorial for a given number.<br>c. Write a lambda function that checks whether a given string starts with a specific character.<br>**Practical 6:-**Write programs for the following:<br>a. Write a program to compute number of characters and words in a string.<br>b. Write a program to create a custom Python module containing | |

| | | |
|---|---|---|
| | at least two functions. Then write a separate script to import and test these functions.<br>**Practical 7:-**Write programs for the following:<br>a. Write a program that takes two lists and returns True if they have at least one common member.<br>b. Write a Python script to sort (ascending and descending) a dictionary by value.<br>**Practical 8:-**Write programs for the following:<br>a. Write a program to accept and pass radius to a function that returns area and circumference (using tuple).<br>b. Write a program to perform basic file operations on text files and binary files.<br>c. Write a Python program to read last n lines of a file.<br>**Practical 9:-**Write programs for the following:<br>a. Write a program to count the occurrences of a specific word in a file using regular expressions.<br>b. Write a program to extract all hyperlinks (<a href="…">) from HTML File.<br>**Practical 10:-**Write programs for the following:<br>a. Write a program that compares two dates (in DD/MM/YYYY format) and prints which one is earlier.<br>b. Write a program to measure program execution time.<br>c. Write a program using the calendar module to print the weekday of the first day of a given month and year. | |
| | **Module 2:** | |
| | **Practical 1:-**Write a program that performs basic operations on an array, including:<br>a. Adding an element at a given index within the array.<br>b. Removing an element from a specified index in the array.<br>**Practical 2:-**Implement a program to manage a singly linked list with the following functionalities:<br>a. Initialize an empty singly linked list.<br>b. Add a new node at the start, at the end, and at a specified position within the list.<br>c. Delete a node from a given position in a linked list.<br>**Practical 3:-**Develop a program that performs the following tasks:<br>a. Implement a stack data structure using an array.<br>b. Evaluate a postfix expression using the stack implementation.<br>**Practical 4:-**Write a program to:<br>a. Implement a queue data structure using an array. | **30 Hrs** |

| | | |
|---|---|---|
| | b. Simulate a basic queue system, such as managing customers waiting in line.<br>**Practical 5:-**Design a program to manage a Binary Search Tree (BST) with the following capabilities:<br>a. Construct an empty Binary Search Tree.<br>b. Insert new nodes into the BST following BST properties.<br>c. Search for a specific node within the BST.<br>**Practical 6:-**Write a program to perform different types of tree traversals on a binary tree, including:<br>a. Inorder traversal<br>b. Preorder traversal<br>c. Postorder traversal<br>**Practical 7:-**Develop a program to<br>a. Store and retrieve data from the hash table.<br>b. Write a program to implement the collision technique<br>**Practical 8:-**Write programs to implement and compare the following sorting algorithms:<br>a. Bubble sort<br>b. Insertion sort<br>c. Selection sort<br>**Practical 9:-**Write programs to implement and compare: a. Linear search b. Binary search (on a sorted array)<br>**Practical 10:-**Design a simple program that integrates and manipulates more than one data structure. | |
| **10** | **Text Books:**<br>1. Learning Python, Fourth Edition by Mark Lutz Copyright © 2009 Mark Lutz. Published by O'Reilly Media, Inc.<br>2. Python Basics: A Practical Introduction to Python<br>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<br>**Reference Books:**<br>1. Let Us Python, Yashwant. B. Kanetkar, BPB Publication, 2019<br>2. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018<br>3. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017<br>4. A Simplified Approach to Data Structures, Lalit Goyal, Vishal Goyal, Pawan Kumar SPD,1st 2014<br>5. Problem Solving in Data Structures & Algorithms Using C by Hemant Jain ,1st Edition, BPB Publications, 2018<br>6. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 4th Edition, MIT Press,2022 | |
| **11** | **Internal Continuous Assessment: 40%** | **Semester End Examination: 60%** |

| | | |
|---|---|---|
| **12** | **Continuous Evaluation through:**<br>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 totalling to 50 marks and can be converted to 20 marks. | **30 marks practical exam of 2 hours duration** |
| **13** | **Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination Practical Slip:**<br>Q1. From Module 1 13 marks<br>Q2. From Module 2 12marks<br>Q3. Journal and Viva 05 marks | |

# Vertical – 4

## Vocational Skill Course

# (VSC)

**NIRMALA MEMORIAL FOUNDATION COLLEGE OF COMMERCE AND SCIENCE (AUTONOMOUS)**

## Name of the Course: APPLIED MATHEMATICS

| Sr.No. | Heading | Particulars |
|---|---|---|
| **1** | **Description the course :**<br>**Including but Not limited to:** | This course is designed for developing competency of the students in the applications of various mathematical concepts. It is equipped with Complex numbers, Laplace transform, Inverse Laplace transform, Differential equations of first order with first degree and higher degree. This course introduces basic concepts of Algebra and prepares students to study further courses in linear and abstract algebra. |
| **2** | **Vertical :** | Vocational Skill Course |
| **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):**<br><br>**CO1**: Ability to interpret the mathematical results in physical or practical terms for complex numbers.<br>**CO2**: Know and to understand various types of methods to solve Laplace transform.<br>**CO3**: Apply the knowledge of Laplace Transforms to solve the problems.<br>**CO4**: Know and to understand various types of methods to solve differential equations.<br>**CO5**: Apply the knowledge of differential equations to solve the problems.<br>**CO6**: Inculcate the habit of Mathematical Thinking through Indeterminate forms. | |
| **8** | **Course Outcomes (OC):**<br><br>**OC1.** Familiar with the various forms and operations of a complex number.<br>**OC2.** Find the Laplace transform of a function using definition.<br>**OC3**. Find the Inverse Laplace transform of a function using definition.<br>**OC4**. Solve Differential equations of first degree and first order.<br>**OC5**. Solve Differential equations of first degree and higher order. | |

| | | |
|---|---|---|
| **9** | **Modules:-**<br>**Module 1:** | |
| | **1.1 Complex Numbers:** Complex number, Equality of complex numbers, Graphical representation of complex number (Argand's Diagram), Polar form of complex numbers. Polar form of x+iy for different signs of x.y, Exponential form of complex numbers, Mathematical operation with complex numbers and their representation on Argand's Diagram, Circular functions of complex angles, Definition of hyperbolic function. Relations between circular and hyperbolic functions, Inverse hyperbolic functions.<br>**1.2 The Laplace Transform:** Introduction. Definition of the Laplace Transform, Table of Elementary Laplace Transforms. Theorems on Important Properties of Laplace Transformation, First Shifting Theorem, Second Shifting Theorem, Convolution Theorem, Laplace Transform of Derivatives.<br>**1.3 Inverse Laplace Transform:** Shifting Theorem, Partial fractions Methods, Use of Convolution Theorem, Solution of Ordinary Linear Differential Equations with Constant Coefficients, Laplace Transformation of Special Function,<br>Periodic Functions, Heaviside Unit Step Function, Dirac-delta Function (Unit Impulse Function). | **15 Hrs** |
| | **Module 2:** | |
| | **2.1 Equation of the first order and of the first degree:** Separation of variables, Equations homogeneous in x and y, Non-homogeneous linear equations, Exact differential Equation, Integrating Factor, Linear Equation and equation reducible to this form, Method of substitution.<br>**2.2 Differential equation of the first order of a degree higher than the first**: Introduction, Solvable for p (or the method of factors), Solve for y, Solve for x, Clairaut's form of the equation, Method of Substitution.<br>**2.3 Linear Differential Equations with Constant Coefficients:** Introduction, The Differential Operator, Linear Differential Equation f(D) y = 0, Different cases depending on the nature of the root of the equation f(D) = 0, Linear differential equation f(D) y = X, The complimentary Function, The inverse operator 1/f(D) and the symbolic expression for the particular integral | **15 Hrs** |

| | | |
|---|---|---|
| **10** | **Books and References:**<br>1.A text book of Applied Mathematies Vol I, P. N. Wartikar and J. N.Wartikar, Pune Vidyathi Griha,7* ,1995<br>2.A text book of Applied Mathematies Vol II , P. N. Wartikar and J. N. Wartikar, Pune Vidyathi Griha,7" .1995<br>3.Higher Engineering Mathematies, Dr. B. S.Grewal, Khanna Publications. | |
| **11** | **Internal Continuous Assessment: 40%** | **Semester End Examination: 60%** |
| **12** | **Continuous Evaluation through:**<br>Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks<br>Quizzes/ Presentations/ Assignments: 5 marks<br>Total: 20 marks | **Format of Question Paper: External Examination (30 Marks)– 1 hr duration** |
| **13** | **Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)**<br>Q1: Attempt any three (out of five) from Module 1 (15 marks)<br>Q2: Attempt any three (out of five) from Module 2 (15 marks) | |

NIRMALA MEMORIAL FOUNDATION COLLEGE OF COMMERCE AND SCIENCE (AUTONOMOUS)

# B.Sc. (Information Technology)

# Semester - IV

# Vertical – 1
# Major

**NIRMALA MEMORIAL FOUNDATION COLLEGE OF COMMERCE AND SCIENCE (AUTONOMOUS)**

## Name of the Course: Core Java Techniques

| Sr.No. | Heading | Particulars |
|---|---|---|
| **1** | **Description the course :**<br>**Including but Not limited to:** | Core Java course focuses on teaching students how to design, develop, and maintain software applications using the Java programming language. The course covers fundamental to advanced concepts of Java, enabling students to understand object- oriented programming (OOP) principles, data structures, algorithms, and real-world application 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):**<br>**CO 1:** Understand and Apply Object-Oriented Programming (OOP) Concepts.<br>**CO 2:** Identify the key components of a class and object in Java, including attributes (fields), methods, and constructors.<br>**CO 3:** Apply sound software engineering principles in Java by organizing code into classes and methods with proper access control identifiers<br>**CO 4:** Use tools and techniques like unit testing, as well as IDE debugging tools to find and fix issues within Java programs.<br>**CO 5.** Effectively use Java's collection framework (e.g., Lists, Sets, Maps) to manage and process groups of related objects.<br>**CO 6.** Use OOP concepts in designing and building solutions to real-world problems, ensuring the application is modular, maintainable, and reusable. | |

| | | |
|---|---|---|
| **8** | **Course Outcomes (OC):**<br>OC1. Understand the basics of Java and its runtime environment.<br>OC2. Be proficient in using Java's data types, control flow statements, and OOP principles such as classes, inheritance, and exception handling.<br>OC3. Creating own classes and objects<br>OC4. Develop mini projects using Class, Interface and exception handling | |
| **9** | **Modules:-**<br>**Module 1:** | |
| | **Introduction to Java Programming** : History of Java and its Evolution, Features of Java (Platform Independence, Object Oriented), Data Types and Variables, Operators Constants and Literals, Type Casting<br>**Decision Making and Loops :** If-else Statements, Switch Statement, Loops (For, While, Do-While), Break and Continue Statements<br>**Classes and Objects :** Array, Arrays String class and String methods, String Buffer and StringBuilder, Object-Oriented Programming Concepts, Defining Classes and Creating Objects, Instance Variables and Methods, Constructors, this Keyword, super keyword, Types of Classes, Scope Rules, Access Modifier, constants, static members of a class, garbage collection.<br><br>**Inheritance:** Its types, Superclass and Subclass, Final classes and methods **Polymorphism:** Compile-time and Runtime Polymorphism **Interfaces:** Defining and Implementing Interfaces, Abstract Classes and Methods, Multiple Interface Implementation | 15 Hrs |
| | **Module 2:** | |
| | **Packages:** Introduction to predefined packages, User Defined Packages, Access specifier, Java Built-in packages<br>**Exception handling :** Try, Catch, and Finally Blocks, Throw and Throws Keywords<br>**Introduction to Threads:** Creating and Running Threads, Thread Lifecycle<br>**Introduction to JFC and Swing :** Features of the Java Foundation Classes, Swing API Components, JComponent Class, Containers and Panels, Labels, Buttons, RadioButton, Check Boxes, Text-Entry Components, Menus<br>**Layouts:** Flow Layout, Grid Layout, Border Layout<br>**Event Handling:** Delegation Event Model, Events, Event classes, Event listener interfaces, Using delegation event model, adapter classes | 15 Hrs |

| | | |
|---|---|---|
| **10** | **Books and References:**<br>1. Java: The Complete Reference Herbert Schildt MC-Graw HILL 12th EDITION 2022<br>2. Core Java, Volume I: Fundamentals Hortsman Pearson 9th 2013<br>3. Core Java, Volume II: Advanced FeaturesGary Cornell and Hortsman Pearson 8th 2008<br>4. Core Java: An Integrated Approach R. Nageswara Rao DreamTech 1st 2008 | |
| **12** | **Internal Continuous Assessment: 40%** | **Semester End Examination: 60%** |
| **13** | **Continuous Evaluation through:**<br>Class test: 15 marks<br>Quizzes/ Presentations/ Assignments: 5 marks<br>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)**<br>Q1: Attempt any three (out of five) from Module 1 (15 marks)<br>Q2: Attempt any three (out of five) from Module 2 (15 marks) | |

## Name of the Course: Software Engineering

| Sr.No. | Heading | Particulars |
|---|---|---|
| 1 | **Description the course :**<br>**Including but Not limited to:** | This course introduces core concepts of software engineering and the Software Development Life Cycle (SDLC), including requirements analysis and documentation. It covers key development models—Waterfall, Prototyping, Iterative, and Agile—with a focus on Scrum. Topics also include verification, validation, software testing, test design, and COTS product reuse. |
| 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):**<br><br>**CO1.** Explain the principles of software engineering and the stages of the Software Development Life Cycle (SDLC).<br>**CO2.** Identify, classify, and document functional and non-functional software requirements.<br>**CO3.**. Compare and apply various software development models, including Waterfall, Prototyping, Iterative, and Agile methodologies.<br>**CO4.** Demonstrate understanding of Agile principles, Scrum framework, and sprint management practices.<br>**CO5.** Plan and implement verification and validation strategies during software development.<br>**CO6.** Apply different software testing techniques, including test case design, system testing, and COTS product integration. | |
| 8 | **Course Outcomes (OC):**<br><br>**OC1.** Describe the fundamentals of software engineering and explain each phase of the Software Development Life Cycle (SDLC).<br>**OC2**. Analyze and document software requirements, distinguishing between functional and non-functional requirements.<br>**OC3**. Evaluate and select appropriate software development process models for given project scenarios.<br>**OC4**. Apply Agile methodologies, including Scrum practices such as sprint planning, roles, workflow, and retrospectives, in software projects.<br>**OC5**. Develop and implement verification and validation plans to ensure software | |

<table>
<tr><td></td><td colspan="2">quality and compliance.<br>**OC6**. Design effective test cases and apply suitable testing strategies and levels, including system testing and COTS reuse.</td></tr>
<tr><td rowspan="4">**9**</td><td colspan="2">**Modules:-**<br>**Module 1:**</td></tr>
<tr><td>Software and Software Engineering, Software Development Life Cycle, Software Requirements: Functional and Non-functional requirements, Documentation of the software requirements, Software Development Process Models: • Waterfall Model. • Prototyping. • Iterative Development, Agile Software Development, Agile Software Development Process, Principles of Agile Software Development, Scrum in Software Development, Sprint Planning, Sprint Roles, Sprint workflow and process, Sprint Retrospectives</td><td>**15 Hrs**</td></tr>
<tr><td colspan="2">**Module 2:**</td></tr>
<tr><td>Verification and Validation: Planning Verification and Validation,Software testing: Types of Software Testing ,Types of System Testing Test Case Design , Levels of testing, COTS product reuse**.**</td><td>**15 Hrs**</td></tr>
<tr><td>**10**</td><td colspan="2">**Books and References:**<br>1. "Software in 30 Days" by Ken Schwaber and Jeff Sutherland, Wiley, 1st edition (2012).<br>2. "Scrum Insights for Practitioners: The Scrum Guide Companion" by Hiren Doshi, PracticeAgile Solutions, 1st edition (2016).<br>3. "A Scrum Book: The Spirit of the Game" by Jeff Sutherland and James O. Coplien, Pragmatic Bookshelf, 1st edition (2019).</td></tr>
<tr><td>**11**</td><td>**Internal Continuous Assessment: 40%**</td><td>**Semester End Examination: 60%**</td></tr>
<tr><td>**12**</td><td>**Continuous Evaluation through:**<br>Class test: 15 marks<br>Quizzes/ Presentations/<br>Assignments: 5 marks<br>Total: 20 marks</td><td>**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**</td></tr>
<tr><td>**13**</td><td colspan="2">**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)**<br>Q1: Attempt any three (out of five) from Module 1 (15 marks)<br>Q2: Attempt any three (out of five) from Module 2 (15 marks)</td></tr>
</table>

**NIRMALA MEMORIAL FOUNDATION COLLEGE OF COMMERCE AND SCIENCE (AUTONOMOUS)**

## Name of the Course: Computer Networks

| Sr.No. | Heading | Particulars |
|---|---|---|
| 1 | **Description the course :**<br>**Including but Not limited to:** | This course provides students knowledge and skills to understand and implement the Networking skills. It will help them to implement Virtual networks to understand and resolve real-world network problems. |
| 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):**<br>**CO1**. To understand the basic concepts of networks<br>**CO2**. Understand basic concepts in OSI Model , distinguishing Factors in TCP/IP ,IP addressing Schemes<br>**CO3**. Understand How the communication happens across the network<br>**CO4**. Understanding of various Routing protocol and their implementation | |
| 8 | **Course Outcomes (OC):**<br>**OC1** .Understanding the Transport layer protocols and their utilities<br>**OC2** .Various application layer protocols and their implementation<br>**OC3 .**Mailing Services and web services implementation | |
| 9 | **Modules:-**<br>**Module 1:** | |
| | **Introduction:** Network Definition , components of data communication , Data flow , Characteristics of data communication , Analog and digital Signals and their characteristics , Conversions from one form to another form. Layered architecture , OSI reference Model, TCP/IP Protocol Suite, IPV 4 Addressing classfull , classless , Subnet mask and Protocol and IPV6 Addresses and Protocol.<br>Address Resolution Protocol (ARP), Internet Control. Message Protocol Version 4 (ICMPv4), Mobile IP, Unicast Routing Protocols (RIP, OSPF and BGP) | **15 Hrs** |

| | | |
|---|---|---|
| | **Module 2:** | |
| | **Flow control protocols :** Stop and wait protocols , sliding window protocols (selective repeat ARQ , go back n ARQ)<br>User Datagram Protocol (UDP), Transmission Control Protocol (TCP)<br>**Host Configuration:** DHCP, Domain Name System (DNS)<br>**Remote Login**: TELNET and SSH, File Transfer: FTP and TFTP ; World Wide Web and HTTP,<br>**Electronic Mail:** SMTP, POP, IMAP and MIME. | **15 Hrs** |
| **10** | **Books and References:**<br>1. Data communication and Networking : Behrouz Forouzan<br>2. Computer Network : Tanenbaum<br>3. Computer Network a Top down approach: James F.Kurose , Keith W. Ross | |
| **11** | **Internal Continuous Assessment: 40%** | **Semester End Examination: 60%** |
| **12** | **Continuous Evaluation through:**<br>Class test : 15 marks<br>Quizzes/ Presentations/ Assignments: 5 marks<br>Total: 20 marks | **Format of Question Paper: External Examination (30 Marks)– 1 hr duration** |
| **13** | **Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)**<br>Q1: Attempt any three (out of five) from Module 1 (15 marks)<br>Q2: Attempt any three (out of five) from Module 2 (15 marks) | |

## Name of the Course:

## Core Java Techniques + Computer Networks Practical

| Sr.No. | Heading | Particulars |
|---|---|---|
| 1 | **Description the course : Including but Not limited to:** | **Core Java Techniques Practical**<br>This course is a stepping stone to learn other languages. This course provides students hands-on experiences of coding exercises and projects.<br>**Database Management System's practical** approach is useful to gain the knowledge for software backend development. It benefits to user by providing data definition, data access, reduced data redundancy, data integrity, data sharing, data organizing, data consistency, data accuracy, and security |
| 2 | **Vertical :** | Major |
| 3 | **Type :** | Practical |
| 4 | **Credits :** | 2 credits |
| 5 | **Hours Allotted :** | 60 Hours |
| 6 | **Marks Allotted:** | 50 Marks |
| 7 | **Course Objectives(CO):**<br><br>**CO1.** Understand core Java programming concepts, including data types, control structures, and object-oriented programming principles.<br>**CO2** Develop the ability to implement inheritance, polymorphism, interfaces, and abstract classes effectively.<br>**CO3.** Gain hands-on experience with exception handling, multithreading, and dynamic initialization.<br>**CO4.** Learn to apply Java programming to solve real-world problems, such as matrix operations and finding areas/volumes.<br>**CO5.** Enhance debugging and problem-solving skills using Java's rich standard libraries and features.<br>**CO6.** Basic foundation of LAN<br>**CO7.** various command line utilities to be tested<br>**CO8.** Practical implementation of IP Subnetting<br>**CO9.** Testing of various Routing Protocols | |
| 8 | **Course Outcomes (OC):**<br><br>**OC1**. Write efficient Java programs to perform arithmetic operations, manage control flow, and manipulate strings.<br>**OC2**. Demonstrate knowledge of object-oriented concepts by implementing inheritance, polymorphism, and interfaces.<br>**OC3**. Apply exception handling mechanisms to create robust Java applications. | |

| | | |
|---|---|---|
| | **OC4**.1Implement multithreading and explore dynamic initialization for advanced Java programming.<br>**OC5**. Solve computational problems, such as matrix operations and factorial calculation, using packages and Java constructs.<br>**OC6**. Implementation of utility protocols<br>**OC7**. Understanding Basic Security features<br>**OC8**. Network Traffic and Packet Analysis<br>**OC9**. Basic Understanding of Wireless Network | |
| **9** | **Modules:-**<br>**Module 1:** | |
| | **Practical 1:**<br>a. Write a program in Java to demonstrate Boolean value by checking whether the age is appropriate to vote. The age must **greater than or equal to** 18. Otherwise output "Not old enough to vote."<br>b. Write a program in Java to print a string 10 times using a for loop.<br>**Practical 2:**<br>a. Write a program in Java to check the grade of marks using a switch case.<br>b. Write a program in Java to find the biggest element among three numbers using if else.<br>**Practical 3:**<br>a. Write a program in Java to create a class and access all data members and methods using the object and compute the area and perimeter of a circle.<br>b. Write a program in Java to access member variables using the constructor.<br>**Practical 4:**<br>a. Write a program in Java to multiply two matrices.<br>b. Write a program in Java to calculate the area of a rectangle using single inheritance.<br>c. Write a program in Java to demonstrate multilevel inheritance.<br>**Practical 5:**<br>a. Write a program in Java to demonstrate hierarchical inheritance.<br>b. Write a program in Java to find the area and perimeter of a circle using an abstract class.<br>**Practical 6:**<br>a. Write a program in Java to find the average of three numbers using the method overloading.<br>b. Write a program in Java to find average of three numbers using method overriding.<br>**Practical 7:**<br>a. Write a program in Java to create an interface area. | **30 Hrs** |

| | | |
|---|---|---|
| | Find the area of a circle.<br>b. Write a program in Java to find the sum and average of three numbers using the super keyword.<br>**Practical 8:**<br>a. Write a program in Java to find the volume of a box using constructor overloading.<br>b. Write a program in Java to demonstrate exception handling in case of variable/constant divided by zero.<br>**Practical 9:**<br>a. Write a program in Java to implement multiple inheritance using the interface.<br>b. Write a program in Java to implement thread.<br>**Practical 10:**<br>a. Write a program in Java to find the factorial of a number using the package.<br>b. Write a program in Java to import the package.<br>c. Write a program to implement Flow, Grid and Border Layout using swing.<br>d. Write program to demonstrate following events Action Mouse Key | |
| | **Module 2:** | |
| | **Practical 1:-** Configuring LAN setup<br>a. Planning and Setting IP networks<br>b. Configuring subnet<br>c. Using, linux-terminal or Windows-cmd, execute following networking<br>commands and note the output: ping, traceroute, netstat, arp, ipconfig, Getmac, hostname, NSLookUp, pathping, SystemInfo<br>**Practical 2:-**IPv4 Addressing and Subnetting<br>a. Given an IP address and network mask, determine other information about the IP address such as:<br>a. Network address , Network broadcast address , Total<br>b. number of host bits , Number of hosts.<br>b. Given an IP address and network mask, determine other information about the IP address such as:<br>c. The subnet address of this subnet<br>d. The broadcast address of this subnet<br>e. The range of host addresses for this subnet<br>f. The maximum number of subnets for this subnet mask<br>g. The number of hosts for each subnet<br>h. The number of subnet bits ,The number of this subnet<br>**Practical 3:-**Configure Static IP routing using.<br>**Practical 4:-**Configure IP routing using RIP.<br>**Practical 5:-**Configuring Simple and multi-area OSPF | **30 Hrs** |

| | | |
|---|---|---|
| | **Practical 6:-**Configuring BGP protocol (Multi-Autonomous)<br>**Practical 7:-**Configuring server and client.<br>a. Configure DHCP<br>b. Configure DNS<br>c. Configure HTTP<br>d. Configure Telnet<br>e. Configure FTP<br>**Practical 8:-**Configure basic security features for networks<br>**Practical 9:-**Using Wireshark, network analyzer, set the filter for ICMP, TCP, HTTP, UDP,<br>FTP and perform respective protocol transactions to show/prove that the network analyzer is working<br>**Practical 10:-**create a wireless network of multiple PCs using appropriate access point.<br>**Practical 11:-**IPV6 Addressing Basics | |
| **10** | **Books and References:**<br>1. Core Java 8 for Beginners Vaishali Shah, Sharnam Shah SPD 1st 2015<br>2. Java: The Complete Reference Herbert Schildt McGraw Hill 9th 2014<br>3. Murach's beginning Java with Net Beans Joel Murach , Michael Urban SPD 1st 2016<br>4. Core Java, Volume I: Fundamentals Hortsman Pearson 9th 2013<br>5. Core Java, Volume II: Advanced Features Gary Cornell and Hortsman Pearson 8th 2008<br>6. Core Java: An Integrated Approach R. Nageswara Rao DreamTech 1st 2008<br>7. TCP/IP Protocol Suite Behrouz A. Forouzan Tata McGraw Hill 2010<br>8. Data Communication and Networking Behrouz A. Forouzan Tata McGraw Hill<br>9. Computer Networks Andrew Tanenbaum Pearson Fifth 2013 | |
| **11** | **Internal Continuous Assessment: 40%** | **Semester End Examination: 60%** |
| **12** | **Continuous Evaluation through:**<br>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 totalling to 50 marks and can be converted to 20 marks. | **30 marks practical exam of 2 hours duration** |

| | |
|---|---|
| **13** | **Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination Practical Slip:**<br>Q1. From Module 1 13 marks<br>Q2. From Module 2 12marks<br>Q3. Journal and Viva 05 marks |

# Vertical – 4

## Skill Enhancement Course

# (SEC)

## Name of the Course: Arduino Programming

| Sr.No. | Heading | Particulars | |
|---|---|---|---|
| 1 | **Description the course : Including but No** | This course covers the basics of programming and building electronic projects using the Arduino microcontroller. Students will learn to design circuits on breadboards and write code to interact with various sensors and actuators. Practical sessions include working with LEDs, light, temperature, humidity, gas, motion sensors, servo motors, and input devices like joysticks. The course emphasizes hands-on learning to help students develop skills in embedded systems and sensor integration for real-world applications. | |
| 2 | **Vertical :** | Major | |
| 3 | **Type :** | Practical | |
| 4 | **Credits :** | 2 credits | |
| 5 | **Hours Allotted :** | 30 Hours | |
| 6 | **Marks Allotted:** | 50 Marks | |
| 7 | **Course Objectives(CO):**<br>**CO1:** To understand the basics of Arduino hardware, circuits, and programming.<br>**CO2:** To learn to use breadboards for building electronic circuits.<br>**CO3:** To write Arduino programs that interface with different sensors and actuators.<br>**CO4:** To collect and process data from sensors like light, temperature, humidity, and gas sensors.<br>**CO5:** To control motors and other output devices using Arduino.<br>**CO6:** To develop problem-solving skills by creating embedded systems and automation projects. | | |
| 8 | **Course Outcomes (OC):**<br>**OC1**: Build and program Arduino-based circuits using breadboards.<br>**OC2:** Interface and use various sensors including light, temperature, humidity, and gas sensors.<br>**OC3:** Control output devices like LEDs, servo motors, and joysticks through Arduino code.<br>**OC4:** Collect, process, and respond to sensor data in embedded systems.<br>**OC5:** Troubleshoot and debug Arduino circuits and programs.<br>**OC6:** Design and implement basic automation and sensor-driven projects. | | |
| 9 | **Practical 1:-** Introduction to Arduino circuits and bread boarding Blinking of LEDs<br>**Practical 2:-** Program using Light Sensitive Sensors | | **30 Hrs** |

| | | |
|---|---|---|
| | **Practical 3:-** Program using temperature sensors<br>**Practical 4:-** Programs using humidity sensors<br>**Practical 5:-** Programs using Line tracking sensors<br>**Practical 6:-** Programs using Ultrasonic Sensors<br>**Practical 7:-** Programs using digital infrared motion sensors<br>**Practical 8:-** Programs using gas sensors<br>**Practical 9:-** Programs using servo motors<br>**Practical 10:-** Programs making Joystick with Arduino | |
| **10** | **Android Programming**<br><br>**Textbooks:**<br><br>1. *Getting Started with Arduino* by Massimo Banzi and Michael Shiloh<br>2. *Exploring Arduino* by Jeremy Blum<br><br>**Reference Books:**<br><br>1. *Arduino Cookbook* by Michael Margolis<br>2. *Programming Arduino: Getting Started with Sketches* by Simon Monk<br>3. *Practical Electronics for Inventors* by Paul Scherz and Simon Monk | |
| **11** | **Internal Continuous Assessment: 40%** | **Semester End Examination: 60%** |
| **12** | **Continuous Evaluation through:**<br>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 totalling to 50 marks and can be converted to 20 marks. | **30 marks practical exam of 2 hours duration** |
| **13** | **Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination Practical Slip:**<br>**Q1. From Module 1 13 marks**<br>**Q2. From Module 2 12marks**<br>**Q3. Journal and Viva 05 marks** | |