1. FOUNDATION COURSES OFFERED BY THE DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING, NSUT.

COURSE 1: COMPUTER PROGRAMMING

A. OVERVIEW OF THE COURSE:

Name of Course	Computer Programming
Offering Department	COMPUTER SCIENCE & ENGINEERING
B.Tech Branches to which this course is offered	All Branches

OVERVIEW:

The Course on **Computer Programming** is offered to the first-year students The objective of this course is to provide a comprehensive introduction to the world of programming. Through this course, students will learn the fundamental principles and techniques of coding using the Python language. The emphasis is on nurturing problem-solving abilities and cultivating a logical mindset essential for effective programming. Students will gain hands-on experience in writing and executing Python code, enabling them to develop simple yet functional programs. The course aims to familiarize students with essential data structures and control flow, empowering them to tackle real-world programming challenges with confidence. By the end of the course, students will have the skills to design and implement basic software solutions. This strong foundation in computer programming will serve as a steppingstone for their future studies and careers in various disciplines, from computer science to data science, web development, and beyond.

B. SYLLABUS

COURSE NO	TITLE OF THE COURSE	COURSE STRUCTURE	PRE-REQUISITE

FCCSE002	Computer Programming	3L - 0T - 2P	None

COURSE OUTCOMES (COs)

After completion of this course, the students are expected to be able to demonstrate the following knowledge, skills, and attitudes:

1. To identify and understand the syntax and features of Python.
2. To utilize control structures and practice the Python concepts like classes, objects & functions to solve programming problems.
3. To use arrays, strings and other data structures and to compare and contrast among different python libraries.
4. To implement file handling operations and perform input/output operations.
5. To design object-oriented Python programs and apply Python programming concepts to real-world scenarios.

COURSE CONTENTS

Unit I

Introduction to Python Programming: Introduction to Python and Environment Setup, Python Syntax and Basic Data Types, Variables, Operators, and Expressions, Control Structures and Conditional Statements, Loops and Iterations: For, While, For-else, Range function Sentinel-Controlled Iteration , break and continue Statements, Defining Functions, Functions with Multiple Parameters, Default Parameter Values, Keyword Arguments, Arbitrary Argument Lists, Lists and Tuples, Associated in-built functions, Dictionaries and Sets

Unit II

Object-Oriented Programming in Python: Object-Oriented Programming (OOP) Fundamentals, Classes and Objects, Inheritance and Polymorphism, Advanced OOP Concepts (e.g., Encapsulation, Abstraction), Generators and Iterators, Functional Programming Techniques (e.g., Lambda Functions, Map, Filter), Modules and Packages

Unit III

Arrays and Strings: Introduction to NumPy Arrays, Array Operations and Functions, Indexing and slicing arrays, Array Manipulation and Stacking, multidimensional arrays and matrices, Creating and manipulating strings, String Operations, Regular Expressions, pattern matching and substitution, Working with Dates and Times, Introduction to pandas Series and DataFrames, Data Analysis with Pandas (e.g., Filtering, Grouping, Aggregation)

Unit IV

File and Exception handling: File handling in Python, Different types of files and their uses, Updating text files: appending and modifying data, Understanding serialization and deserialization, Working with JSON files for data storage and retrieval, Handling Exceptions using try and except statements, Catching and handling specific exceptions, Raising custom exceptions, ‘finally’ clause for cleanup operations, Explicitly raising exceptions with ‘raise’, Working with CSV files

Unit V

Web Development and data analysis using python: Introduction to Web Development with Python, Introduction to Flask Framework, defining routes and handling HTTP methods (GET, POST), Request and response objects in Flask, Building Web Applications with Flask, Introduction to Data Visualization, Data Visualization with Matplotlib

SUGGESTED READINGS

1. R. Nageswara Rao, “**Core Python Programming**” Dreamtech Press, 3rd Edition, 2021.
2. Martin C. Brown, “**Python: The Complete Reference**” McGraw Hill Publications, 1st Edition, 2021.
3. Paul J. Deitel, Harvey Deitel “**Python for Programmers**”, Pearson publications, 1st Edition, 2020.
4. Yashavant Kanetkar, “Let Us Python”. BPB Publications, 5th edition,2023.

C. LESSON PLAN for THEORY LECTURE

S.No.	CONTENT	NUMBER OF LECTURES	Unit
1	Introduction to Python and Environment Setup	1	**Unit-I (10)**
2	Python Syntax and Basic Data Types	1	
3	Variables, Operators, and Expressions	1	
4	Control Structures and Conditional Statements	1	
5	Loops and Iterations: For, While, For-else, Range function	1	
6	Sentinel-Controlled Iteration, break and continue Statements	1	
7	Defining Functions, Functions with Multiple Parameters	1	
8	Default Parameter Values, Keyword Arguments, Arbitrary Argument	1	
9	Lists and Tuples, Associated in-built functions, Dictionaries and Sets	2	
CLASS TEST-I			
10	Object-Oriented Programming (OOP) Fundamentals	1	**Unit-II (7)**
11	Classes and Objects	1	
12	Inheritance and Polymorphism	1	
13	Advanced OOP Concepts (e.g., Encapsulation, Abstraction)	1	
14	Generators and Iterators	1	
15	Functional Programming Techniques	1	
16	Modules and Packages	1	
17	Introduction to NumPy Arrays	1	**Unit-III (7)**
18	Array Operations and Functions, Indexing and slicing arrays	1	
19	Array Manipulation and Stacking, multidimensional arrays and matrices	1	
MID SEMESTER EVALUATION			
20	Creating and manipulating strings, String Operations	1	
21	Regular Expressions, pattern matching and substitution	1	
22	Working with Dates and Times	1	
23	Introduction to pandas Series and DataFrames	1	
24	Data Analysis with Pandas (e.g., Filtering, Grouping, Aggregation)	1	**Unit-IV (9)**
25	file handling in Python, Different types of files and their uses	1	
26	Updating text files: appending and modifying data	1	
27	Understanding serialization and deserialization	1	
28	Working with JSON files for data storage and retrieval	1	
29	Handling Exceptions using try and except statements	1	
30	Catching and handling specific exceptions, Raising custom exceptions	1	
31	‘finally’ clause for cleanup operations, Explicitly raising exceptions with ‘raise’	1	
32	Working with CSV files	1	
33	Introduction to Web Development with Python,	1	**Unit-V (7)**
34	Introduction to Flask Framework	1	
35	Defining routes and handling HTTP methods (GET, POST)	1	
36	Request and response objects in Flask	1	
37	Building Web Applications with Flask	1	
38	Introduction to Data Visualization	1	
39	Data Visualization with Matplotlib	1	

D. LESSON PLAN for LAB

Lab Class No.	Name of the Experiment	
1	Install Python and set up the development environment.; Write a Python program to print "Hello, World!"; Write a Python program to calculate the area of a circle given the radius.;	
2	Write a Python program to check if a number is even or odd.; Implement a simple calculator using conditional statements; Write a Python program to print the Fibonacci series using a for loop.	
3	Implement a function to check if a given string is a palindrome.; Perform various operations on lists (e.g., sorting, slicing).; Use dictionaries to store and retrieve student grades.	
4	Create a class to represent a book with attributes and methods.; Implement inheritance by creating subclasses for different types of books.; Write a generator function to generate the Fibonacci series.	
5	Use lambda functions, map, and filter to perform operations on a list.; Create a module that contains functions for mathematical operations.; Import and use functions from external packages (e.g., math, random).	
6	Create and manipulate NumPy arrays.; Perform basic operations and indexing on arrays.	
MID SEMESTER EVALUATION		
7	Implement string operations (e.g., concatenation, slicing).; Use regular expressions to validate email addresses.	
8	Read data from a text file and perform operations.; Handle exceptions for file operations and input validation.	
CLASS TEST-2 (LAB EVALUATION)		
9	**Mini Project**	a. Implement backend using python for Mini Project
10		b. Create a Flask application with routes and HTTP methods.; Deploy API for Mini Project
11		c. Integrate your backend with frontend using API for Mini Project
12		d. Mini Project Discussion and Evaluation
END SEMESTER EVALUATION		

E. SELF STUDY

Sr. No.	Topic	
1	Fibonacci Sequence with *Vedic Mathematics*: In Vedic Mathematics, there is a unique method to generate the Fibonacci sequence using a technique called "Urdhva-Tiryagbhyam" sutra. This sutra can help generate the nth Fibonacci number efficiently. Task: Write a Python function that uses the Urdhva-Tiryagbhyam sutra from Vedic Mathematics to calculate the nth Fibonacci number	Unit V
2	Implement a Python function that uses the *Chakravala method* to find the smallest positive integer y for a given input a. The function should return the values of x and y for which the equation $ax^2 + 1 = by^2$ holds true.	Unit II & III & V
3	Create an interactive quiz game using Python that tests players' knowledge of Indian knowledge systems, related to ancient and Vedic science.	
4	Any topic as suggested by Course instructor	