Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore
Shri Vaishnav Institute of Technology and Science
Choice Based Credit System (CBCS) in the Light of NEP-2020
Generic Elective

COURSECODE	CATEGORY	COURSE NAME	TEACHING &EVALUATION SCHEME								
			THEORY			PRACTICAL		L	T	P	CREDITS
			END SEM University Exam	Two Term Exam	Teachers Assessment*	END SEM University Exam	Teachers Assessment*				
GUEE301		**Basics of Programming**	60	20	20	0	0	4	0	0	4

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

Course Educational Objectives:

This course introduces students to the fundamental concepts of programming. Students will learn how to write simple programs, understand logical thinking, and explore the joy of creating software. The course will focus on providing a solid foundation in problem-solving and programming logic, using an accessible programming language like Python. By the end of the course, students will be able to write their own basic programs, understand programming concepts, and appreciate the creative aspects of coding.

Course Outcomes:

1. Understand the basics of programming and the role it plays in solving real-world problems.
2. Learn how to use control structures to make decisions and repeat tasks in programs.
3. Understand how to organize code efficiently using functions.
4. Learn how to store and manipulate multiple values efficiently using data structures.
5. Learn the principles of object-oriented programming and how to structure code using classes and objects.
6. Apply programming knowledge to creative and practical projects.

UNIT I
Introduction to Programming and Problem Solving

- Overview of programming languages and their uses. Concepts of algorithms
- Introduction to the Python programming language.
- Setting up the development environment (installing Python, using IDEs like Visual Studio Code or PyCharm).
- Writing and running the first program: "Hello, World!"
- Basic data types: integers, strings, and floating-point numbers.
- Variables and constants.
- Simple input/output operations.

UNIT II
Control Structures and Logic

- Conditional statements: if, elif, else.
- Logical operators: and, or, not.
- Loops: for loops and while loops.
- Break and continue statements.
- Introduction to debugging: understanding errors and fixing them.

UNIT III

Functions and Modular Programming

- Introduction to functions and their significance
- Defining and calling functions in Python.
- Function parameters and return values.
- The concept of scope: local and global variables.
- Recursion: an introduction to recursive functions.
- Modular programming

UNIT IV

Data Structures and Collections

Lists: creating, accessing, and modifying lists.
Tuples: immutable sequences.
Dictionaries: key-value pairs and their usage.
Sets: working with unique collections of items.
Iterating over collections with loops.

UNIT V

Introduction to Object-Oriented Programming

Introduction to classes, objects, methods, and attributes.
Defining classes and creating objects.
Instance variables vs class variables.
Inheritance and polymorphism
Encapsulation and abstraction.

Text Books:

1. "Python for Everybody" by Charles Severance (for beginners)

Lab Experiments: (Only for Practice)

1. Write a simple program that greets the user by name.
2. Create a basic calculator that performs addition, subtraction, multiplication, and division.
3. Write a program that checks if a number is prime.
4. Create a program that generates multiplication tables up to a given number.
5. Design a simple "Guess the Number" game using loops and conditions.
6. Write a function to calculate the factorial of a number.
7. Create a program that checks whether a string is a palindrome using functions.
8. Design a function to compute the Fibonacci sequence.
9. Create a program that manages a simple to-do list using lists.
10. Write a program that counts the frequency of each character in a given string using dictionaries.
11. Implement a program that checks for duplicate items in a list using sets.
12. Write a class to represent a "Book" with attributes like title, author, and price.
13. Create a class "Person" and derive a class "Student" that adds additional attributes.
14. Develop a simple banking system with classes for Account and Transaction.