Gyanmanjari Innovative University

Course Syllabus
Gyanmanjari Engineering College
Semester-4 (B.Tech)

Subject: Python Programming BETCE14307

Type of course: Major (Core)

Prerequisite: Programming fundamental, logic and problem-solving skills, mathematical logic.

Rationale:
Python is an open-source, high-level, general-purpose programming language used for software development. It is one of the most popular programming languages in the world today and known for its simplicity as well as rich library. It is a widely used programming language in various domains, such as Automation, Server-side Web Development, Tools Development, Game Programming, Blockchain, Data Science, Artificial Intelligence, Machine Learning, Big Data etc. It's relatively easy to learn to use and incredibly versatile. This course aims to teach the basics of Python programming. The course focuses on how to use the building blocks of Python programming to solve different problems. At the end of the course, students will be able to develop simple applications using Python programming. This knowledge will provide a solid foundation for exploring advanced applications of Python programming in the different domains mentioned above.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks
				Theory Marks		Practical Marks		CA	
CI	T	P	C	ESE	MSE	V	P	ALA	
4	0	2	5	60	30	10	20	30	150

Legends: CI-Class Room Instructions; T-Tutorial; P-Practical; C-Credit; ESE- End Semester Examinations; MSE – Mid Semester Examination; V – Viva; CA – Continuous Assessment; ALA – Active Learning Activity.

Continuous Assessment:

Sr. No	Active Learning Activities	Marks
1	**Identify a Problem or Opportunity:** (Change problem Identification) Start by identifying a specific problem or opportunity that could be addressed through an IT project. This could be a business challenge, a process that could be improved, or a new product or service that could be developed. Students have to upload project definitions with abstract in the GMIU portal.	5
2	**Define Project Scope and Objectives:** Once you've identified the problem or opportunity, define the scope of your project and set clear objectives. Determine what you want to achieve with the project, what deliverables are expected, and what resources (such as budget, time, and personnel) are available. Students have to upload basic details like what timing it needed for completing the project in GMIU portal.	5
3	**Develop a Project Plan:** Create a detailed project plan that outlines the tasks, milestones, and timeline for your project. Identify the roles and responsibilities of team members, allocate resources, and establish a communication plan to keep stakeholders informed. Students have to upload a JPEG or PNG file of your project blueprint in the GMIU portal.	10
4	**Execute and Monitor the Project:** With your project plan in place, start executing the project according to the plan. Monitor progress regularly, track milestones, and make adjustments to the plan as needed to ensure the project stays on track and meets its objectives. After completing the project students have to upload a project report (PDF file) with source code in the GMIU portal.	10
	Total	30

Course Content:

Sr.No	Coursecontent	Hrs.	Weightage %
1	**Unit 1: Introduction to Python:** Python Brief, Python's popularity and applications,Basic features and strengths,Setting Up Python,Installing Python,Running your first Python program,Basic Syntax and Variables,Variables and data types (int, float, str, bool),Basic input/output, Operators available in python.	7	12%

2	**Unit 2: Control Structures and Functions**: Conditional Statements, If statements,If else statement,Nested if else,if-elif-else Ladder, Looping statements,For loops,While, loops, Loop control statements (break, continue), Functions, Defining and calling functions, Function parameters and return values.Types of Functions.	10	17%
3	**Unit 3: Data Structures:** Lists, Creating and modifying lists, Accessing elements, List methods (append, extend, remove, etc.), Tuples and Sets, Tuple creation and immutability, Set creation and uniqueness, Dictionaries, Dictionary creation and key-value pairs, Dictionary methods (keys, values, items).	12	19%
4	**Unit 4: File Handling and Modules:** Working with Files, Opening and closing files, Reading and writing text files, Exception handling for file operations, Introduction to Modules, Importing modules, Creating and using custom modules,PIP.	15	22%
5	**Unit 5: Advanced Topics OOP:** Object-Oriented Programming (OOP), Classes and objects Inheritance and polymorphism, Exception Handling, Handling exceptions with try...except blocks, Introduction to Libraries and Frameworks, Overview of popular Python libraries (e.g., NumPy, pandas, Matplotlib), Introduction to web frameworks (e.g., Flask, Django)	16	30%
	Total	60	100%

Distribution of Theory Marks (**Revised Bloom's Taxonomy**)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage %	20%	35%	35%	10%	-	-

Course Outcome:

After learning the course, the students should be able to:	
CO1	Understanding Python's popularity, applications, basic features, and strengths, as well as be able to set up Python.
CO2	Will demonstrate proficiency in constructing conditional statements.
CO3	Proficient in creating, modifying, and accessing elements in lists, employing list methods.

CO4	Demonstrate competence in handling files through operations like opening, closing, reading, and writing text files.
CO5	Proficiency in OOP concepts and will gain insight into popular Python libraries.

List of Practical

Sr. No	Description	Unit No	Hrs
1	Installation of python plug-in Print Hello, World! Program.	1	1
2	Simple calculator (addition, subtraction, multiplication, division)	1	1
3	Temperature converter (Celsius to Fahrenheit)	1	1
4	Basic string operations (concatenation, slicing)	1	1
5	Checking if a number is even or odd	1	1
6	Find the maximum of two numbers using if-else	2	1
7	Print numbers from 1 to 10 using a for loop	2	1
8	Calculate the factorial of a number using a while loop	2	1
9	Create a function to check if a number is prime	2	2
10	Function to calculate the area of a rectangle	2	1
11	Create a shopping list using a list	3	1
12	Calculate the average of a list of numbers	3	1
13	Create a tuple of student information (name, age, grade)	3	2
14	Build a set of unique colors from a list	3	1
15	Create a dictionary of contact information (name, email, phone)	3	1
16	Read and display the contents of a text file	4	1
17	Write data to a text file (e.g., save user information)	4	2
18	Create a simple calculator module (add, subtract, multiply)	4	1
19	Import the math module to calculate square roots	4	1
20	Build a custom utility module for common functions (e.g., string manipulation)	4	1
21	Create a class for representing a bank account	5	1
22	Inherit from a Shape class to create specific shapes (e.g., Circle, Square)	5	1
23	Handle exceptions in a user input validation function	5	1
24	Use NumPy to perform basic array operations (e.g., addition, multiplication)	5	2
25	Initialize a simple Flask web application with a route and template	5	2
		Total	30

Instructional Method:

The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by blackboard, may also use any of tools such as demonstration, role-play, Quiz, brainstorming, MOOCsete. From the content 10% topics are suggested for flipped mode instruction.

Students will use supplementary resources such as online videos, NPTEL/SWAYAM videos, e-

courses, Virtual Laboratory The internal evaluation will be done on the basis of Active Learning Assignment. Practical/Viva examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

Reference Books:

[1] Introduction to Computer Science Using Python: Charles Dierbach.
[2] Python Programming: An Introduction to Computer Science: John Zelle
[3] Introduction to Programming Using Python: Liang Y. Daniel