



Gyanmanjari
Innovative University

Syllabus
Gyanmanjari Institute of Technology
Sem – 3 (B.Tech)

Subject: Python Programming: Building Scalable and Intelligent Applications- BET2CS13306

Type of course: Professional Core

Prerequisite: Programming fundamentals, logic and problem-solving skills, mathematical logic

Rationale:

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it suitable for a wide range of applications. Python is widely used in web development, data analysis, machine learning, artificial intelligence, automation, and scientific computing. Its clear syntax and dynamic typing enable rapid development and easy debugging, making it ideal for both beginners and experienced programmers. Python offers a vast standard library, strong community support, and seamless cross-platform compatibility. Tools like IDLE, PyCharm, and Jupyter Notebook support interactive programming. As an open-source language with growing industry demand, Python is a valuable skill in today's technology landscape.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks		Total Marks
CI	T	P	C	SEE	CCE	
4	0	2	5	100	50	150

Legends: CI-Class Room Instructions; T – Tutorial; P - Practical; C – Credit; SEE - Semester End Evaluation; CCE-Continuous and Comprehensive Evaluation.



Course Content:

Sr. No	Course Content	Hrs.	% Weightage
1	Introduction to Python: <u>Theory Topics:</u> Python Introduction, Python's Popularity, Features of Python, Applications of Python, Why Python, Installing Python, Introducing Python Frameworks, Python Program Structure, Basics of Python: Working with programming modes, Interactive Mode, Scripting Mode, Comments, Tokens, Keywords, Identifiers, Literals, Variables & Constants, Standard Data Types(int, float, str, bool), Processing Input and Output, print, raw_input, Working with Python files, Indentation for a block of statements, Operators and Expression, String: Creating String, Formatting Strings, String Formatting Methods, Math and Comparisons, Conversion of Numbers to String. <u>Practical:</u> 1. Write a program that displays the message "Hello, World!" on the screen. 2. Write a program to perform simple calculations - addition, subtraction, multiplication, and division - using two input numbers. 3. Write a program converts a temperature from Celsius to Fahrenheit using the formula: Fahrenheit = (Celsius $\times$ 9/5) + 32. 4. Write a program that takes the user's first and last name separately and combines them to display the full name. 5. Write a program that joins two or more strings into a single string using the + operator. 6. Write a program that extracts a portion of a string using slicing syntax string[start: end]. 7. Write a program that takes a sentence as input and displays a specific portion (e.g., first 5 characters). 8. Write a program that reverses a given string using slicing with a negative step string [-1]. 9. Write a program to find the length of a string. 10. Write a program to find the sum of the first five positive integers. 11. Write a program to find an average age of 5 employees. 12. Write a program to find a data type of "Hello World!". 13. Write a program to find the perimeter to a rectangle. 14. Write a program to find the area of square and area of triangle. 15. Write a program to calculate simple interest and compound interest.	18	20%



Evaluation Method:			
Sr. No.	Evaluation Methods	SEE	CCE
1	Random Code Challenges (Alien Cards): Students will be provided with small cards containing unsolved examples or problems from reference books, or a spinner will be used to allocate a random micro-challenge. Students must optimize and refactor the given code to improve performance, readability, and efficiency while ensuring that the expected output remains unchanged.	20	-
2	Active Learning Activity (ALA): Python Concept Drill – Code and Reflect Students will receive a worksheet or live coding environment containing challenges such as Syntax Debug and Data Type Detective. Students must complete the allocated challenge and upload the completed activity to the GMIU Web Portal.	-	10
Total		20	10
Examination Style:			
1. Random Code Challenges (Alien Cards) (20 Marks) Students will be provided with small cards containing unsolved examples or problems selected from reference books, or a spinner will be used to allocate a random micro-challenge. Students must refactor and optimize the given Python code to improve its performance, readability, and efficiency while ensuring that the expected output remains unchanged.			
2. Active Learning Activity (ALA): Python Concept Drill – Code and Reflect (10 Marks): Students will receive a worksheet or live coding environment containing challenges such as Syntax Debug and Data Type Detective. Each student must complete the allocated challenge, identify and correct the errors, execute the program, and briefly explain the applied solution. The completed activity must be uploaded to the GMIU Web Portal within the specified timeline.			



2	Python Programming Constructs: <u>Theory Topics:</u> Conditional Statements (if statement, if-else statement, if-elif statement, Nested if statement), Looping statements (While, Else, For loop, While loop, Loop else statement), Break and Continue, Range Function, Mathematical Operations, Functions, Defining and calling functions, Function parameters and return values (Pass by reference vs Pass by value), Types of Functions, Default Argument Values, Scope of Variables, The Return Statements. <u>Practical:</u> 16. Write a program to compare two numbers and find the maximum using if-else statements. 17. Write a program to determine whether a given number is even or odd. 18. Write a program to determine whether a given number is positive, negative, or zero. 19. Write a program to display numbers from 1 to 10 using a for loop. 20. Write a program that counts how many even and odd numbers are present in a given series of numbers. 21. Write a program to find the factorial of a number entered by the user using a while loop. 22. Write a Python program to generate and display the Fibonacci series for numbers between 0 and 50. 23. Write a program that defines a function to calculate the area of a rectangle using its length and width. 24. Write a program that defines a function to determine whether a given number is prime or not. 25. Write a program that converts temperatures between Celsius and Fahrenheit in both directions — from Celsius to Fahrenheit and from Fahrenheit to Celsius using user defined functions.	18	20%
---	---	----	-----



Evaluation Method:			
Sr. No.	Evaluation Methods	SEE	CCE
1	Code Carnival: The Logic Fair: Students will participate in a Python-powered carnival where each booth contains a mini-challenge or puzzle based on real-life problems using conditional statements, loops, and functions.	20	-
2	Active Learning Activity (ALA): Python in Daily Life – Solve, Simulate, and Automate: Students will complete real-life-inspired Python coding challenges by applying conditional statements, loops, and functions to develop logical, reusable, and well-structured solutions and upload the completed activity to the GMIU Web Portal.	-	10
	Total	20	10
Examination Style:			
1. Code Carnival: The Logic Fair (20 Marks): Students will participate in a Python-powered carnival where each booth contains a mini-challenge or puzzle based on real-life problems. Students must solve the allocated challenges using appropriate conditional statements, loops, and functions. This activity simulates real-world logic implementation, gamifies the coding experience, and promotes deeper learning through scenario-based problem-solving.			
2. Python in Daily Life – Solve, Simulate, and Automate (10 marks) Students will complete real-life-inspired Python coding challenges such as a Smart Alarm System, Movie Ticket Counter, or similar applications. They must apply conditional statements, loops, and functions to develop logical, reusable, and well-structured solutions. The completed source code and required activity details must be uploaded to the GMIU Web Portal within the specified timeline.			
3	Data Structures: <u>Theory Topics:</u> Lists (Creating and modifying list, accessing list, Joining List, Replicating List, List Slicing, List methods (append, extend, remove, etc.)), Tuple (Creating Tuple, Accessing Tuples, Joining Tuples, Replicating Tuples, Tuple Slicing, Tuple immutability), Sets (set creation and uniqueness, Accessing and Joining, Replicating and	18	20%



Slicing), Dictionary (Dictionary creation and key-value pairs, Dictionary methods (keys, values, items), working with dictionary, properties.

Practical:

26. Write a program to build a shopping list using a list.
27. Write a program to compute the average value from a list of numbers.
28. Write a program that defines a tuple containing a student's name, age, and grade.
29. Write a program to access a third element in a tuple.
30. Write a program to create a set of unique color names from a list.
31. Write a program that stores contact details like name, email, and phone number using a dictionary.
32. Write a python program to print all dictionary keys.

Evaluation Method:

Sr. No.	Evaluation Methods	SEE	CCE
1	Build Your Own Mini Data Organizer in Python: Students will create a Python-based organizer tool to manage collections such as books, students, groceries, or tasks using lists, tuples, sets, and dictionaries.	20	-
2	Active Learning Activity (ALA): Virtual Shopping Cart Simulation: In a group of two, students will develop a virtual shopping cart using Python data structures and upload the completed activity to the GMIU Web Portal.	-	10
Total		20	10

Examination Style:

1. Build Your Own Mini Data Organizer in Python (20 Marks)

- Students will create a Python-based organizer tool that simulates managing a collection of items such as books, students, groceries, or tasks. They must use lists, tuples, sets, and dictionaries appropriately to store, organize, access, update, and display the required data.

2. Active Learning Activity (ALA): Virtual Shopping Cart Simulation (10 Marks):

- In a group of two, students will build and manage a virtual shopping cart using Python data structures. They will use a list to manage cart items, tuples to store product details, a set to manage promo codes, and a dictionary to track product quantities and prices. The completed source code and activity details must be uploaded to the GMIU Web Portal within the specified timeline.



4	Exception, File Handling and Modules: <u>Theory Topics:</u> Exceptions: Introduction, Errors, Exceptions, Handling Exceptions, Raising Exceptions, Try.....Finally, Raise, Assert, Finally blocks, The with statement, Working with Files, Opening and closing files, Reading and writing text files, Exception handling for file operations, Introduction to Modules, Exploring built-in modules. importing modules, Namespaces and Scoping, Creating and using custom modules, PIP. <u>Practical:</u> 33. Write a program to divide one number from another number with exception handling and show ZeroDivisionError. 34. Write a program that validates user input and handles exceptions appropriately. 35. Write a program to accept two numbers from the user and handle ValueError if non-numeric input is entered. 36. Write a program that raises a custom exception using the raise statement if the user's age is below 18. 37. Write a program using assert to ensure that a number entered is positive. 38. Write a Python program to demonstrate the use of the finally block even when an exception occurs. 39. Write a program to Read and display the contents of a text file. 40. Write a program that collects user input and writes the data to a text file. 41. Write a program to create a simple calculator module (add, subtract, multiply). 42. Write a program that uses the math module to find the square root of a number. 43. Write a program to build a custom utility module for common functions (e.g., string manipulation). Evaluation Method: <table border="1"> <thead> <tr> <th>Sr. No.</th><th>Evaluation Methods</th><th>SEE</th><th>CCE</th></tr> </thead> <tbody> <tr> <td>1</td><td>Code Optimization: Students will solve and optimize the given Python problems related to exception handling and file handling.</td><td>20</td><td>-</td></tr> <tr> <td>2</td><td>Active Learning Activity (ALA): Exception Comic Strip Creator: Individual students will create a story-based visual representation of Python exception handling, discuss their ideas, share their insights with the class, and upload the completed activity to the GMIU Web Portal.</td><td>-</td><td>10</td></tr> <tr> <td></td><td>Total</td><td>20</td><td>10</td></tr> </tbody> </table>	Sr. No.	Evaluation Methods	SEE	CCE	1	Code Optimization: Students will solve and optimize the given Python problems related to exception handling and file handling.	20	-	2	Active Learning Activity (ALA): Exception Comic Strip Creator: Individual students will create a story-based visual representation of Python exception handling, discuss their ideas, share their insights with the class, and upload the completed activity to the GMIU Web Portal.	-	10		Total	20	10	18	20%
Sr. No.	Evaluation Methods	SEE	CCE																
1	Code Optimization: Students will solve and optimize the given Python problems related to exception handling and file handling.	20	-																
2	Active Learning Activity (ALA): Exception Comic Strip Creator: Individual students will create a story-based visual representation of Python exception handling, discuss their ideas, share their insights with the class, and upload the completed activity to the GMIU Web Portal.	-	10																
	Total	20	10																



	Examination Style: Code Optimization (20 Marks) - Students will be provided with two suboptimal Python programs based on exception handling and file handling. They must optimize and refactor the given code to improve performance, readability, and efficiency while ensuring that the expected output remains unchanged. Active Learning Activity (ALA): Exception Comic Strip Creator (10 Marks): - Individual students will create a comic strip or storyboard illustrating how Python handles exceptions such as KeyError and ZeroDivisionError. Characters representing try, except, and finally will explain the program flow through speech bubbles in a story-based format. Students may use paper, a whiteboard, or digital tools such as Canva, Pixton, or Storyboard That. They must discuss their ideas, share their insights with the class, and upload the completed activity to the GMIU Web Portal within the specified timeline.		
5	Advanced Topics OOP: <u>Theory Topics:</u> Object-Oriented Programming (OOP), Creating Classes, Creating Instance Objects, Accessing Attributes, Destroying objects, Inheritance and polymorphism, Exception Handling, Handling Exception 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) <u>Practical:</u> - Write a Python program that defines a class to represent a bank account with relevant attributes and methods. - Create specific shape classes like Circle and Square by inheriting from a general Shape class in Python. - Use NumPy in a Python program to carry out basic operations on arrays, such as addition and multiplication. - Create a basic Flask web application in Python that includes a route and renders a template. - Create a class named Teacher with the following specifications: Private Members: name (string, max 20 characters), subject (string, max 10 characters), basic, da, hra (float values), salary (float, calculated) Member Function: calculate () – an inline method that computes the total salary as the sum of basic, da, and hra.	20	20%



	Public Methods: readdata() – accepts input values for the teacher's details and calls the calculate() method. displaydata() – displays all the stored information, including the calculated salary. 49. Create a class named Paper with width and height as attributes. The class should include methods to calculate: Area as: width * height Perimeter as: 2 * (width + height) Pass an object of this class as an argument to demonstrate calculations using another function or class. Evaluation Method: <table> <tr> <th>Sr. No.</th><th>Evaluation Methods</th><th>SEE</th><th>CCE</th></tr> <tr> <td>1</td><td>My First Python Assistant: Students will create a simple Python assistant using a class, object creation, method calls, exception handling, and one suitable Python library.</td><td>20</td><td>-</td></tr> <tr> <td>2</td><td>Active Learning Activity (ALA): Build-a-Blog with Flask or Django – Mini Project: In a group of two, students will develop a functional web application using Flask or Django and upload the completed project to the GMIU Web Portal.</td><td>-</td><td>10</td></tr> <tr> <td></td><td>Total</td><td>20</td><td>10</td></tr> </table> Examination Style: My First Python Assistant (20 Marks) - Students will create a simple Python assistant program using a class, object creation, basic method calls, exception handling, and one suitable Python library. The assistant should perform the assigned operations correctly and handle possible errors appropriately. Active Learning Activity (ALA): Build-a-Blog with Flask or Django – Mini Project (10 Marks): - In a group of two students create a mini blog, to-do app, or feedback form using Flask or Django using object-oriented classes for models. Also adding simple styling and testing the app locally. Understand how backend frameworks use OOP and exception handling to manage real-world applications.	Sr. No.	Evaluation Methods	SEE	CCE	1	My First Python Assistant: Students will create a simple Python assistant using a class, object creation, method calls, exception handling, and one suitable Python library.	20	-	2	Active Learning Activity (ALA): Build-a-Blog with Flask or Django – Mini Project: In a group of two, students will develop a functional web application using Flask or Django and upload the completed project to the GMIU Web Portal.	-	10		Total	20	10		
Sr. No.	Evaluation Methods	SEE	CCE																
1	My First Python Assistant: Students will create a simple Python assistant using a class, object creation, method calls, exception handling, and one suitable Python library.	20	-																
2	Active Learning Activity (ALA): Build-a-Blog with Flask or Django – Mini Project: In a group of two, students will develop a functional web application using Flask or Django and upload the completed project to the GMIU Web Portal.	-	10																
	Total	20	10																



Suggested Specification table:

Distribution of Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage %	10%	15%	20%	10%	15%	30%

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcome:

After learning the course, the students should be able to:	
CO1	Develop basic Python programs using syntax, data types, input/output operations, and string manipulation.
CO2	Apply conditional statements, loops, and functions to solve logical programming problems.
CO3	Use lists, tuples, sets, and dictionaries for efficient data storage and manipulation.
CO4	Implement exception handling, file operations, and modules to develop robust Python programs.
CO5	Apply object-oriented programming, Python libraries, and frameworks to develop advanced applications.

Instructional Method:

The course delivery method will depend upon the requirement of content and needs of students. The teacher, in addition to conventional teaching methods by black board, may also use any tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

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 the Active Learning Assignment.

Practical/Viva examination will be conducted at the end of semester for evaluation of performance of students in the laboratory.



Reference Books:

- [1] Introduction to Computer Science Using Python - A Computational Problem - Solving Focus, Charles Dierbach, Wiley.
- [2] Python Programming: An Introduction to Computer Science, Fourth Edition, John M. Zelle, Shroff/Franklin, Beedle & Associates.
- [3] Head First Programming: A Learner's Guide to Programming Using the Python Language, Second edition, David Griffiths, Paul Barry, O'Reilly.
- [4] Learning Python, Fifth edition, Lutz M Shroff.
- [5] Beginning Python, Payne, J. (2010), Wiley Publishing.



Suggested Assessment Guidelines:

Module-1: Introduction to Python

Random Code Challenges (Alien Cards) (20 Marks)		
Criteria	Description	Marks
Code Analysis	Correctly identifies the logic and areas for improvement in the given code.	5
Code Optimization	Refactors the code to improve readability, efficiency, and performance.	5
Error Handling & Best Practices	Applies appropriate exception handling and follows Python coding standards.	5
Output Verification & Explanation	Produces the correct output and clearly explains the optimization made.	5
Total		20

Module-2: Python Programming Constructs

Code Carnival: The Logic Fair (20 Marks)		
Criteria	Description	Marks
Problem Analysis	Correctly understands the given scenario and identifies the required logic.	5
Program Development	Implements the solution using appropriate conditional statements, loops, and functions.	5
Code Quality	Writes readable, efficient, and well-structured Python code following coding standards.	5
Output & Explanation	Produces the correct output and clearly explains the implemented logic.	5
Total		20

Module-3: Data Structures

Build Your Own Mini Data Organizer in Python (20 Marks)		
Criteria	Description	Marks
Data Structure Design	Effectively uses lists, tuples, sets, and dictionaries to organize data.	5
Program Implementation	Implements the required functionalities correctly using appropriate Python concepts.	5
Code Quality	Writes clean, readable, and well-structured Python code following best practices.	5
Output & Functionality	Produces the expected output and demonstrates the working of the data organizer.	5
Total		20



Module-4: Exception, File Handling and Modules

Code Optimization (20 Marks)		
Criteria	Description	Marks
Code Analysis	Identifies inefficiencies and areas for improvement in the given Python programs.	5
Code Optimization	Refactors the code to improve readability, efficiency, and maintainability without changing the output.	5
Exception & File Handling	Applies appropriate exception handling and follows best practices for file operations.	5
Output Verification & Justification	Produces the correct output and clearly explains the optimization techniques used.	5
Total		20

Module-5: Advanced Topics OOP

My First Python Assistant (20 Marks)		
Criteria	Description	Marks
Class & Object Design	Creates appropriate classes, objects, and methods following OOP principles.	5
Program Functionality	Implements the required assistant features using methods and one Python library.	5
Exception Handling	Uses appropriate exception handling to manage invalid inputs and runtime errors.	5
Output & Code Quality	Produces the expected output with clean, readable, and well-structured code.	5
Total		20

