Course Syllabus
Gyanmanjari College of Computer Application
Semester-2(MCA)

Subject: Python Programming - MCAXX11510

Type of course: Major(Core)

Prerequisite: Basic concept of Programming Language

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 widely used programming language in various domains, such as Automation, Server-side Web Development, Tools Development, Game Programming, Block chain, 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.
To develop proficiency in creating based applications using the Python Programming Language.
To be able to understand the various data structures available in Python programming language and apply them in solving computational problems.

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	
3	0	4	5	60	30	10	20	30	150

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

Course Content:

Sr. No	Course content	Hrs	% Weightage
1	**Introduction to Python** The basic elements of Python, Objects, expressions and numerical Types, Variables and assignments, IDLE Branching programs, Strings and Input, Iteration Structured Types, Mutability and Higher-order Functions: Tuples, Ranges, Lists and Mutability (Cloning and list comprehension), Strings, Tuples and Lists, Dictionaries	8	20
2	**Functions, Exception, Modules and Files** Functions: Difference between a Function and a Method, Defining a Function, Calling a Function, Returning Results from a Function, Returning Multiple Values from a Function, Functions are First Class Objects, Pass by Object Reference, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Variable Length Arguments, Local and Global Variables, The Global Keyword, Passing a Group of Elements to a Function, Recursive Functions, Anonymous Functions or Lambdas (Using Lambdas with filter() Function, Using Lambdas with map() Function, Using Lambdas with reduce() Function), Function Decorators, Generators, Structured Programming, Creating our Own Modules in Python, The Special Variable name_____ Exceptions: Errors in a Python Program (Compile-Time Errors, Runtime Errors, Logical Errors), Exceptions, Exception Handling, Types of Exceptions, The Except Block, the assert Statement, User-Defined Exceptions, Logging the Exceptions Files: Files, Types of Files in Python, Opening a File, Closing a File, Working with Text Files Containing Strings, Knowing Whether a File Exists or Not, Working with Binary Files, The with Statement, Pickle in Python, The seek() and tell()Methods	15	30
3	**Classes and Object-oriented Programming** Abstract Data Types and classes, Inheritance, Encapsulation and Information hiding Mortgages and Extended Examples Case Study: Banking Application	7	15
4	**Regular Expressions in Python** REs and Python: Regular Expressions, Sequence Characters inRegularExpressions,QuantifiersinRegularExpressions,Special CharactersinRegularExpressions,UsingRegularExpressions on Files, Retrieving Information from a HTML File Case Study: Screen Scrapping	7	15

5	**Python's Database Connectivity** Verifying the MySQLdB Interface Installation, Working with MySQL Database, Using MySQL from Python, Retrieving All Rows from a Table, Inserting Rows into a Table, Deleting Rows from a Table, Updating Rows in a Table, Creating Database Tables through Python	8	20

Continuous Assessment:

Sr. No	Active Learning Activities	Marks
1	**Code Challenge:** A "code challenge" refers to a problem or task in the python programming that individuals or groups are invited to solve. These challenges are designed to test and improve coding skills, problem-solving abilities, and algorithmic thinking. Student have to upload it on GMIU web portal.	10
2	**Debugging:** Student will be assign a code containing intentional error, student need to Identify error, correct and upload on GMIU web portal.(Group of Four or individual)	10
3	**Mini-Project:** Student have to prepare a Mini-project (Group of four or individual) on given definition or they can choose its own definition. Student have to upload abstract and project in GMIU web portal.	10
Total		30

Suggested Specification table with Marks (Theory) : 60

Distribution of Theory Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage	30%	25%	25%	10%	10%	0

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 a bove table.

Course Outcome:

After learning the course the students should be able to:	
CO1	Develop programs to solve the given simple computational problems.
CO2	Apply control flow structures to solve the given problems.
CO3	Implement data structures lists, tuples, sets and dictionaries to solve the given problems.

CO4	Apply modular programming approach to solve given problems using user-defined Functions.
CO5	Perform string manipulation and file operations to solve a given problem.

List of Practical

Sr.No.	Title	Unit	Approx. Hrs. Required
1	Write a program in python to swap two variables without Using temporary variable.	1	1
2	Write a Python Program to Convert Decimal to Binary, Octal And Hexadecimal	1	1
3	Write a program in python to find out maximum and Minimum number out of three user entered number.	1	1
4	Write a program which will allow user to enter 10 numbers and display largest odd number from them. It will display appropriate message in case if no odd number is found.	1	1
5	Write a Python program to check if the number provided by The user is an Armstrong number.	1	1
6	Write a Python program to check if the number provided by The user is a palindrome or not.	1	1
7	Write a Python program to perform following operation on given string input: a) Count Number of Vowel in given string b) Find and replace operation	1	4
8	Write a program in python to implement Factorial series up to User entered number. (Use recursive Function)	2	1
9	Write a program in python to implement Fibonacci series up To user entered number. (Use recursive Function)	2	1
10	Write a program in Python to implement readline, readlines, writeline and writelines file handling mechanisms.	2	2
11	Write a program in python to implement Salary printing file read operation. (File format: Employee No, name, deptno, basic, DA, HRA, Conveyance) should perform below operations. a) Print Salary Slip forgiven Employee Number b) Print Employee List for Given Department Number	2	4
12	Write a program in python to implement Railway Reservation System using file handling technique. System should perform below operations. a. Reserve a ticket for a passenger. b. List in formation all reservations done for today's trains.	2	4
13	Write a program which will implement decorators for Functions and methods in python.	2	2
14	Write a program to read CSV file and generate output using HTML table.	2	2

15	Create Web Database Application "Address Book" with options to a) add/ insert a record b) modify a record c) display a record d) delete a record	5	8
16	Create Web Database Application "Event Registration" with options to a) Event Registration b) Cancel Registration c) display a record	5	8
17	Write a program in python to implement simple interest and compound interest values on chart using PyLab. Show the difference between both.(Note: Use of object oriented Paradigm is compulsory.)	3	10
	a) Recognize following strings bit, but, bat, hit, hat or hut		
	b) Match any pair of words separated by a single space, that is, first and last names.		
	c) Match any word and single letter separated by a comma and single space, as in last name, first initial.		
	d)Match simple Web domain names that begin with www and end with a ".com" suffix; for example, www.yahoo.com. Extra Credit: If your regex also supports other high-level domain names, such as .edu, .net, etc. (for example: www.foothill.edu).		
	e) Match a street address according to your local format (keep your regex general enough to match any number of street words, including the type designation). For example, American street addresses use the format: 1180 Bordeaux Drive. Make your regex flexible enough to support multi-word street names such as: 3120 Dela Cruz Boulevard.		
18	Create utility script to process telephone numbers such that	4	8
	a. Area codes (the first set of three-digits and the accompanying hyphen) are optional, that is, your regex should match both 800-555-1212 as well as just 555-1212.		
	b. Either parenthesized or hyphenated area codes are supported, not to mention optional; make your regex match 800-555-1212,555-1212, and also(800)555-1212.		
		Total	**60**

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 black board, may also use any of tools such as demonstration, role play, Quiz, brain storming, 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 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] Wesley J Chun, Core Python Applications Programming, 3rd Edition. Pearson
[2] Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python, Pearson
[3] Luke Sneeringer, Professional Python, WROX
[4] Doug Hellmann, The python 3 standard Library by example, Pearson Education
[5] Alex Martelli, Python Cookbook, O'REILLY
[6] Laura Cassell, Python Projects, WROX