



**FORMAT FOR COURSE CURRICULUM**

**Course Title: Programming in Python**

**Credit Units: 4**

**Course Level: UG**

**Course Code: TO BE ISSUED**

L	T	P/S	SW/F W	TOTAL CREDIT UNITS
3	0	2	-	4

**Course Objectives:**

1. Learn Syntax and Semantics and create Functions in Python
2. Handle Strings and Files in Python
3. Understand Lists, Dictionaries and Regular expressions in Python
4. Implement Object Oriented Programming concepts in Python
5. Build Web Services and introduction to Network and Database Programming in Python

**Pre-requisites:** Student should be familiar with fundamental programming constructs in any programming language like C, C++ etc.

**Course Contents/Syllabus:**

	Weightage (%)
<b>Module I</b> Introduction: Features, Installation of Python, Basic Syntax, Variable and Data Types, Operators	15%
<b>Module II</b> Conditional Statements: If, If- else, Nested if-else, Looping: For, While, Nested loops, Control Statements: Break, Continue String Manipulation: Accessing Strings, Basic Operations, Function and Methods Input-Output: Printing on screen, Reading data from the keyboard, Opening and closing file, Reading and writing files	20%
<b>Module III</b> Functions: Defining a function, Calling a function, Types of functions, Function Arguments, Global and local variables Functions, Class and object, Attributes, Inheritance	25%

<b>Module IV</b>	<b>25%</b>
Lists: Introduction, Accessing list, Operations, Working with lists, Function and Methods Tuple: Introduction, Accessing tuples, Operations, Functions and Methods Dictionaries: Introduction, Accessing values in dictionaries, Working with dictionaries, Properties, Functions	
<b>Module V</b>	<b>15%</b>
Database: Introduction, Connections, Executing queries GUI Programming: Introduction, Tkinter programming, Tkinter widgets, Button, Radio Buttons, Checkboxes, Functions	

### **Student Learning Outcomes:**

1. Examine Python syntax and semantics and be fluent in the use of Python flow control and functions
2. Demonstrate proficiency in handling Strings and File Systems
3. Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.
4. Interpret the concepts of Object-Oriented Programming as used in Python
5. Implement exemplary applications related to Network Programming, Web Services and Databases in Python

### **Pedagogy for Course Delivery:**

1. Chalk and Talk
2. LCD/SMART Boards
3. Web Resources
4. Student Seminars
5. Guest Lectures

### **Lab/ Practicals details, if applicable:**

#### **List of Experiments:**

- Write a program to read two integers and find the sum, diff, mult and div.
- Write a program to find the simple interest for a given value P, T and R. The program must take the input from the user.
- Write a python program to generate first n Fibonacci number and factorial of n using functions
- Write a python program to count the number of Characters in a given string and hence to display the characters of string in reverse order.
- Write a python program to
  - To create two new files f1 and f2

- To read and display the contents , count the number of lines and find the word whose count is more in f1 and f2 respectively.
- To create and display the file f3 which is a combination of f1 and f2.
- Write a Python program to convert a list of characters into a string.
- Write a Python program to find the index of an item in a specified list.
- Write a Python program to get the frequency of the elements in a list.
- Write a Python program to convert a list of tuples into a dictionary
- Write an Object-Oriented Python program to create two Time objects: *current Time*, which contains the current time; and *bread Time*, which contains the amount of time it takes for a bread maker to make bread. Then we'll use add Time to figure out when the bread will be done. Write the print Time function to display the time when the bread will be done by the bread maker.
- Design a Python program as described below:
  - a. Create a class called Palindrome.
  - b. In your Palindrome class, create a method called *reverse()* which takes a string argument. Your method should return the reverse of the argument as a string.
  - c. Create a second method in Palindrome called *isPalindrome()* which takes a string argument. This method should return True if the argument is a palindrome and False otherwise.
  - d. Write some code to test your new Palindrome class and print out results of your testing to the user. Give some consideration to what sort of strings you might want to use for your testing.
- A bookshop details contains the Title of the book and Number of copies of each title. As books are added to the shop, the number of copies to each should increase and as books are sold, the number of copies in each should decrease. Implement this scenario using dictionary data type in Python.
- What is Google Geocoding Services? Write and Explain Python Program to prompt user for a search string (city , call the Geocoding API and extracting information from the returned XML).

**Assessment/ Examination Scheme:**

Theory L/T (%)	Lab/Practical/Studio (%)	End Term Examination
75%	25%	70%

**Theory Assessment (L&T):**

Continuous Assessment/Internal Assessment					End Term Examination
Components (Drop down)	Class Test	Home Assignment	Attendance	Viva	
Weightage (marks)	10	7	5	8	70

**Lab/ Practical/ Studio Assessment:**

	Continuous Assessment/Internal Assessment				End Term Examination		
Components (Drop down)	PR	LR	VIVA-VOCE	ATTENDANCE	PERFORMANCE	VIVA	TOTAL
Weightage )	10	10	5	5	35	35	70

**Text Reading:**

1. Charles R. Severance, "Python for Everybody: Exploring Data Using Python 3", 1st Edition, Create Space Independent Publishing Platform, 2016.
2. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2nd Edition, Green Tea Press, 2015.
3. Reema Thareja, "Python Programming Using Problem Solving Approach" Oxford university press, 2017

**References:**

1. Charles Dierbach, "Introduction to Computer Science Using Python", 1st Edition, Wiley India Pvt Ltd. ISBN-13: 978-8126556014
2. Mark Lutz, "Programming Python", 4th Edition, O'Reilly Media, 2011. ISBN-13: 978-9350232873
3. Wesley J Chun, "Core Python Applications Programming", 3rd Edition, Pearson Education India, 2015. ISBN-13: 978-9332555365
4. Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich, "Data Structures and Algorithms in Python", 1<sup>st</sup> Edition, Wiley India Pvt Ltd, 2016. ISBN-13: 978- 8126562176

**Additional Reading:**

- NPTEL
- SWAYAM

**Any other Study Material:**

- TUTORIALS POINT