

OBJECT ORIENTED PROGRAMMING THROUGH PYTHON LABORATORY

III Semester: AE								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
AITB08	Core	L	T	P	C	CIA	SEE	Total
		1	0	2	2	30	70	100
Contact Classes: 12	Tutorial Classes: Nil	Practical Classes: 24			Total Classes: 36			
<p>OBJECTIVES: The course should enable the students to:</p> <ol style="list-style-type: none"> I. To be able to introduce core programming basics and program design with functions using Python programming language. I. To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques. II. To understand the high-performance programs designed to strengthen the practical expertise. <p>COURSE LEARNING OUTCOMES (CLOs): The students should enable to:</p> <ol style="list-style-type: none"> 1. Analyze a given problem and develop an algorithm to solve the problem. 2. Describe the fundamental programming constructs and articulate how they are used to develop a program. 3. Gain knowledge to identify appropriate Python language constructs to write basic programs. 4. Identify the right data representation formats based on the requirements of the problem. 5. Understand branching statements, loop statements and use them in problem solving. 6. Identify the right string function to write string programs. 7. Learn data types and use them to solve Lists. 8. Identify the right string function to write string programs in python. 9. Distinguish Create, run and manipulate Python Programs using core data structures like Lists, multidimensional lists. 10. Understand the concept of class and objects. 11. Differentiate call by Object and call by Object reference parameter passing mechanisms and constructor. 12. Demonstrate the implementation of inheritance. 13. Use polymorphism to process objects depending on their class. 14. Understand overriding magic methods. 15. Understand the concept of event-driven programming. 								
LIST OF EXPERIMENTS								
Week-1	BASICS OF PYTHON							
<p>Write Python programs for the following:</p> <ol style="list-style-type: none"> a. Purposefully raise Indentation Error and Correct it. b. Compute distance between two points taking input from the user (Pythagorean Theorem). c. To takes numbers as command line arguments and print its sum. 								
Week-2	CONTROL FLOW							
<p>Write Python programs for implementing the following:</p> <ol style="list-style-type: none"> a. Checking whether the given number is even number or not. b. Finding the factorial of a number. c. Print the prime numbers below 100. 								

Week-3	STRINGS
Write Python programs for implementing the following: <ul style="list-style-type: none"> a. Count the numbers of characters in the string and store them in a dictionary data structure b. Using split and joins methods in the string and trace a birthday with a dictionary data structure. 	
Week-4	LIST
Write Python programs to for the following: <ul style="list-style-type: none"> a. Finding mean, median, mode for the given set of numbers in a list. b. Function dups to find all duplicates in the list. 	
Week-5	MULTI DIMENSIONAL LIST
Write Python programs for the following: <ul style="list-style-type: none"> a. Addition of two square matrices. b. Multiplication of two matrices. 	
Week-6	CLASS
Write Python programs to implement the following: <ul style="list-style-type: none"> a. Find the validity of a string of parentheses, '(', ')', '{', '}', '[' and ']'. These brackets must be close in the correct order, for example "()" and "(){ }" are valid but "D", "{(D]}" and "{ { { " are invalid. b. Get all possible unique subsets from a set of distinct integers. 	
Week-7	METHODS
Write Python programs to do the following <ul style="list-style-type: none"> a. Create a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle. b. Create a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle. 	
Week-8	CONSTRUCTORS
Write Python program to implement constructors.	
Week-9	INHERITANCE
Write Python program to implement inheritance.	
Week-10	POLYMORPHISM
Write Python program to implement Polymorphism.	
Week-11	OVERRIDING MAGIC METHODS
Write Python program to override Magic Methods.	
Week-12	EVENT-DRIVEN PROGRAMMING
Write Python program to create a simple calculator, where the user will enter a number in a text field, and either add it to or subtract it from a running total, which we will display. We will also allow the user to reset the total.	

Text Books:

1. R Nageswara Rao, "Core Python Programming", Dreamtech press, 2017 Edition.
2. Dusty Philips, "Python 3 Object Oriented Programming", PACKT Publishing, 2 nd Edition, 2015.

Reference Books:

1. Rance D. Necaie, "Object-Oriented Programming in Python Documentation Release 1", University of Cape Town and individual contributors, 2017.

Web References:

1. <https://www.w3resource.com/python-exercises/class-exercises/>
2. <https://www.rithmschool.com/courses/python-fundamentals-part-2/python-object-oriented-programming-exercises>