

ADVANCED PYTHON PROGRAMMING LABORATORY

III Semester: CSE / IT / CSIT

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P		C	CIA	SEE
ACSC11	Core	1	0	2	2	40	60	100
Contact Classes: 12	Tutorial Classes: Nil	Practical Classes: 24			Total Classes:36			

Prerequisite: Python Programming Laboratory

I. COURSE OVERVIEW:

Python is a general purpose and high level programming language. Python programming language was developed to provide a way to develop code that's easy to create and understand. While Python contains the same basic structures like other languages, it also offers unique functionality that makes life as a programmer easier. This course will provide the fundamentals of computer programming in Python with topics that include basic decisions and loops, object-oriented programming and graphical user interfaces.

II. COURSE OBJECTIVES:

The students will try to learn:

- I. The programming skills in advanced Python.
- II. The Object-oriented programming skills in Python.
- III. The skill of to design graphical-user interfaces (GUI) in Python.
- IV. An ability to write database applications in Python.

LIST OF EXPERIMENTS

Week – 1: DICTIONARY

- a. Write a program to count the numbers of characters in the string and store them in a dictionary data structure
- b. Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.
- c. Write a program combine_lists that combines these lists into a dictionary.
- d. Write a Python program to find shortest list of values with the keys in a given dictionary.

Week – 2: NESTED LISTS

- a. Write a program to read a 3 X 3 matrix and find the transpose.
- b. Write a program to perform addition, subtraction of two 3 X 3 matrices.
- c. Write a program to perform multiplication of two 3 X 3 matrices.
- d. Write a program to check whether two given 3 X 3 matrices are identical or not.

Week – 3: USER DEFINED FUNCTIONS

- a) Write a function ball_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.
Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius. If (distance between two balls centers) \leq (sum of their radii) then (they are colliding)
- b) Write a function to find mean, median, mode for the given set of numbers in a list.
- c) Write a function nearly_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.
(HINT:- Nearly equal string : Python – Jython, Perl – Pearl)
- d) The mathematician Srinivasa Ramanujan found an infinite series that can be used to generate a numerical approximation of $1/\pi$:
Write a function called estimate_pi that uses this formula to compute and return an estimate of π . It should use a while loop to compute terms of the summation until the last term is smaller than $1e-15$ (which is Python notation for 10^{-15}).
You can check the result by comparing it to math.pi.

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

Week – 4: MODULES

- a. Install packages requests, flask and explore using (pip)
- b. Write a Python program that imports requests and fetch content from wiki page.
- c. Write a Python program to generate a series of unique random numbers by using random module
- d. Write a Python program to find the substrings within a string using re module.

Week – 5: DATE AND TIME

- a. Demonstrate Basic date and time classes, Different time formats, Converting between formats, Formatting dates and times, Parsing date/time information.
- b. Write a script that reads the current time and converts it to a time of day in hours, minutes, and seconds, plus the number of days since the epoch.
- c. Design a Python script to determine the difference in date for given two dates in YYYY:MM:DD format(0 <= YYYY <= 9999, 1 <= MM <= 12, 1 <= DD <= 31) following the leap year rules.
- d. Design a Python Script to determine the time difference between two given times in HH:MM:SS format.(0 <= HH <= 23, 0 <= MM <= 59, 0 <= SS <= 59).

Week – 6: CLASS AND OBJECTS

- a. Create a class ATM and define ATM operations to create account, deposit, check_balance, withdraw and delete account. Use constructor to initialize members.
- b. Make a class Employee with a name and salary. Make a class Manager inherit from Employee. Add an instance variable, named department. Write a method that prints manager's name, department and salary. Make a class Executive inherit from Manager. Write a method that prints the string "Executive" followed by the information stored in the Manager super class object.
- c. A hospital wants to create a database regarding its indoor patients. The information to store include a) Name of the patient b) Date of admission c) Disease d) Date of discharge. Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the database. Create a derived class to store the age of the patients. List the information about to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age).

Week – 7: FILE HANDLING

- a. Write a python program to read the file contents and do the following operations
 - i) Print each word of a file in reverse order.
 - ii) Print each line of a file in reverse order.
Sample Input: Python Programming
Sample Output: Programming Python
 - iii) Display the content of a without whitespaces.
- b. Demonstrate Serializing Data with XML and JSON:
 - i) Working with XML modules in Python
 - ii) Start with Element Tree
 - iii) Parsing XML
 - iv) Updating an XML tree
 - v) Creating a new document
 - vi) YAML, other formats as time permits
 - vii) Reading, Writing JSON
 - viii) Reading/writing CSV fileS
- c. Write a Python program to store N student's records containing name, roll number and branch. Print the given branch student's details only.

Week – 8: EXCEPTION HANDLING

- Read two numbers n_1 and n_2 . Write a function to compute n_1/n_2 and use try/except to catch the exceptions.
- Write a Python program to detect and handle the exception while solving the quadratic equation.
- Write a Python program to handle the run time errors while doing file handling operation.
- Write a Python program to create and raise user defined exception.

Week – 9: MULTITHREADING AND TESTING

- Write a python program to create two threads to keep a count of number of even numbers entered by the user.
- Write a JAVA program that creates threads by extending Thread class .First thread display “Good Morning” every 1 sec, the second thread displays “Hello” every 2 seconds and the third display “Welcome” every 3 seconds.
- Write a test-case to check the function even_numbers which return True on passing a list of all even numbers.
- Write a test-case to check the function reverse_string which returns the reversed string.

Week – 10: NUMPY

- Using Numpy, write a basic array of operations on single array to add x to each element of array and subtract y from each element of array.
- Using Numpy, write a program to add, subtract and multiply two matrices.
- Write a Python program to do the following operations:

Library: NumPy

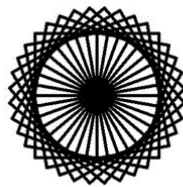
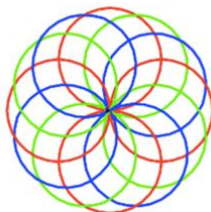
- Create multi-dimensional arrays and find its shape and dimension
- Create a matrix full of zeros and ones
- Reshape and flatten data in the array
- Append data vertically and horizontally
- Apply indexing and slicing on array
- Use statistical functions on array - Min, Max, Mean, Median and Standard Deviation
- Dot and matrix product of two arrays
- Compute the Eigen values of a matrix
- Solve a linear matrix equation such as $3 * x_0 + x_1 = 9$, $x_0 + 2 * x_1 = 8$
- Compute the multiplicative inverse of a matrix
- Compute the rank of a matrix
- Compute the determinant of an array

Week – 11: GUI

- Design a GUI based calculator to perform arithmetic operations like addition, subtraction, multiplication and division. (Hint: Expression Calculator using tk)
- Design a GUI based application to convert temperature from Celsius to Fahrenheit.
- Write a python program to perform various database operations (create, insert, delete, update).

Week – 12: GRAPHICS

- Consider turtle object. Write functions to draw triangle, rectangle, polygon, circle and sphere. Use object oriented approach
- Design a Python program using the Turtle graphics library to construct a turtle bar chart representing the grades obtained by N students read from a file categorizing them into distinction, first class, second class, third class and failed.
- Write a python figures using turtle



program to implement the following

III. Reference Books:

1. Michael H Goldwasser, David Letscher, “Object Oriented Programming in Python”, Prentice Hall, 1st Edition, 2007.
2. Yashavant Kanetkar, Aditya Kanetkar, “Let us Python, BPB publication, 1st Edition, 2019.
3. Ashok Kamthane, Amit Kamthane, “Programming and Problem solving with Python”, McGraw Hill Education (India) Private Limited, 2018.
4. Taneja Sheetal, Kumar Naveen, “Python Programming – A Modular Approach”, Pearson, 2017.
5. R Nageswara Rao, “Core Python Programming”, Dreamtech Press, 2017 Edition.
6. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, “How to Think Like a Computer Scientist: Learning with Python 3”, 3rd Edition, 2015.
7. Paul Barry, “Head First Python a Brain Friendly Guide”, O’Reilly, 2nd Edition, 2016.
8. Dainel Y.Chen “Pandas for Everyone Python Data Analysis” Pearson Education, 2019.

IV. Web References:

1. <https://realpython.com/python3-object-oriented-programming/>
2. <https://python.swaroopch.com/oop.html>
3. https://python-textbok.readthedocs.io/en/1.0/Object_Oriented_Programming.html
4. <https://www.programiz.com/python-programming/>
5. <https://www.geeksforgeeks.org/python-programming-language/>
6. <http://www.ict.ru.ac.za/Resources/cspw/thinkcspy3/thinkcspy3.pdf>