K. S. INSTITUTE OF TECHNOLOGY

An Autonomous Institution under VTU, Approved by AICTE

Department of Computer Science and Engineering

FIRST/SECOND SEMESTER SYLLABUS

Course: Python Programming Lab		Semester	I/II
Course Code	**25BPLL109A/209A**	CIE Marks	50
Teaching Hours/Week (L:T:P:S)	0:0:2:0	SEE Marks	50
Total Hours of Pedagogy	24	Total Marks	100
Credits	1	Exam Hours	3
Examination type (SEE)	Practical		

Course Objectives (Course Skill Set)

1. Learn the syntax and semantics of the Python programming language.
2. Illustrate the process of structuring the data using lists, tuples
3. Appraise the need for working with various documents like Excel, PDF, Word and Others.
4. Demonstrate the use of built-in functions to navigate the file system.
5. Implement the Object Oriented Programming concepts in Python.

Note:

1. The laboratory syllabus consists of PART-A. The maximum marks for the laboratory course are 100.
2. PART-A are considered for CIE and SEE.
3. Students have answer 1(one) question from PART-A. The questions set for SEE shall be from among the experiments under PART-A. It is evaluated for 100 marks out of the maximum 100 marks.

List of Experiments

1. a. Develop a python program to read 2 numbers from the keyboard and perform the basic arithmetic operations based on the choice. (1-Add, 2-Subtract, 3-Multiply, 4-Divide).

   b. Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not.
2. a. Develop a program to generate Fibonacci sequence of length (N). Read N from the console.

   b. Write a python program to create a list and perform the following operations
   - Inserting an element
   - Removing an element
   - Appending an element
   - Displaying the length of the list
   - Popping an element
   - Clearing the list
3. a. Read N numbers from the console and create a list. Develop a program to print mean, variance and standard deviation with suitable messages.

   b. Read a multi-digit number (as chars) from the console. Develop a program to print the frequency of each digit with a suitable message.

4. Develop a program to print 10 most frequently appearing words in a text file. [Hint: Use a dictionary with distinct words and their frequency of occurrences. Sort the dictionary in the reverse order of frequency and display the dictionary slice of the first 10 items.
5. Develop a program to read 6 subject marks from the keyboard for a student. Generate a report that displays the marks from the highest to the lowest score attained by the student. [Read the marks into a 1-Dimesional array and sort using the Bubble Sort technique].
6. Develop a program to sort the contents of a text file and write the sorted contents into a separate text file. [Hint: Use string methods strip(), len(), list methods sort(), append(), and file methods open(),
   readlines(), and write()].
7. Develop a function named DivExp which takes TWO parameters a, b, and returns a value c (c=a/b). Write a suitable assertion for a>0 in the function DivExp and raise an exception for when b=0. Develop a suitable program that reads two console values and calls the function DivExp.
8. Define a function that takes TWO objects representing complex numbers and returns a new complex number with the sum of two complex numbers. Define a suitable class 'Complex' to represent the complex number. Develop a program to read N (N >=2) complex numbers and to compute the addition of N complex numbers.
9. Text Analysis Tool: Build a tool that analyses a paragraph: frequency of each word, longest word, number of sentences, etc.
10. Develop Data Summary Generator: Read a CSV file (like COVID data or weather stats), convert to dictionary form, and allow the user to run summary queries: max, min, average by column.
11. Develop Student Grade Tracker: Accept multiple students' names and marks. Store them in a list of tuples or dictionaries. Display summary reports (average, topper, etc.).
12. Develop a program to display contents of a folder recursively (Directory) having sub-folders and files (name and type).

Course outcome (Course Skill Set)

At the end of the course, the student will be able to:

CO1: Demonstrate proficiency in handling loops and creation of functions.
CO2: Develop python programs using core data structure.
CO3: Make use of file operations and Python standard libraries for programming.
CO4: Apply concepts of Python modules and examine the OOP concepts for Application using python.
CO5: Illustrate the concepts of Object-Oriented Programming as used in Python.

Suggested Learning Resources: (Text Book/ Reference Book/ Manuals):

Text books:

1. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers- How to think like a computer scientist: learning with python 3. Green Tea Press, Wellesley, Massachusetts,2020 https://media.readthedocs.org/pdf/howtothink/latest/howtothink.pdf

Reference books / Manuals:

1. Al Sweigart," Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners",2nd Edition, No Starch Press, 2022. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/)
   Kyla McMullen, Elizabeth Matthews and June Jamrich Parsons, Programming with Python, Cengage, 2023.

Web links and Video Lectures (e-Resources):

https://www.learnbyexample.org/python/ https://www.learnpython.org/

https://pythontutor.com/visualize.html#mode=edit

Teaching-Learning Process (Innovative Delivery Methods):

The following are sample strategies that educators may adopt to enhance the effectiveness of the teaching-learning process and facilitate the achievement of course outcomes.

1. Engineering tool usage for the conduction of experiment
2. Demonstration through ICT tools
3. Use of virtual labs (https://www.vlab.co.in/)

Assessment Structure:

The assessment for each course is equally divided between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each component carrying **50% weightage** (i.e., 50 marks each).
The CIE marks awarded shall be based on the continuous evaluation of the laboratory report using a defined set of rubrics. Each experiment report can be evaluated for 30 marks. The laboratory test (duration 03 hours) at the end of the last week of the semester /after completion of all the experiments (whichever is early) shall be conducted for 50 marks and scaled down to 20 marks. For both CIE and SEE, the student is required to conduct one experiment each from Part A .

- To qualify and become eligible to appear for SEE, in the **CIE component**, a student must secure **a minimum of 40% of 50 marks**, i.e., **20 marks**.
- To pass the **SEE component**, a student must secure **a minimum of 35% of 50 marks**, i.e., **18 marks**.

A student is deemed to have **successfully completed the course** if the **combined total of CIE and SEE is at least 40 out of 100 marks**