RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

102902/CO400A: PYTHON FOR ENGINEERS
LAB CYCLE: S4 CSE

Department Vision

To become a Centre of Excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Educational Objectives(PEO)

PEO 1: Graduates shall have up-to-date knowledge in Computer Science & Engineering along with interdisciplinary and broad knowledge on mathematics, science, management and allied engineering to become computer professionals, scientists and researchers.

PEO 2: Graduates shall excel in analysing, designing and solving engineering problems and have life-long learning skills, to develop computer applications and systems, resulting in the betterment of the society.

PEO 3: Graduates shall nurture team spirit, ethics, social values, skills on communication and leadership, enabling them to become leaders, entrepreneurs and social reformers.

Programme Specific Outcomes (PSOs)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills:
The ability to identify, analyze and design solutions for complex engineering problems in

multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills
The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills:
The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes

CO1: Write, test and debug fundamental Python programs using decision and control statements. (Cognitive Knowledge : Understand)

CO2: Illustrate the use of Python Data Structures like lists, tuples and dictionaries (Cognitive Knowledge : Apply)

CO3: Write, test and debug fundamental Python programs using user defined functions, higher order and built-in functions.(Cognitive Knowledge : Apply)

CO4: Illustrate graphical user interface solutions using Python libraries. (Cognitive Knowledge : Apply)

CO5: Illustrate Object Oriented programs with exception handling. (Cognitive Knowledge : Apply)

EXPERIMENTS

DAY 1: Familiarization of data types and input- output statements

1. Write a program to take three inputs: a name, age, and city. Print the output in a sentence like: "Hello, [Name]! You are [Age] years old and live in [City]."

2. Write a Python script that:
    a. Declare a variable in the program.
    b. Determines and prints its data type and the variable. (do for all datatypes)

3. Write a Python script to find the area and perimeter of two different circles and display them.

4. Write a Python program that takes marks for 5 subjects as input, calculates the total, percentage, and displays the result in a formatted way.

5. Write a Python program to calculate simple interest and display.

DAY 2: Decision making, branching and looping statements

1. Write a program to implement a simple calculator and perform all arithmetic operations.
2. Write a program to find the largest of three numbers using the 'if' statement.
3. Write a program to print numbers from 1 to 30, skipping multiples of 3. Use 'continue' to skip the iteration when a number is divisible by 3.
4. Write a program that takes an integer as input and calculates the sum of its digits using a 'while' loop.
5. Write a program to take an integer input n and calculate the sum of squares of all numbers from 1 to n and display the result.
6. Write a program to take inputs for the coefficients a, b, and c of a quadratic equation ax^2 + bx + c = 0. Calculate and display the roots of the equation (real or complex).
7. Write a program to take an email address as input and check if it is valid. (Basic validation: must contain @ and .).

DAY 3: Python Data Structures – Strings, List, Tuples, Dictionaries

1. Write a Python program to take a string as input and:
    a. Print the length of the string
    b. Convert the string to uppercase and lowercase.
    c. Check if it is palindrome
    d. Replace all vowels with _
    e. Count the number of vowels

2. Write a Python program to create a list of 10 random numbers between 1 and 100:
    a. Find the Maximum , Minimum and average of the list.
    b. Reverse the list.
    c. Create a list of squares of all even numbers.

3. Write a Python program to create a tuple of 10 numbers:
    a. Calculate and print the sum and product of the elements
    b. Check if the specific number is in the tuple.
    c. Add two new elements to the list.

4. Write a Python program to create a dictionary with 5 key- value pairs:
    a. Print all the keys , values , items in the dictionary.
    b. Update the value of a specific key.
    c. Add a new value to the dictionary.

5. Write a program to ask the user to input numbers separated by spaces.
    a. Convert the input string into a list of integers.
    b. Calculate and print the sum of the list.

6. Write a Python program to check the strength of a password based on length, use of uppercase, lowercase, numbers, and special characters.

7. Write a Python program to input a string and count the number of vowels, consonants, and spaces.

DAY 4: Function & Function calls

1. Write a Python program using functions for finding factorial of a number.

2. Write a Python program using functions for finding fibonacci series upto n numbers.

3. Write a Python function that recommends a course based on a learner's current progress and rating from other learners. The system works as follows:
    a. A learner provides their current skill level and the category of interest.
    b. The function searches a list of courses and recommends a course with the highest rating that matches the learners input.
    c. If no matching course is found, suggest the learner to improve their skills before enrollment.

DAY 5: Object Oriented Programming using Python

1. Write a Python program to create a class representing a bank. Include methods for managing customer accounts and transactions.

2. Write a Python program to create a class representing a shopping cart. Include methods for adding and removing items, and calculating the total price.

3. Develop a system to conduct online tests.
Requirements:

- **Classes**: Question, Quiz, Student.
- **Features**:
  - Allow teachers to add questions.
  - Conduct a quiz for students.
  - Evaluate and display scores.
  - Support multiple-choice and short-answer questions.

DAY 6: GUI Implementation

1. Write a python program to develop a GUI based english dictionary. A file or database stores English words and their meanings. On pressing a button, retrieve the meaning of a given word. The application must also help the users count the numbers of words starting with a given character. The application must also accept a number n and must display the first word of length n. Application can be accessed only if the user is authenticated via username and password.

DAY 7: Data Analysis

1. Analyse the titanic dataset using the below link and do the following questions. https://bit.ly/3GnTf3t

   a. Display the entire data set
   b. Display the number of males and females
   c. Display the details of females alone
   d. Count and display the number of people survived and display the details

DAY 8: Files and Operations

You are working as a data analyst for a retail company. The company wants to understand the sales performance across different regions and product categories. You are provided with a CSV file `sales_data.csv` containing the following columns:

- `OrderID`: Unique identifier for each order
- `Product`: Name of the product sold
- `Category`: Category of the product (e.g., *Electronics, Furniture, Clothing*)
- `Region`: Region where the sale was made (e.g., *North, South, East, West*)
- `Quantity`: Number of units sold
- `Price`: Price per unit
- `OrderDate`: Date of the order

Your task is to perform the following analysis using pandas:

1. **Load the Data:** Read the `sales_data.csv` file into a pandas DataFrame.
2. **Data Cleaning:** Handle any missing or erroneous data. If `Quantity` or `Price` is missing, fill it with the median value of the respective columns.
3. **Sales Analysis:**
   - Calculate the total sales amount for each order (`Total Sales = Quantity * Price`).
   - Determine the total sales for each region.
   - Identify the top 3 product categories with the highest sales.
4. **Time-Series Analysis:**
   - Group the sales data by month and calculate the total monthly sales.
5. **Visualization:** (Optional)
   - Plot a bar chart showing total sales per region.
   - Plot a line chart of monthly sales trends.
6. **Export Results:** Save the analysis results into a new CSV file `sales_analysis.csv`.

DAY 9: Data Visualization

1. Read and explore the given dataset (Using Panda)
   Create Visualizations
   a. Bar chart
   b. Line Plot
   c. Pie chart
   d. Heatmap
   Add interactive visualizations using plotly

DAY 10: Image Processing Functions

1. Explore image processing by performing various operations like resizing, rotating, filtering, edge detection, and histogram analysis using Python libraries like OpenCV and PIL (Pillow).
    a. Read and Display an Image.
    b. Resize and Rotate the Image.
    c. Apply Filters (Grayscale, Blur, Sharpen).
    d. Perform Edge Detection.
    e. Histogram Analysis of Pixel Intensities.

Mr. Sandy Joseph (S4 CS A)

Mr. Paul Augustine (S4 CS B)

Ms. Jisha Mary Jose (S4 CS C)

Dr. Preetha K G

(HOD, CSE)