**SCRIPTING LANGUAGES LAB**

|  |  |
| --- | --- |
| **Course Code: ISL58** | **Credits: 0:0:2:0** |
| **Course Coordinator: Dr.Mydhili K Nair** | **Contact Hours: 28P** |

|  |  |
| --- | --- |
| **COURSE CONTENTS** | |
| **PART A** | |
| * **Python Introduction** - Simple Problems in Python based on:   **1)** Selection Constructs **2)** Looping Constructs   * **Using Python -** Simple Problems in Python based on:   **1)** Lists 2**)** Tuples 3**)** Dictionary   * **Combining Python Basics and its Data Structures -** Problems in Python based on:   Selection, Looping with Lists, Tuples, Dictionaries   * **Using Python -** Problems in Python based on:   **1)** Functions **2)** Class   * **Python for Data Science** - Data Manipulation on IRIS Data Set * **Python for Data Science** -Data Visualization based on IRIS Data Set * **HTML Basics Introduction** - Creating Static Pages with different HTML Components * **HTML Basics Introduction** - Creating Static Pages with hyperlinks and images * **Javascript Introduction** - Basics of client side scripting * **Using Javascript** - HTML along with Javascript on client side * **Combining HTML & Javascript** - Programs with Client side event handling scenarios * **Using Javascript** - Loading JSON data dynamically on the client side. | |
| **PART B** | |
| 1. **Introduction to Python:** Write Python programs to do the following: 2. Read a list of elements. Create a new list having all the elements minus the duplicates (Use functions). Use one-line comprehensions of create a new list of even numbers. Create another list reversing the elements. 3. Write a python program to count the frequency of words in a given file. 4. Read a list of numbers. Uses a recursive function to find the maximum of ‘n’ numbers. | |
| 1. **Introduction to Python Functions:** Write a temperature converter python program, which is menu driven. Each such conversion logic should be defined in separate functions. The program should call the respective function based on the user’s requirement. The program should run as long as the user wishes so. Provide an option to view the conversions stored as list of tuples with attributes - from unit value, to unit value sorted by the user’s choice (from-value or to-value). | |
| 1. **Python Classes:** Write a python class to reverse a sentence (initialized via constructor) word by word. Example: “I am here” should be reversed as “here am I”. Create instances of this class for each of the three strings input by the user and display the reversed string for each, in descending order of number of vowels in the string. | |
| 1. **Python for Data Science:** Load *Titanic* *Dataset* into one of the data structures (*NumPy* or *Pandas*). Perform data pre-processing on this dataset. Create dataframes, perform computations and visualize the results appropriately. | |
| 1. **Python File Handling & List Comprehension**: Write a python program to read contents of a file (filename as argument) and store number of occurrences of each word in a dictionary. Display the top 10 words with most number of occurrences in descending order. Store the length of each of these words in a list and display the list. Write a one-line reduce function to get the average length and one-line list comprehension to display squares of all odd numbers and display both. | |
| 1. **Python for Data Science:** Download the *Weather Dataset* from any online repository. 2. Change the Column Attributes in Data-frame to read as specified. **E.g**.  |  |  |  |  | | --- | --- | --- | --- | | **Old Attribute** | **New Attribute** | **Old Attribute** | **New Attribute** | | Mean Sea Level PressureIn | mean\_pressure | Max VisibilityMiles | max\_visibilty |  1. Use the inbuilt functions to evaluate and analyze the weather data set. 2. Perform suitable data visualizations. | |
| 1. **JavaScript – Client Side Validation:** Design any two case studies that creates and validates a HTML form at the client side using Javascript. Case Study examples could be Grade Calculation from student marks entered, Bakery Menu & Price calculation of items bought etc. | |
| 1. **Introduction to JavaScript (JS):** Dynamically loading JSON data - Implement a HTML+JS application that has a JSON Array with details of different kinds of data. Example: Model, Name, Price, Year. Display details of each vehicle dynamically by only showing details of the vehicle that the user has selected (via mouse-over). | |
| 1. **Python and JavaScript - Form Validation:** Design any case study that creates and validates a HTML form. On the server side, use Python to display the details entered by the user at the client side along with appropriate validation messages. Use Display appropriate messages if error occurs, and redirect to the another HTML page if successful. | |
| 1. **Python and JavaScript - ATM Application:** Design a HTML form that displays user’s current balance, an input field to enter amount and buttons to withdraw or deposit money. Validate the form such that   **i)** Negative amount cannot be entered and Users cannot withdraw more than 5000 at one time  **ii)** Users cannot withdraw amount greater than their balance and cannot deposit more than 10000 at one time. Also users can perform at most 5 transactions. Update the balance accordingly and ensure relevant data is not lost on closing the browser. | |
| 1. **Python and JavaScript - Shopping Cart Application:** Design a simple Shopping Cart application which allows users to add items to their cart from a list of products. Allow users to view their cart (items and quantities of each). Ensure that items in the cart persist even after closing the application. On selecting buy, print out a bill of items in the cart. Perform any necessary validation. Demonstrate data persistence even after the browser is closed. | |
| 1. **Python for Data Science -** Data Science and Machine Learning on *Boston Housing dataset* - Download Boston Housing dataset from online archives. Load the dataset into one of the data structures (*NumPy* or *Pandas*). Visualize the dataset (Features vs price plot) using *matplotlib*. Apply Linear Regression (Source to learn:*scikit*) on the loaded dataset to predict prices of new data. | |
| **Course Outcomes:** | |
| **1.** | Use internal and external Python libraries, data structures, functions inherent to Python in-order to handle data and use JavaScript to develop command line applications that create child processes to execute other programs and capture their output.(PO-1,2, 5, 6, 10,12) (PSO-1, 2, 3) |
| **2.** | Apply Python as a scripting language to analyze huge datasets, apply data science related statistics on  datasets (PO-1, 2, 5, 6, 10, 12) (PSO-1, 2, 3) |
| **3.** | Design and develop a simple web application with client-side JavaScript, server-side Python, using Flask – a micro-framework and develop JavaScript applications that can manipulate HTML pages dynamically, in response to user driven events and do client-side form validations. (PO-1, 2, 5, 6, 10, 12) (PSO-1, 2, 3) |