Ramaiah Institute of Technology
Department of Information Science and Engineering

Instructions for Scripting Languages Lab **Course Code: ISL57**

Test Date	Syllabus
Dec 26, Dec 31	Part A 1 to 5 Part B 1 to 5
Jan 1, Jan 2	Part A 6 to 10 Part B 6 to 10

- Refer the program list in Drive
- Record submission: 30th Dec within 3 pm; Record books will be evaluated by faculty concerned. Records must be submitted in Ramaiah Record format. Marks will be deducted otherwise.
- No retests permitted.
- In a batch only 50 students will be permitted. (First Come First Serve)
- Class Representatives are asked to collect the names of those students appearing for tests on different dates by tonight.
- Online Viva Test : 29th Dec (9.30 AM to 10.00 AM). URL will be shared with all of you.
- Grading Scheme for Evaluation:
    - Both programs executed : Part A(8 Marks), Part B (12 Marks)
    - Change of Question (either/both Part A, Part B): - 4 Marks
        - After change of question, the student will be asked to choose the appropriate question he or she wants to change.
    - Did not start typing either Part A or Part B program:
        - Can give 2 Marks if the student is able to answer the viva questions about the program(what it does, logic etc.) Else zero.
    - Partially typed the program at least main logic:
        - Part A: 4 Marks can be given
        - Part B : 6 Marks can be given
    - Compilation Errors:
        - Part A: 4 Marks can be given
        - Part B : 6 Marks can be given
    - Partial Output:
        - Part A: 5 Marks can be given
        - Part B : 7 Marks can be given

SCRIPTING LANGUAGES LABORATORY	
	PART A
	Refer the detailed PART A Questions given below. The solutions are uploaded in the drive folder shared by Evangeline Mam.
	PART B
	1. Introduction to Python: Write Python programs to do the following: **a)** Read a list of elements. Create a new list having all the elements minus the duplicates (Use functions). Use one-line comprehensions to create a new list of even numbers. Create another list reversing the elements. **b)** Write a python program to count the frequency of words in a given file. **c)** Read a list of numbers. Uses a recursive function to find the maximum of 'n' numbers.
	2. 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 a list of tuples with attributes - from unit value, to unit value sorted by the user's choice (from-value or to-value).
	3. 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 the number of vowels in the string.
	4. 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.
	5. Python File Handling & List Comprehension: Write a python program to read contents of a file (filename as argument) and store the number of occurrences of each word in a dictionary. Display the top 10 words with the 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.
	6. Python for Data Science: Load *'Black Friday 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.
	7. 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.
	8. 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).

	9. Python for Data Science - Load *'Iris 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.
	10. Python for Data Science - Load *'Student Performance 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.

PART – A Questions

1	Write Python code to do the following: i. Create list with inputs from user ii. Determine minimum and maximum elements in the list iii. Insert new element into the list iv. Delete an element from the list v. Determine if an element is present in the list.
2	Write a python program to create a class 'Rectangle'. This class should include a constructor to initialize the dimensions. Include a function in the class to compute the area of the rectangle. Create objects of the class and print area.
3	(i) Create a dictionary that contains the atomic element symbol and its name. (ii)Add a unique & duplicate element into this dictionary by interacting with the user. Observe the output and justify it. (iii) Display the number of atomic elements in this dictionary (iv) Ask the user to enter an element to search in the dictionary. Display appropriate results. Rewrite this program so that these operations are inside a function called 'AtomicDictionary'. Create another python file called "Atomic.py" and execute this function in it.
4	Create a Python class called 'Student' having 'name', 'age' as attribute along with a list having the marks obtained for three subjects. (i) Create a constructor to initialize two objects of this class. (ii) Create a member function called 'display' printing the details of a specific object (iii) Ask users to enter the values for an object through an 'accept' member function. (iv) Display these details.
5	Create a list of 6 numbers. Use 'list-comprehension' to create a new list where each element in the original list is multiplied by 3. Use 'lambda' and 'reduce' function to find the sum of the elements of the original list as well as the new list.

6	Create a form to get customer feedback. Following features to be present in the form 1. Text box to get customer name and display on the same page. 2. Radio button to get gender and display on the same page. 3. Text area to get feedback and display on the same page.
7	Create a HTML form to accept a number. Include a button to check whether the input is divisible by 3 or 7. Display the result. Ensure the data entered in the text box is a number only. Use Javascript for this client side scripting. Display appropriate error texts. Also show the execution when the javascript code is in a different file.
8	Create a HTML form to simulate a simple calculator. The arithmetic operations are to be displayed as radio buttons. Enter two numbers in two text boxes. Show the result. Handle the error case of 'divide by zero'. Also show the execution of this code with the javascript code as a ".js" file outside the HTML
9	Create a HTML form which contains one text area. Enter a sentence. Find the length of the longest word of that sentence using javascript. Also show the execution of this code with the javascript code as a ".js" file outside the HTML.
10	Create a HTML to perform Currency conversion from ● USD and Rupee and vice versa ● Euro to Rupee and vice versa