

CSAI 1101	Python Programming Lab	L	T	P	C
Version 1.0		0	0	2	1
Pre-requisites/Exposure	Basic knowledge of computer system and elementary mathematics				
Co-requisites	--				

Course Objectives

1. To understand the fundamental of python programming language.
2. To write python scripting elements such as variables and flow control structures.
3. To use python library packages for solving domain problems.

Course Outcomes

CO 1	To know the basic python programming concepts, data structures & regular expressions.
CO 2	Discuss file handling operations and understand OOPS concepts using python.
CO 3	Develop modules and implement web development framework.
CO 4	Discuss the role of python in advance technology.

Catalog Description

This course introduces the basic concepts of procedural and object-oriented programming using python programming language. This course also provides practical knowledge and hands-on experience in designing and implementing data structures. Activities covered include introduction to python programming language, datatypes, operators, loop structures, decision-making statements, fundamental data structures, functions, Classes and Objects, Constructor, File Handling, Exception Handling and Numpy module.

List of Experiments

Lab Exercise	Contents
Experiment No 1	Introduction
Experiment No 2 & 3	Basic Data Types
Experiment No 4	Strings
Experiment No 5 & 6	Functions
Experiment No 7 & 8	Files
Experiment No 9	Exceptions

Experiment No 10 & 11	OOPS
Experiment No 12 & 13	GUI using tkinter library
Experiment No 14	Usage of numpy
Experiment No 15	Usage of pandas

Experiment 1 - Introduction

Q1. Given an integer n perform the following conditional actions:

- If n is odd, print Weird
- If n is even and in the inclusive range of 2 to 5, print Not Weird
- If n is even and in the inclusive range of 6 to 20, print Weird
- If n is even and greater than 20, print Not Weird

Q2. WAP to read an integer 'n' from STDIN. For all non-negative integers $i < n$, print i^2 on a separate line.

Sample input

3

Sample Output

1

2

4

Example

The list of non-negative integers that are less than 3 is [0, 1, 2]. The squares of each number is given below:

0

1

4

Q3. WAP to read an integer from STDIN. Without using any string methods, print the following on a single line:

123...n

Note that ... represents the consecutive values in between.

Example

n=5

Output- 12345

Experiment 2 & 3– Basic Data Types

Q1. WAP to read the record of n students in a dictionary containing key/value pairs of name: [marks]. Print the average of the marks obtained by the particular student correct to 2 decimal places.

Input Format

The first line contains the integer n , the number of student's records. The next n lines contain the names and marks obtained by a student, each value separated by a space.

Sample Input

3

Krishna 67 68 69
Arjun 70 98 63
Malika 52 56 60

Sample Output

56.00

Q2. WAP to input a list of scores for N students in a list data type. Find the score of the runner-up and print the output.

Sample Input

N = 5

Scores= 2 3 6 6 5

Sample output

5

Note:

Given list is [2, 3, 6, 6, 5]. The maximum score is 6, second maximum is 5. Hence, we print 5 as the runner-up score.

Q3. Rupal has a huge collection of country stamps. She decided to count the total number of distinct country stamps in her collection. She asked for your help. You pick the stamps one by one from a stack of country stamps. Find the total number of distinct country stamps using a suitable data type.

Note: Apply your knowledge of the `.add()` operation from set to help your friend Rupal.

Experiment 4- Strings

Q1. WAP to enter a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left.

Sample Input

ABCD CDC

CDC

Sample Output

2

Q2. WAP to input the first name, middle and last name of a person. Your task is to print the initials of the first and middle name separated by a dot (.)

The last name should be followed by a dot and a space where the first letter is capital.

Sample Input

Mohandas KaramChand Gandhi

Sample Output

M.K. Gandhi

Q3. Given a string containing both upper and lower case alphabets. Write a Python program to count the number of occurrences of each alphabet (case insensitive) and display the same.

Sample Input

ABaBCbGc

Sample Output

2A

3B

2C
1G

Experiment 5 & 6- Functions

Q1. Using functions, re-write and execute Python program to:

1. Add natural numbers upto n where n is taken as an input from user.
2. Print Fibonacci series till nth term (Take input from user).

Q2. At an airport, a traveler is allowed entry into the flight only if he clears the following checks:

1. Baggage Check
2. Immigration Check
3. Security Check

The logic for the check methods are given below:

check_baggage (baggage_weight)

- returns True if baggage_weight is greater than or equal to 0 and less than or equal to 40. Otherwise returns False.

check_immigration (expiry_year)

- returns True if expiry_year is greater than or equal to 2030 and less than or equal to 2050. Otherwise returns False.

check_security(noc_status)

- returns True if noc_status is 'valid' or 'VALID', for all other values return False.

traveler()

- Initialize the traveler Id and traveler name and invoke the functions check_baggage(), check_immigration() and check_security() by passing required arguments.

Refer the table below for values of arguments.

Variable	Value
traveler_id	1001
traveler_name	Jim
baggage_weight	35
expiry_year	2019
noc_status	VALID

- If all values of check_baggage(), check_immigration() and check_security() are true,
 - display traveler_id and traveler_name
 - display "Allow Traveler to fly!"Otherwise,
 - display traveler_id and traveler_name
 - display "Detain Traveler for Re-checking!"

Invoke the traveler() function. Modify the values of different variables in traveler() function and observe the output.

Q3. Write a Python program to find the maximum and minimum values in a given list of tuples using lambda function.

Original list with tuples:

[('V', 62), ('VI', 68), ('VII', 72), ('VIII', 70), ('IX', 74), ('X', 65)]

Output-

Maximum and minimum values of the said list of tuples:

(74, 62)

Experiment 7 & 8 - Files

Q1. Write a Python program to:

1. read a file.
2. add backslash (\) before every double quote in the file contents.
3. write it to another file in the same folder.
4. print the contents of both the files.

For example:

If the first file is 'TestFile1.txt' with text as:

Jack said, "Hello Pune".

The output of the file 'TestFile2.txt' should be:

Jack said,\"Hello Pune\".

Q2. Consider a file 'rhyme.txt' in D Drive with following text:

```
Jingle bells jingle bells
Jingle all the way
Oh what fun it is to ride
In a one horse open sleigh
Jingle bells jingle bells
Jingle all the way
```

Write a Python program to count the words in the file using a dictionary (use space as a delimiter). Find unique words and the count of their occurrences (ignoring case). Write the output in another file "words.txt" at the same location.

Q3. Assume a file city.txt with details of 5 cities in given format (cityname population(in lakhs) area(in sq KM)):

Example:

Dehradun 5.78 308.20

Delhi 190 1484

.....

Open file city.txt and read to:

- a. Display details of all cities
- b. Display city names with population more than 10Lakhs
- c. Display sum of areas of all cities

Experiment 9 -Exceptions

Q1. Input two values from user where the first line contains N, the number of test cases. The next N lines contain the space separated values of a and b. Perform integer division and print a/b. Print the error code in the case of *ZeroDivisionError* or *ValueError*.

Sample input

1 0
2 \$
3 1

Sample Output

Error Code: integer division or modulo by zero

Error Code: invalid literal for int() with base 10: '\$'

3

Q2. Assume the following Python code-

Rewrite the code to handle the exceptions raised. Print appropriate error messages wherever applicable.

```
mylist = [1,2,3,"4",5]

sum = 0

for i in mylist:

    sum = sum + i

print(sum)

print(mylist[5])
```

Q3. You have already created a Python program to implement the following in file handling section:

1. read a file.
2. add backslash (\) before every double quote in the file contents.
3. write it to another file in the same folder.
4. print the contents of both the files.

For example:

If the first file is 'TestFile1.txt' with text as:

Jack said, "Hello Pune".

The output of the file 'TestFile2.txt' should be:

Jack said,\"Hello Pune\".

Modify your code to implement Exception handling. Print appropriate error messages wherever applicable.

Experiment 10 & 11- OOPS

Q1.

- Create a class Employee with following properties
 - First Name
 - Last Name
 - Pay
 - Email : should be automatically generated as
 - Firstname + '.' + Lastname + "@company.com"
- Test the code with following information of an Employee:
 - First name is : **Mohandas**
 - Last name is : **Gandhi**
 - Pay is : **50000**

Employee
Properties: First Name Last Name Pay Email

Q2. Perform the following instructions:

- a) Create a Vehicle class with max_speed and mileage as instance attributes. Additionally, create a method named seating_capacity() using the below syntax:

```
def seating_capacity(self, capacity):  
    return f"The seating capacity of a {self.name} is {capacity} passengers"
```

- b) Create child class 'Bus' that will inherit all of the variables and methods of the Vehicle class. Set the seating capacity of the bus to 50 using super().
- c) Create a Bus object that will inherit all of the variables and methods of the Vehicle class and display it.
- d) Define a class attribute "color" with a default value white. I.e., Every Vehicle should be white.

Q3.

Q: List the risk associated with the implementation of Account class. Suggest a solution.

```
class Account:  
    def __init__(self, initial_amount):  
        self.balance = initial_amount  
    def withdraw(self, amount):  
        self.balance = self.balance - amount  
    def deposit(self, amount):  
        self.balance = self.balance + amount  
  
ac = Account(1000)  
ac.balance = 2000 #stmt1  
ac.balance = -1000 #stmt2  
print(ac.balance) #stmt3
```

Experiment 12 & 13- GUI using tkinter library

Q1.

- a) import Tkinter package and create a window and set its title
- b) set the default window size using geometry function
- c) Create a label with "Hello" text in it and set its position on the form.
- d) Add a button to the window with "CLICK ME" written on it.
- e) change the foreground and background color for the button created above
- f) Create a function that will be executed when the button is clicked and print "Button was clicked" on clicking the button

Q2. This is the continuation of Question1, add the given below features in the above program:

- a) Take **user name** as input using the Tkinter Entry class
- b) Print the entered text (username) on clicking the button.
- c) Create three RadioButtons as displayed below

d) Print the currently selected radio button or the radio button value.

Q3. Write a program to accept following details from a student using GUI

1. Name of the student (using Textbox)
2. Gender (Using radio button)
3. Qualification (Using List)
4. Marks of three subjects (using Textbox)

Compute the percentage of the student and display it in a textbox.

Experiment 14 - Numpy

Q1.

- a. Convert numbers = [1, 2.0, 3] to numpy array and convert all elements to string type.
- b. Create a 2 D array through list and set dtype as int32
- c. Find the rows and columns of the 2d array created in part b
- d. Print 10 random numbers between 1 and 100.

Q2.

- a) Write a NumPy program to get help on the add function
- b) Write a NumPy program to test whether none of the elements of a given array is zero
- c) Write a NumPy program to test whether any of the elements of a given array is non-zero
- d) Write a NumPy program to generate an array of 15 random numbers from a standard normal distribution

Experiment 15 - pandas

Q1. Refer the given excel file and perform various operations using pandas library:

0	GOOGL	27.82	87	845	larry page
1	WMT	4.61	484	65	n.a.
2	MSFT	-1	85	64	bill gates
3	RIL	not available	50	1023	mukesh ambani
4	TATA	5.6	-1	n.a.	ratan tata

- a. Read the above excel file in python.
- b. How do I write this file to a new file "new.csv"?
- c. Include column names in this file. Use 'ticker', 'eps', 'revenue', 'price', 'people' as column names.
- d. Convert all not available or n.a. values to NAN and also convert negative revenues to NAN because revenues can never be negative.
- e. Fill NAN values using a suitable approach.
- f. Write a function to change n.a value appearing in WMT to Sam Walton

Text Books

1. Python Programming by IBM

Reference Books

2. Python Programming by “Reema Thareja”
3. Learning Python – Mark Lutz, O’reilly publication

Continuous Evaluation- There will be continuous evaluation for all practical subjects of SCS during the semester. The performance of a student in a Practical subject will be evaluated as per process given below:

1. Components of evaluation

- a. Viva voce / Quiz (50%) + Performance & Records (50%).
- b. Lab performance and record evaluation shall be a continuous process throughout the semester.
- c. Minimum three Viva voce/ Quiz based on practical sessions shall be conducted during the semester.

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Outcomes(PSOs):

Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3
CO1									1			1	3		
CO2									1			1	1		
CO3													2	2	
CO4				1									2		

1 = Weak

2 = Moderate

3 = Strong