Assignment 1A - OOPS & Basics of Python

Set – 1:

1. Accept the cost price and selling price of an item as a user input. Write a program to determine whether the seller has made a profit or incurred a loss. Also, determine the quantum of profit or loss.

2. Write a program to accept a number from the user and find the sum of the digits of the number; repeat the operation till the sum gets to be a single digit number.

3. Write a program to accept a number "n" from the user; find the sum of the series $1/2^3+1/3^3+1/4^3$……. $+1/n^3$

4. Write a program to print the Fibonacci series up to the number 34. (Example: 0,1,1,2,3,5,8, 13,… The Fibonacci series starts with 0 and 1, and the succeeding numbers of the series are arrived at by adding the previous 2 numbers.)

5. Write a program to accept a number "n" from the user; then display the series 1,3,5,7,9,…,n and find the sum of the numbers in this series.

Set-2:

Functions and Native Datatypes

1. Write a program to create an array of size 16; randomly generate numbers in the range 1 – 75 and store in this array.
   a. Sort the array elements in descending order.
   b. Print the list of prime numbers present in the array.

   Hint: Import the "random" module in your program and use the random.randint() function.

2. Write a program that uses a dictionary to store student details. The key should be *student_id* and the value *student_name*. The program must have the following options.
   a. Add student details by getting *student_id* and *student_name* as input from the user.
   b. Display all the student details. If the dictionary is empty, a message "Dictionary is empty" should be displayed.
   c. Display the details of a particular student by getting the *student_id* from the user. If the entered *student_id* does not exist in the dictionary, a message "Entered student_id does not exist" should be displayed.

3. Write program to generate the first "n" ("n" being the input from the user) prime numbers and store those numbers in a list.

   Perform the following operations on the list:
   a. Display all the elements in ascending order by using predefined function.
   b. Find the sum of all the elements in the list and display the same.
   c. Square each element in the list and display the new list.

OOPS

4. Create a class ***Person*** which has the attributes ***id***, ***name*** and ***address*** as well as the method ***displaydetails()*** which displays these attributes.

   Create two subclasses ***Student*** and ***Staff***.

   The ***Student*** subclass contains two attributes ***course*** and ***fees paid***. Override the ***displaydetails()*** method so that these two attributes are also displayed along with ***id***, ***name*** and ***address***.

   The ***Staff*** subclass contains the attributes ***year of joining*** and ***salary***. Override the ***displaydetails()*** method so that these two attributes are also displayed along with ***id***, ***name*** and ***address***.

   Create a container (use list or array) to store details of 2 students and 3 staff members. Iterate through the container and display the details of each student and staff member.

Exception Handling

5. Create a class ***Employee*** having the following attributes - ***empId***, ***empName***, ***designation***, ***basic_pay*** and ***hra*** and the following method.

calculatehra() – method used to calculate hra
printdetails() – method used to show the details of the employee

hra is calculated based on the designation per the details given below:

Designation	Hra
Manager	10% of basic_pay
Officer	12% of basic_pay
Clerk	5% of basic_pay

Define a constructor with the following 4 parameters: ***empId***, ***empName***, ***designation*** and ***basic_pay***. On creation of the object, check that the basic_pay should be greater than 500. If not, raise a custom exception ***LowbalanceException***. If this exception is not raised then display the details of the employee created by invoking the method ***printdetails()***.

Set-3:

1. Create a database named “PY_Assignment_<yourName>”. Create a table named “Student_Details” with the following attributes:
   a. Reg_no
   b. Student_Name
   c. Branch

d. Batch
e. Total Marks scored in SEM-I

Perform the following operations:

i. Insert 5 records into the table through PY Scripts.
ii. Add 5 marks as grace to each student record and update the “Total Marks scored in SEM-I”
iii. Remove the details of the student who has scored the lowest marks. (You can consider the Reg_no of the student who has cored the lowest marks in the “where” clause).
iv. Display the details of all the students.

2. Write a Python script to display the following output:
    a. Write into the file
    b. Read from the file
    c. Exit

- When the user selects option “a”, create a file named “Demonetization.txt” and write a paragraph about the effects of demonetization through Python script.
- When the user selects option “b”, display the contents of this file on the console.
- When the user selects option “c”, display the message “You have chosen to exit the operation. Thanks.”.

If the user chooses option “b” without creating the file, display an error message “File not yet created. Please create the file before accessing it.”.

3. Write a Python script to read the current date from the user. If the user enters the date as 12-31-2016 (mm-dd-yyyy), replace all the occurrences of “-“ with “/”. On the other hand, if the user enters the date as 12/31/2016 (mm/dd/yyyy), replace all occurrences of “/” with “-“.

4. Write a Python script to convert all the following words from singular to plural:
    a. Story -- Stories
    b. Emergency -- Emergencies
    c. Qualify – Qualifies
    d. Fly -- Flies

    Hint: Replace the last occurrence of ‘y’ in the word with ‘ies’.

5. Check if the following statements are interrogative or assertive.
    a. “Do you have a Pen?”
    b. “An eagle flies high in the sky.”

c. “Does aluminium rust?”

<u>Hint</u>: If a statement begins with does/do/is/are, they can be considered as interrogative. If not, they are assertive.