St. Cecilia's Public School
Subject: Computer Science
Class: XII
Session: 2020-21

List of Python Programs for Practical File

Q1. Write a program to make functions to find the number of uppercase letters, lowercase letters and digits in a string passed as an argument to these functions. Name the functions as **noofupper(), nooflower(), noofdigits()** and call them in the program to generate a statistical table of the above.

Q2. Write a program to generate a random number in the range **m-n** where **m** and **n** are entered by the user.

Q3. Create a module named **'Listopr.py'** which has functions to:

- Display the list
- Read the list
- Sort the list using bubble sort method
- Search for an element in the list

Write a program to import the module and call these functions in the program using a menu driven program.
Note: Don't use built-in functions of Python library

Q4. Write a program to read a **text file line by line** and display the words separated by a **'$'** sign.

Q5. Write a program to create a copy a text file **'Old.txt'** by changing all instances of digits with a **'#'** sign in the new file named as **'New.txt'**.

Q6. Write a program to read a text file **'letters.txt'** containing many lines of text to find the **most commonly occurring word(s).**

Q7. Write a program to create a **binary file** named **'student.dat'** containing **(rollno, name, marks)** and write a function **search**() to search for a rollno in the file and display the details. Call the function in the program.

Q8. Write functions to **delete a record** in the **binary file 'student.dat'** created in Q7 using:

- **Name of student**
- **Rollno of student**

Call these functions in the __main__ program to delete the records depending upon **user's choice** and display the **remaining records.**

Q9. Write a program to create a file named **'Item.dat'** having the following dictionary as a record

{"Itemid":value, "Itname":value, "Price":value}

Also display all the records of the file.

Q10. Write a function to update the file **'Item.dat'** defined in Q9 by updating the itemname and price of items for an itemid. Use this function in a program for updating the file.

Q11. Write a program to create a CSV file named **"Empl.csv"** to store employee data (Employee id, employee name, designation and salary) with the delimiter comma. Obtain the data from the user and write 3 records into the file. Also display the contents of the file.

Q12. Write a program to display the employee's data whose employee id has been given by the user from the file **"Empl.csv"** created in Q11.

Q13. Write a program to read the file **"Empl.csv"** and creates another file **"newemp.csv"** with the delimiter as semi colon ';'. Also display the contents of the new file.

Q14. Write an interactive menu driven program to implement Stack as list containing integer values. The menu contains the following options:

1. Push
2. Pop
3. Display stack
4. Exit

Q15. Write an interactive menu driven program to implement a Stack containing travel details (PNo, PName) of a passenger. The menu contains the following options:

1. Push
2. Pop
3. Display stack
4. Exit

Q16. Write an interactive menu driven program to implement a Queue containing membership details (Memno, MemName) of a member in a sports club. The menu contains the following options:

1. Insert
2. Delete
3. Display Queue
4. Exit

Q17. Design a Python application that fetches all the records from the **empl** table of the **emp** database and displays those records who match the job entered by the user.

Q18. Design a Python application that fetches only those records from the **Doctor table** (created in MySQL for practical queries) of the **Hospital database** whose department is **cardiology** and displays them.

Q19. Design a Python application that deletes records from the **Empl** table of database **emp** that are from **deptno 20**. Also display all the remaining records.

Q20. Design a Python application that increases the **charges by 100** of the doctors of **'Surgery'** department in the table **Doctor** of the **Hospital** database. Also display all the records.

****