**Lab Exercises 5 (Based on Modules, Packages & File Handling)**

1. Write a Python program to read an entire text file.

2.   Write a program that counts lines and characters in a file. With your favorite text editor, code a Python module called **mymod.py**, which exports three top-level names:

a) **A countLines(name) function** that reads an input file and counts the number of lines in it

b) A countChars(name) function that reads an input file and counts the number of characters in it

c) A test(name) function that calls both counting functions with a given input file­name.

All three mymod functions should expect a filename string to be passed in.

Now, test your module interactively, using import and name qualification to fetch your exports.

3. Test your mymod module from Exercise 2 interactively, by using from to load the exports directly, first by name, then using the from\* variant to fetch every­thing.

4.    Now, add a line in your mymod module that calls the test function automati­cally only when the module is run as a script, not when it is imported The line you add will probably test the value of \_\_name\_\_ for the string "\_\_main\_\_",. Import the module and test its functions interactively.

5.    Write a second module**, myclient.py**, which imports mymod and tests its functions; run myclient from the system command line. If myclient uses from to fetch from mymod, will mymod’s functions be accessible from the top level of myclient? What if it imports with import instead? Try coding both variations in myclient and test interactively, by importing myclient .

6.      *Package imports*. Finally, import your file from a package. Create a subdirectory called mypkg nested in a directory on your module import search path, move the mymod.py module file you created in exercises 2 or 4 into the new directory, and try to import it with a package import of the form: import mypkg.mymod.

#### 7.  Write a Python program to read first n lines of a file.

#### 8. Write a Python program to append text to a file and display the text.

#### 9. Write a Python program to read a file line by line and store it into a list.

#### 10. Write a program to print each line of a file in reverse order.

#### 11. Write a Python program to write a list content to a file.

#### 12. Write a program to compute the number of characters, words and lines in a file.

#### 13. Write a program to accept string/sentences from the user till the user enters “END” to. Save the data in a text file and then display only those sentences which begin with an uppercase alphabet.

#### 14. Write a program to enter the following records in a binary file:

Item No integer

Item\_Name string

Qty integer

Price float

Number of records to be entered should be accepted from the user. Read the file to display the records in the following format:

Item No:

Item Name :

Quantity:

Price per item:

Amount: ( to be calculated as Price \* Qty)