SET 1

1. Write a program that does the following in order:
   - Asks the user to enter a number “x”
   - Asks the user to enter a number “y”
   - Prints out number “x”, raised to the power “y”.
   - Prints out the log (base 2) of “x”.

2. Write a program that prompts for a file name, then opens that file and reads through the file, and print the contents of the file in upper case. Use the file **words.txt** to produce the output below.

3. Write a program that prompts for a file name, then opens that file and reads through the file, looking for lines of the form: X-DSPAM-Confidence: 0.8475

Count these lines and extract the floating point values from each of the lines and compute the average of those values and produce an output. Do not use the sum() function or a variable named sum in your solution.

You can download the sample data at http://www.py4e.com/code3/mbox-short.txt

4. Open the file **romeo.txt** and read it line by line. For each line, split the line into a list of words using the split() method. The program should build a list of words. For each word on each line check to see if the word is already in the list and if not append it to the list. When the program completes, sort and print the resulting words in alphabetical order.

5. Open the file **mbox.txt** and read it line by line. When you find a line that starts with 'From ' like the following line: From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008

You will parse the From line using split() and print out the second word in the line (i.e. the entire address of the person who sent the message). Then print out a count at the end.

Hint: make sure not to include the lines that start with 'From:'. Also look at the last line of the sample output to see how to print the count.

6. Write a program to read through the mbox-short.txt and figure out who has sent the greatest number of mail messages. The program looks for 'From ' lines and takes the second word of those lines as the person who sent the mail. The program creates a Python dictionary that maps the sender's mail address to a count of the number of times they appear in the file. After the dictionary is produced, the program reads through the dictionary using a maximum loop to find the most prolific committer.

7. Write a program to read through the **mbox.txt** and figure out the distribution by hour of the day for each of the messages. You can pull the hour out from the 'From ' line by finding the time and then splitting the string a second time using a colon.

From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008

Once you have accumulated the counts for each hour, print out the counts, sorted by hour.

SET 2

1. Write Python program to demonstrate Multiple Inheritance.

2. Program to demonstrate the Overriding of the Base Class method in the Derived Class.

3. Pretend that someone has already done the specific parsing, and has left you with variables that contain the following information for a news story:

● **globally unique identifier (GUID)** - a string

● **title** - a string

● **description** - a string

● **link to more content** - a string

● **update** - a datetime

We want to store this information in an object that we can then pass around in the rest of our program. Your task, in this problem, is to write a class, **NewsStory**, starting with a constructor that takes (guid, title, description, link, update) as arguments and stores them appropriately. **NewsStory** also needs to contain the following methods:

● get_guid(self)

● get_title(self)

● get_description(self)

● get_link(self)

● get_pubdate(self)

The solution to this problem should be relatively short and very straightforward. Once you have implemented **NewsStory** all the **NewsStory** test cases should work.

SET 3

1. Write Python program to calculate the Arc Length of an Angle by assigning values to the radius and angle data attributes of the class ArcLength.

2. Write Python Program to simulate a Bank Account with support for **depositMoney**, **withdrawMoney** and **showBalance** Operations.

3. Given three Points (x1, y1), (x2, y2) and (x3, y3), write a Python program to check if they are Collinear.

SET 4

1. Write Pythonic code to create a function named **move_rectangle()** that takes an object of Rectangle class and two numbers named **dx** and **dy**. It should change the location of the Rectangle by adding **dx** to the **x** coordinate of corner and adding **dy** to the **y** coordinate of corner.

2. Write a function which receives a variable number of strings as arguments. Find unique characters in each string.

3. Write Pythonic code to create a function called most_frequent that takes a string and prints the letters in decreasing order of frequency. Use dictionaries.

4. Consider the string ‘brontosaurus’. Write Pythonic code that implements and returns the functionality of histogram using dictionaries for the given string. Also, write the function print_hist() to print the keys and their values in alphabetical order from the values returned by the histogram function.