DEPARTMENT: Computer Science & Engineering		
QUESTION BANK		
SEMESTER: VI	SUBJECT: Python Application Programming	SUB CODE: 15CS664

Module 1

1. Explain each of the following using an example of a human capability: (1) Central processing unit, (2) Main Memory, (3) Secondary Memory, (4) Input Device, and (5) Output Device. For example, "What is the human equivalent to a Central Processing Unit"?

2 a. What is the function of the secondary memory in a computer?
b. Where in the computer is a variable such as "x" stored after the following Python line finishes?

```
x = 123
```

c. How do you fix a "Syntax Error"?

3. Write a program to prompt the user for hours and rate per hour to compute gross pay.

```
Enter Hours: 35
Enter Rate: 2.75
Pay: 96.25
```

4. Rewrite your pay program using try and except so that your program handles non-numeric input gracefully by printing a message and exiting the program. The following shows two executions of the program:

```
Enter Hours: 20
Enter Rate: nine
Error, please enter numeric input
```

5. Write a program which prompts the user for a Celsius temperature, convert the temperature to Fahrenheit, and print out the converted temperature.

6. Write a program to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error message.

If the score is between 0.0 and 1.0, print a grade using the following table:

Score	Grade
>= 0.9	A
>= 0.8	B
>= 0.7	C
>= 0.6	D
< 0.6	F

7. Rewrite the grade program from the previous chapter using a function called computegrade that takes a score as its parameter and returns a grade as a string.

Module 2

1. Write a program which repeatedly reads numbers until the user enters "done". Once "done" is entered, print out the total, count, and average of the numbers. If the user enters anything other than a number, detect their mistake using try and except and print an error message and skip to the next number.

2. Write a program that prompts for a list of numbers as above and at the end prints out both the maximum and minimum of the numbers instead of the average.
3. Write a program to display Armstrong numbers within a given range
4. Take the following Python code that stores a string: ' str = 'X-DSPAM-Confidence:0.8475' Use find and string slicing to extract the portion of the string after the colon character and then use the float function to convert the extracted string into a floating point number.
5. Write a program to read through a file and print the contents of the file (line by line) all in upper case.
6. Write a program to prompt for a file name, and then read through the file and print all the line starting with string subject striping the white space from left

Module 3

1. Write the program that prompts the user for a list of numbers and prints out the maximum and minimum of the numbers at the end when the user enters "done". Write the program to store the numbers the user enters in a list and use the max() and min() functions to compute the maximum and minimum numbers after the loop completes. Enter a number: 6 Enter a number: 2 Enter a number: 9 Enter a number: 3 Enter a number: 5 Enter a number: done Maximum: 9.0 Minimum: 2.0
2. See & save the contents of a file romeo.txt **But soft what light through yonder window breaks** **It is the east and Juliet is the sun** **Arise fair sun and kill the envious moon** **Who is already sick and pale with grief** Write a program to 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 function. For each word, check to see if the word is already in a list. If the word is not in the list, add it to the list. When the program completes, sort and print the resulting words in alphabetical order. **Expected Output:** Enter file: romeo.txt ['Arise', 'But', 'It', 'Juliet', 'Who', 'already', 'and', 'breaks', 'east', 'envious', 'fair', 'grief', 'is', 'kill', 'light', 'moon', 'pale', 'sick', 'soft', 'sun', 'the', 'through', 'what', 'window', 'with', 'yonder']

3. Write a program that categorizes each mail message by which day of the week the commit was done. To do this look for lines that start with "From", then look for the third word and keep a running count of each of the days of the week. At the end of the program print out the contents of your dictionary (order does not matter). Sample Line: From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008 Sample Execution: Enter a file name: mbox-short.txt {'Fri': 20, 'Thu': 6, 'Sat': 1}
4. Write a program that reads a file and prints the letters in decreasing order of frequency. Your program should convert all the input to lower case and only count the letters a-z. Your program should not count spaces, digits, punctuation, or anything other than the letters a-z. Compare your findings with standard frequency of English language as given below. a 8.167% b 1.492% c 2.782% d 4.253% e 12.702% f 2.228% g 2.015% h 6.094% i 6.966% j 0.153% k 0.772% l 4.025% m 2.406% n 6.749% o 7.507% p 1.929% q 0.095% r 5.987% s 6.327% t 9.056% u 2.758% v 0.978% w 2.360% x 0.150% y 1.974% z 0.074%
5. Write a simple program to simulate the operation of the grep command on Unix. Ask the user to enter a regular expression and count the number of lines that matched the regular expression:

Module 4

1.What are the attributes of a class in Python
2. What do you mean by "orchestrating the movement of data between objects". Discuss with example code.
3. How multiple inheritance can be used in Python programming. Explain with example program.
4. Define "Life cycle of an object". Explain constructing and destructing the objects with example program.

Module 5:

1. Change the socket program socket1.py to prompt the user for the URL so it can read any web page. You can use split('/') to break the URL into its component parts so you can extract the host name for the socket connect call. Add error checking using try and except to handle the condition where the user enters an improperly formatted or non-existent URL.
2. Change the urllinks.py program to extract and count paragraph (p) tags from the retrieved HTML document and display the count of the paragraphs as the output of your program. Do not display the paragraph text, only count them. Test your program on several small web pages as well as some larger web pages.
3. Change either the www.py4e.com/code3/geojson.py or www.py4e.com/code3/geoxml.py to print out the two-character country code from the retrieved data. Add error checking so your program does not traceback if the country code is not there. Once you have it working, search for "Atlantic Ocean" and make sure it can handle locations that are not in any country.
4. Write a program using SQL commands in python two join two tables