ESC101 : Fundamental of Computing

Lab Test for 10th November 2008

Instructions:

1. The duration of the test is 3 hrs (from **2:00 pm to 5:00 pm**).
2. **Directory Structure:** Create a directory and name it with your roll number. e.g. If your roll number is $Y8001$, the directory should be named $Y8001$ (`Y` should be upper case). Inside this directory, create three directories named **One**, **Two** and **Three** corresponding to the programs for the three problems of this test. In directory **One**, create file *TwinPrimes.java* for the java program for first problem. In directory **Two**, create files *Point.java* and *Point_example.java* for the second problem. In directory **Three**, create file *Permutation.java* for the java program of the third problem.
3. Use of good coding practices (indentation, use of methods, and proper naming of variables and methods) carries weightage.

Problems:

1. **Twin Primes :** (marks = 10)
   A pair of numbers are called *twin primes* if they are both prime and differ from each other by 2. e.g. $(5, 7), (11, 13), (17, 19)$ are twin primes.

   Write a JAVA program that takes a positive integer n from command-line and prints all the pairs of twin prime numbers between 2 and n in the following format:

   (3, 5),

   (5, 7),

   (11, 13),

   ⋮

2. **Point in 3D :** (marks = 10)
   You have to design a JAVA class Point that represents a point in 3 dimensional space.

   The coordinates — (x, y, z) of such a point should be of type *double*.

   Write two constructors for this class. One constructor initializes the point to origin $(0, 0, 0)$. The second constructor takes three arguments of type *double* and initializes the coordinates of the point to the same value as these arguments.

   You should design and implement the following three *non-static* methods:

   (a) **print**: Prints the coordinates of *this* point as: (x, y, z).

   (b) **distance**: Takes an object of the class Point as argument and returns its distance from *this* Point.

   (c) **translate**: Takes three arguments $xTranslate$, $yTranslate$ and $zTranslate$ of type *double* and translates *this* point by those values. i.e. *this* point becomes $(x+xTranslate, y+yTranslate, z+zTranslate)$.

   Now, write a *main* method (in the file *Point_example.java*) that tests the class that you have written. The program takes the coordinates of a point from command line as input. Let us denote this point as P. The program should print the following information with appropriate messages:

   (a) The coodinates of P.

   (b) Distance between P and point $(10, 10, 10)$.

(c) New coordinates of point P after it is translated by 2, -3 and 5 units along x, y, and z directions respectively.

3. **Generalized Permutation** (marks = 20)
Let S be a string of characters and L be a positive integer. All the characters of the string S are not necessarily distinct. However, you may assume that if a character appears multiple times in the string then all those occurrences are contiguous in S. Example of S are `addddnnmmbc`, `dfassser`, `fort`. You have to write a program which receives a string S and a positive integer L from command line and print all permutations of length L from characters of string S. It must be noted that the number of times a character appears in a permutation is bounded by the number of times it appears in the original string S. Some examples are as follows.

- If S=`aaaab`, and $L = 2$, then a correct output is :

```
aa
ab
ba
```

- If S=`sooon`, and $L = 3$, then a correct output is :

```
soo
oso
oos
noo
ono
oon
ooo
osn
ons
nos
nso
sno
son
```