**ESC101 : Fundamental of Computing**

Lab Test for 14th 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 *LCM.java* for the java program for first problem. In directory **Two**, create files *HollowCylinder.java* and *CylinderExample.java* for the second problem. In directory **Three**, create file *Combination.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. **Least Common multiple :** (12 marks) Given a set of positive integers, the *Least Common Multiple* of the set is the smallest positive number which is divisible by each number of the set. Write a program which reads three integers from command line and prints their least common multiple (LCM). For example, for integers $4, 6, 12$, the LCM is 12. For $n = 5, 2, 12$, the LCM is 60. You may assume that the input numbers are positive integers in the range of `int`. You may use the relation between $GCD$ and $LCM$ of two numbers, and the following fact : For any three positive integers $a, b, c$

$$LCM(a, b, c) = LCM(a, LCM(b, c))$$

2. Design a JAVA class *HollowCylinder* for objects which are hollow cylinders with a lid which can be removed to open the cylinder. The attributes should be radius and heights of type double. No method outside the class *HollowCylinder* should be allowed direct access to the attributes of an object of the class *HollowCylinder*.

   Design two constructors for this class.

   - One constructor is parameter-less and initializes the radius and height of the cylinder to 1.
   - Another constructor should have two parameters of type *double* which correspond to the radius and height of the cylinder created.

   Design the following non-static methods which should be accessible from methods external to the class *HollowCylinder*.

   - **radius** : it should return radius of *this* cylinder.
   - **height** : it should return the height of *this* cylinder.
   - **Area** : it should return the outer surface area of *this* cylinder after its lid has been closed. Use $\pi = 3.14$.
   - **Volume** : it should return the volume of *this* cylinder.
   - **Encloses** : it has a parameter $C$ which is reference to an object of class *HollowCylinder* and it should return true if *this* cylinder can enclose the cylinder whose reference is $C$. We say that a cylinder $A$ can enclose another cylinder $B$ if we can place $B$ inside $A$ and close the lid of $A$. You may ignore the thickness of the lid.

Write a JAVA program *CylinderExample.java* which receives two command-line arguments of type double such that the first argument corresponds to the radius and and the second argument corresponds to the height of a cylinder. It creates a cylinder, say $A$, and then initializes its attributes according to the arguments read. It should create another cylinder, say $B$, using parameter-less constructor. It should then compute the following information and print suitable messages on the console.

- The radius and height of cylinder $A$
- The surface area of cylinder $A$.
- The volume of cylinder $A$.
- Whether $A$ can enclose $B$ ?

3. **Generalization of Combination**

Write a program which generates all combinations of $L$ elements from a set $A$ such that each element of the set appears at least $k$ times and at most $m$ times. The set $A$ is given as a string (without any repetition of characters). The four arguments $A$, $L$, $k$ and $m$ are read from command-line in this fixed order only.
Examples are :

- for $A = \{\texttt{a,b,c}\}$, $L = 4$, $k = 1$, $m = 2$, the output should be

```
[a,a,b,c]
[a,b,b,c]
[a,b,c,c]
```

- for $A = \{\texttt{a,b,c}\}$, $L = 3$, $k = 0$, $m = 2$, the output should be

```
[a,b,c]
[a,a,b]
[a,a,c]
[a,b,b]
[a,c,c]
[b,c,c]
[b,b,c]
```

- for $A = \{\texttt{a,b,c}\}$, $L = 3$, $k = 2$, $m = 3$, the output should be

```
Sorry, no such combination is possible.
```