Programming for Engineers (ECE-203)
Winter 2011–12
Programming Assignment 1

IN-LAB COMPONENT

Problem 1: Type up the following program in an editor and save it in a file called `Cube.java` (or type it up in NetBeans or your favorite IDE window). Compile and run it.

```
public class Cube
{
   public static void main( String args[] )
   {
      int number = 5;
      int cube = number*number*number;
      System.out.println( "The cube of " + number + " is " + cube );
   }
}
```

Problem 2: The following program prints out a sullen face. Modify the program to print a smiling face (or one with hair and ears?). The best face artist in each section will receive bonus points! Your "portrait", as far as possible, should not be obvious from looking at the code but only when the code is executed.

```
public class Face
{
   public static void main( String args[] )
   {
      int i = 0;
      int j = 0;
      System.out.println();
      System.out.println( "      " + i + " " + j );
      i = i + 1;
      System.out.println( "       " + i );
      System.out.println( "       =" );
      System.out.println();
   }
}
```

Problem 3: Write a program called `Concat.java` which creates two `String` objects and prints them. It then creates a new `String` which is a concatenation of the previous two. The program should print the third string in its original form and then with all its characters changed to lower case letters. For this exercise, you may need to look up page 1005 of the textbook in Appendix D (the Java Library) on the class `java.lang.String` and its methods.

Problem 4: This is a version of Exercise P2.5 from the textbook. The `intersection` method in the class `Rectangle` computes the *intersection* of two rectangles—that is, the rectangle that is formed by two overlapping rectangles. This method accepts another `Rectangle` object as a parameter and returns a new `Rectangle` object that is the intersection. Write a program called `Intersection.java` that constructs two `Rectangle` objects, prints them, and then prints the intersection. What is printed if the intersection is empty?

Problem 5: Compile and run the program `AvailMemory.java` below. The maximum memory is the maximum amount of memory that the Java virtual machine is allowed to use, not the maximum amount of memory available on the computer. Of this amount, the total memory is the amount of memory that has been currently allocated for use by the Java virtual machine. The free memory is the amount out of the total memory that is not yet in use. The maximum memory and the initial allocation (total memory) can be controlled by you; for e.g., you can try the command `java -Xmx2048m -Xms1024m AvailMemory` to run the program (figure out how you will do this in your development environment). This will allow the Java virtual machine to allocate a maximum of 2048 MByte with an initial allocation of 1024 MBytes.

```
import java.lang.Runtime;

public class AvailMemory
{
    public static void main(String[] args)
    {
        Runtime currentRuntimeEnv = Runtime.getRuntime();

        System.out.println( "Maximum memory (bytes) = "
                                + currentRuntimeEnv.maxMemory());
        System.out.println( "Allocated memory (bytes) = "
                                + currentRuntimeEnv.totalMemory());
        System.out.println( "Free memory (bytes) = "
                                + currentRuntimeEnv.freeMemory());
        System.out.println( "Number of available processors = "
                                + currentRuntimeEnv.availableProcessors());
    }
}
```

TAKE-HOME COMPONENT
(demo due during your next lab session)

Read all of Chapter 1 and Sections 2.1 to 2.10 of Chapter 2.

Problem 6: Write a program `HollePrinter.java` which switches the letters “e” and “o” in a `String`. Demonstrate that the string “Hello, World!” turns into “Holle, Werld!”. *Hint:* Use the method `replace()` repeatedly.

Problem 7: Use the class `GregorianCalendar` as described in Project 2.1 of the textbook (page 77). You will have to consult the Java API to complete this assignment. Your program should print out the following:

(a) The date of your birth (make up a fictitious date since you should not reveal your date of birth to anyone who has no need for it).

(b) The date on which you will be one billion seconds old.

Hint: Consider using the `add()` method of `GregorianCalendar`.