Fall 2012

Final Exam

COSC 117

Name: ____

Write all of your responses on these exam pages. If you need more space please use the backs. Make sure that you show all of your work.

Part 1: Short Answer (4 Points Each): Give a definition or short answer for each of the following.

1. What is the difference between a compiler and an interpreter?

2. What are the three types of programming errors? Briefly describe each of them.

3. What is the scope of a function parameter?

4. What is the difference between an Applet and an Application?

5. What happens when you overload an integer?

Part 2: Program Traces (20 Points Each)

1. For each of the program inputs below write the output of the program.

```
import java.util.Scanner;

public class FinalTrace001 {

   public static int doSomething(int x, int y, int z) {
      x = y + z;
      y--;
      z++;
      return x * y * z;
   }

   public static int doSomethingElse(int x, int y, int z) {
      int temp = 0;
      temp = x;
      x = z;
      z = temp;
      temp = y;
      y = z;
      z = temp;
      return y;
   }

   public static int doYetSomethingElse(int x, int y, int z) {
      while (x < z) {
         y++;
         x++;
      }
      return y;
   }

   public static void main(String[] args) {
      Scanner keyboard = new Scanner(System.in);

      System.out.print("Input x, y, and z: ");
      int x = keyboard.nextInt();
      int y = keyboard.nextInt();
      int z = keyboard.nextInt();

      System.out.println(doSomethingElse(z, y, x));
      System.out.println(doSomething(x, y, z));
      System.out.println(doYetSomethingElse(x, y, z));

      System.out.println(doYetSomethingElse(doYetSomethingElse(x, y, z),
            doSomethingElse(y, z, x), doSomething(z, x, y)));
   }
}
```

(a) `Input x, y, and z: 1 2 3`

(b) `Input x, y, and z: 3 5 1`

2. For each of the program inputs below write the output of the program.

```
import java.util.Scanner;

public class FinalTrace002 {

   public static String doSomething(String s, int x) {
      String retstr = "";
      for (int i = 0; i < s.length(); i += x) {
         String c = s.charAt(i) + "";
         retstr += c;
      }
      return retstr;
   }

   public static int doSomething(int x, String s) {
      int q = 0;
      s = s.toLowerCase();
      for (int i = 0; i < s.length(); i++) {
         char c = s.charAt(i);
         if (c == ’ ’ || c < ’m’)
            q += (x-1);
      }
      return q;
   }

   public static double doSomething(double m, int n) {
      return n*n/2 + m*3;
   }

   public static void main(String[] args) {
      Scanner keyboard = new Scanner(System.in);
      System.out.print("Input n: ");
      int n = keyboard.nextInt();
      System.out.print("Input m: ");
      int m = keyboard.nextInt();
      String dumb = keyboard.nextLine(); // Clear enter key
      System.out.print("Input str: ");
      String str = keyboard.nextLine();
      System.out.println(doSomething(str, n));
      System.out.println(doSomething(n, str));
      System.out.println(doSomething(n, m));
   }
}
```

(a)
```
Input n: 3
Input m: 13
Input str: This is the last COSC 117 test!!!
```

(b)
```
Input n: 2
Input m: 3
Input str: This test is far too easy.
```

3. For each of the program inputs below write the output of the program.

```
import java.util.Scanner;

public class FinalTrace003 {

   public static void copy(int[] A, int[] B) {
      for (int i = 0; i < B.length; i++)
         if (i < A.length)
            B[i] = A[i];
   }

   public static void Print(int[] Arr) {
      for (int i = 0; i < Arr.length; i++) {
         System.out.print(Arr[i] + " ");
      }
      System.out.println();
   }

   public static void main(String[] args) {
      Scanner keyboard = new Scanner(System.in);
      System.out.print("Input list: ");
      int[] A = new int[1000];
      int input = 1;
      int count = 0;
      while (input != 0) {
         input = keyboard.nextInt();
         if (input != 0) {
            A[count] = input;
            count++;
         }
      }

      int[] B = new int[count];
      copy(A, B);
      for (int i = 0; i < count / 2; i++) {
         int temp = B[i];
         B[i] = B[B.length - i - 1];
         B[B.length - i - 1] = temp;
      }
      Print(B);

      copy(A, B);
      for (int i = 0; i < count / 2; i++) {
         B[i] = B[B.length - i - 1];
         B[B.length - i - 1] = B[i];
      }
      Print(B);

      int[] C = new int[count / 2];
      copy(A, B);
      copy(B, C);
      int b = 1;
      for (int i = 0; i < C.length - 1; i++) {
         C[i] += C[i + 1];
         b *= C[i];
      }
      Print(C);
      System.out.println(b);
   }
}
```

(a) `Input list: 1 2 3 4 5 6 7 8 9 0`

(b) `Input list: 1 2 3 4 0 4 3 2 1`

Part 3: Coding (20 Points Each)

1. Write a program that will take an input of a positive integer number from the user, say n. Repeat the following process until n is equal to 1. If n is even replace n by $n/2$ and if n is odd replace n by $3n+1$. The program must print out the sequence along with a count of the number of numbers in the sequence. Also the calculation of the next number must be done in a function that takes in the last number as a parameter and returns the next number.

2. Write a function that takes a one-dimensional array of integers and an integer target value and does a linear search for the target value in the array. The return value is to be the position of the target value in the array and -1 if the target value is not in the array.

3. Write a function that will take in a single string as its only parameter and returns true if the input string is a palindrome or false if the string is not a palindrome. A palindrome is a string that is the same when written forwards or backwards. When checking a palindrome we ignore cases and spaces. So `A Toyota` would be considered a palindrome because if we remove all of the spaces and write all the letters in the same case we get `atoyota` which when written backwards gives exactly the same word.

4. Write a function that will take in a two-dimensional array of doubles as its only parameter and return a one-dimensional array of doubles that holds the row averages of the input array. An example of an input array (left) and its row averages array (right) are below.

```
2    3    3         2.66667    5.33333    4.66667    2.0    4.66667
6    1    9
7    6    1
3    1    2
7    7    0
```

5. Below is the beginning of a Triangle class. The class has data members a, b and c which are the lengths of the sides of the triangle. Fill in the details for each of the function,

   (a) The default constructor sets all sides to 0.

   (b) The other constructor sets the side lengths to the input parameters.

   (c) The `Area` function returns the area of the triangle using Heron's formula, which is, $\sqrt{s(s-a)(s-b)(s-c)}$ where s is the semi-perimeter, which is one-half the length of the perimeter.

   (d) The `Perimeter` method returns the perimeter, which is the sum of the lengths of the sides of the triangle.

   (e) The `isRight` function returns true if the triangle is a right triangle and false otherwise. Recall by the Pythagorean theorem that a triangle is right if the square of one side is equal to the sum of the square of the other two sides.

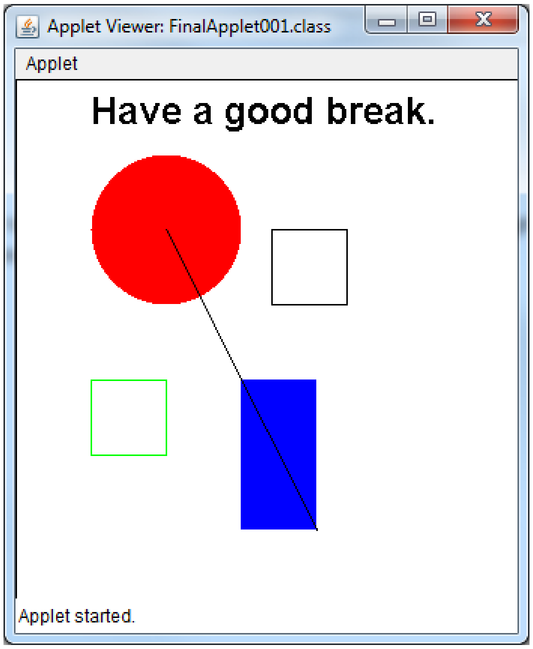
```
public class Triangle {
   private double a;
   private double b;
   private double c;

   public Triangle(){



   }

   public Triangle(double s1, double s2, double s3){



   }
}

public double Area() {

}

public double Perimeter() {

}
```

```
    }

    public boolean isRight() {

    }
}
```

6. The number of ways to select p items from a set of n items is called a combination. For example the number of ways to select 2 items from a set of 4 is 6 as the following example illustrates. Say our set is $\{A, B, C, D\}$ then the possible selections of two of these items are $AB, AC, AD, BC, BD,$ and CD. Just to simplify the language let's denote Comb(n, p) to be the number of ways to select p items from a set of n items. Now if p is greater than n then the calculation Comb(n, p) = 0, you cannot select 7 items from a set of 5. If p is equal to n then the calculation Comb(n, n) = 1 and if p is equal to 0 then the calculation Comb(n, 0) = 1. In all other cases Comb(n, p) = Comb(n − 1, p − 1) + Comb(n − 1, p). Write a recursive function named `Comb` that takes in two integer parameters n and p and returns the value of Comb(n, p).

7. Write an applet that will produce the image to the right. Note that the center of the red circle is at the position (100, 100) and the background is white.

```
import java.awt.*;
import javax.swing.JApplet;

public class FinalApplet001 extends JApplet {
   public void paint(Graphics g) {
      super.paint(g);
      g.setFont(new Font(Font.SANS_SERIF, Font.BOLD, 25));
      int drawWidth = getWidth();
      int drawHeight = getHeight();
```



```
   }
}
```