CS1007 – Object Oriented Programming

Sample Final Examination

Professor S. Hershkop
Department of Computer Science
Columbia University
Spring 2006

1) Define the following terms, use complete sentences:
    a. JComponent

    b. layout manager

    c. deep copy

    d. accessor method

    e. first come first serve schedule

    f. mutex

2) Write a method in Java, called `mySubString` that takes 2 strings as arguments and returns a `boolean` value: **true**, only if one of the strings in a substring of the other. Do NOT use any **manipulation** methods of the `String` class other than `length()`

3. You have just been hired to run the Information Services Department for a medium size business. The person that previously had your job is nowhere to be found and it is up to you to decipher their cryptic uncommented code.
   Determine what the output to the following program would be if the Math.random() method returns 0.1, and you choose Scissors. Write out and describe the output as precisely as you can. (Note: Don't expect this game to work correctly.)

```
//********************************************************************
//  RockPaperScissors.java
//
//
//
//********************************************************************

import java.util.Scanner;
public class RockPaperScissors
{
   //-----------------------------------------------------------------
   //  Plays the Rock-Paper-Scissors game with the user.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      final int OPTIONS = 3;
      final int ROCK = 1, PAPER = 2, SCISSORS = 3;
      final int COMPUTER = 1, PLAYER = 2, TIE = 3;

      int computer, player, winner = 0;
      int wins = 0, losses = 0, ties = 0;
      String tied="tied";
      String again, times ="charms";
      String tie = "tie";

      Scanner console = new Scanner(System.in);

      do
      {
         computer = (int) (Math.random() * OPTIONS) + 1;
         System.out.println();
         System.out.print ("Enter your choice - 1 for Rock, 2 for " +
                           "Paper, and 3 for Scissors: ");
         player = console.nextInt();

         System.out.println ("My choice was " + computer);


         // Determine the winner
         switch (computer)
         {
            case ROCK:
               if (player == SCISSORS)
                winner = COMPUTER;
               else
                  if (player == PAPER)
                     winner = PLAYER;
                  else
                     winner = TIE;


            case PAPER:
```

```
               if (player == ROCK)
                  winner = COMPUTER;
               else
                  if (player == SCISSORS)
                     winner = PLAYER;
                  else
                     winner = TIE;

            case SCISSORS:
               tied = " after me";
             if (player == PAPER)
                  winner = COMPUTER;
               else
                  if (player == ROCK)
                     winner = PLAYER;
                  else
                     winner = TIE;
                 tie = " lucky ";
         }

         // Print results and increment appropriate counter
         if (winner == COMPUTER)
         {
            System.out.println ("I win!");
            losses++;
         }
         else
            if (winner == PLAYER)
            {
               System.out.println ("You win!");
               wins++;
            }
            else
            {
               System.out.println ("We tied!");
               ties++;
            }

         System.out.println();
         System.out.print ("Play again (y/n)?: ");
         again = console.nextLine();
      }
      while (again.equalsIgnoreCase ("y"));
      String we = "They're";

      // Print final results
      System.out.println();
      if(wins>0)
      System.out.println ("You won " + wins + " times.");
      if(losses>0)
      System.out.println ("You lost " + losses + " times.");
      if(ties>0)
      System.out.println (we + tied  + tie +  times);
   }
}
```

Output:

Description/Comments:

4.

a. Explain the difference between implementing an interface and a derived class.

b. Explain the difference between how method parameters are passed for variables that contain object references and variables that contain primitive data types.

5.

a. What is the point of using generics?

b. Discuss the difference
    i. between runtime and compile time errors
    ii. Would you rather have an error discovered at run time or compile time?

6. What advantages are there for programming your GUI in an applet environment vs SWING ?

7. List what, if anything, is wrong with the following code? And what is this code trying to do

```
public class UnorderedPair<T> extends Pair<T>
{
        public unorderedPair()
        {
                setFirst(null);
                setSecond(null);
        }

        public UnorderedPair(T firstItem, E second Item)
        {
                setFirst(firstItem);
                setSecond(secondItem);
        }

        public Boolean equals(Object otherObj)
        {
                if(otherObj = null)
                        return false;

                else if (getClass() != otherObject)
                        return false;
                else
                {
                        UnorderedPair<T> otherPair = (UnorderedPair<t>)otherObject;
                        return (getFirst().equals(otherPair.getFirst()) &&
getSecond().equals(otherpair.getFirst())
                ||
                (getFirst().equals(otherPair.getSecond()) &&
getSecond().equals(otherPair.getFirst()));
                }
}
```

Extra Credit:

For questions 1 – 2, consider the following class definition:

```
public class MtermEx
{
        private int x;

        public MtermEx(int newValue)
        {
                x = newValue;
        }
}
```

1) Which of the following is true about the class MtermEx?
   a) it has no parent class
   b) it's parent class is Object
   c) it's parent class is Java
   d) it can not be extended
   e) it has a default child called Object

2) If q1 and q2 are objects of MtermEx, then q1.equals(q2)
   a) is a syntax error since equals is not defined in the MtermEx class
   b) is true if q1 and q2 both store the same value of x
   c) is true if q1 and q2 reference the same MtermEx object
   d) is never true
   e) throws a NullPointerException

3) A mouse event is generated by
   a) moving the mouse
   b) clicking the mouse button
   c) double clicking the mouse button
   d) dragging the mouse
   e) all of the above

4) A mouse event is a(n)
   a) listener
   b) object
   c) interface
   d) GUI component
   e) all of the above

5) In order to implement the MouseListener interface, the programmer must define all of the following methods except for
   a) mouseClicked
   b) mouseDragged
   c) mouseEntered
   d) mouseReleased
   e) all of the above must be defined to implement the MouseListener interface