Advanced Computer Science Unit 8 Review

Part I: Multiple Choice (12 questions / 4 points each)

1) Is there any implementation of methods in an interface?
   Can a class implement more than one interface?
   Do interfaces contain instance fields?
   Can you construct an interface object?
   Are methods of an interface public or private?
2) Know about the List interface, especially the methods used in the ArrayList class.
3) What must be done to a class that needs to implement the Comparable interface?
4) Given an interface and 5 classes, select the class least suitable for the interface.
5) Know what polymorphism is
6) Given 5 choices, pick the one that would best serve as the interface for the other 4
7) Given two interfaces and a class that implements both of them, be able to select a true statement concerning the compilation & execution of the class.
8) What are the consequences of adding another method to an interface that has already been implemented by a class?
9) Given the following code:

```
public class GPA implements Comparable
{
  private double myGPA;

  public GPA(double g)
  { ... }
  public int compareTo(Object obj)
  { ... }
  public String toString()
  { ... }
}

public class Driver
{
  public static void main(String[] args)
  {
        <insert code here>
  }
  public static Comparable max(Comparable x, Comparable y)
  {
        if (x.compareTo(y) > 0)
              return x;
        else
              return y;
  }
}
```

   Which of the following lines will present a problem when inserted into main?

```
GPA g1 = new GPA(3.76);
Comparable c1 = new GPA(2.98);
Comparable c1 = new Comparable(1.59);
```

10) Given an interface and five classes, select the class least suitable to implement the interface
11) Given an interface and 3 classes trying to implement it. Select the class or classes that implemented it correctly.

12) Given the following code:

```
      public interface Golfer
      {
            int getHandicap();
            void describeStrengths();
      }
      public class Athlete implements Golfer, Comparable
      {
            private String name;

            public Athlete(String n)
            { ... }
            public int getHandicap()
            { ... }
            public void describeStrengths()
            { ... }
            public int compareTo(Object obj)
            { ... }
      }
```

Which of the following code segments will cause an error?

```
      I.    Golfer g = new Athlete("Bob");
            g.describeStrengths();

      II.   Comparable c = new Athlete("Beth");
            int x = c.getHandicap();

      III.  Athlete a1 = new Athlete("Bill");
            Athlete a2 = new Athlete("Burt");
            int x;
            if (a1.compareTo(a2) < 0)
                  x = a1.getHandicap();
            else
                  x = a2.getHandicap();
```

Part II: Free Response (32 points)

```
 13)  // Write the Shape interface.  It has 3 abstract methods: area()
      // which returns a double equal to the area of the geometric shape,
      // perimeter() which returns a double equal to the perimeter of the
      // geometric shape, and toString().

      public interface Shape
      {
            <for you to do>
      }
```

```
 14)  // Create a Rectangle class that implements both Shape and Comparable.

      public class Rectangle implements Shape, Comparable
      {
            private double length;
            private double width;

            public Rectangle(double len, double wid)
            {
                  length = len;
                  width = wid;
            }
```

```
           // Determines and returns area of rectangle
           public double area()
           {
                 <for you to do>
           }

15)        // Determines and returns perimeter of rectangle
           public double perimeter()
           {
                 <for you to do>
           }

16)        // Returns 1 if area of rectangle referenced by this is larger
           // Returns -1 if area of rectangle referenced by obj is larger
           //
           // If areas are equal, check perimeters.  If perimeters are
           // equal return 0.  Return 1 if perimeter of rectangle referenced
           // by this is larger.  Return -1 if perimeter of rectangle
           // referenced by obj is larger.  It is possible you could be
           // comparing a rectangle to some other shape in this method.
           public int compareTo(Object obj)
           {
                 <for you to do>
           }
           public String toString()
           {
                 return “Rectangle [“ + length + “, “ + width + “]”;
           }
     }
```

17) Create a Right Triangle class that implements both Shape and Comparable.

```
     public class RightTriangle implements Shape, Comparable
     {
           private double leg1;
           private double leg2;

           public RightTriangle(double x, double y)
           {
                 leg1 = x;
                 leg2 = y;
           }

           // Determines and returns area of right triangle
           public double area()
           {
                 <for you to do>
           }

18)  // Determines and returns perimeter of right triangle
           public double perimeter()
           {
                 <for you to do>
           }
           public int compareTo(Object obj)
           {
                 // similar to #16 – no need to do it again!
           }
           public String toString()
           {
                 return  “Right Triangle[“ + leg1 + “, “ + leg2 + “]”;
           }
     }
```

19) Next create a Driver class that will test the interfaces Shape & Comparable and the classes Rectangle & RightTriangle. The Driver contains 10 unique objects placed in an ArrayList. You should test consecutive objects to find out which is greater. Your output should look like this:

```
Rectangle[5.2 , 8.5] is greater than Rectangle[3.9 , 4.7]
RightTriangle[3.7 , 5.1] is greater than RightTriangle[1.8 , 8.6]
Rectangle[8.7 , 5.5] is greater than RightTriangle[7.9 , 6.4]
Shapes are equal
RightTriangle[3.0 , 4.0] is less than RightTriangle[2.0 , 6.0]
```

```
import java.util.ArrayList;

public class Driver
{
      public static void main(String[] args)
      {
            Driver app = new Driver();
      }

      public Driver()
      {
            ArrayList<Comparable> list = new ArrayList<Comparable>();

            list.add(new Rectangle(5.2, 8.5));
            list.add(new Rectangle(3.9, 4.7));

            list.add(new RightTriangle(3.7, 5.1));
            list.add(new RightTriangle(1.8, 8.6));

            list.add(new Rectangle(8.7, 5.5));
            list.add(new RightTriangle(7.9, 6.4));

            list.add(new Rectangle(4.0, 5.0));
            list.add(new Rectangle(5.0, 4.0));

            list.add(new RightTriangle(3.0, 4.0));
            list.add(new RightTriangle(2.0, 6.0));



            <for you to complete>



      }
}
```

Java Concepts Review Assignment:

Pages 429 – 431   Exercises: R9.1, R9.3, R9.4, R9.7, R9.8