WIA1002/WIB1002 Data Structure

Lab: Generics

1. a) Modify the following program to become a generic class called StorePairGeneric.

```
public class StorePair {
   private int first, second;

   public StorePair(int first, int second) {
       this.first = first;
       this.second = second;
   }

   public int getFirst() {
       return first;
   }

   public int getSecond() {
       return second;
   }

   public void setPair(int first, int second) {
       this.first = first;
       this.second = second;
   }

   public String toString() {
       return "first = " + first + " second = " + second;
   }
}
```

b) Override the Object equals() method in the StorePair class to compare the first values of two objects for equality.

c) Have the StorePair class implement the Comparable interface. Override the compareTo()method to compare the first values of two objects.

d) Create a test program that creates three objects of the StorePair generic class called a, b and c. Set the first and second values of a, b, c as (6,4), (2,2), (6,3).

e) Invoke the compareTo()and equals() methods that compares the three objects created in (d) in the test program.

2. Create a generic class called `MyGeneric` that accepts one parameter. Declare a variable called `e` for the type parameter. Create a no-arg constructor. Create a constructor that accepts one generic parameter. Create a setter and getter method for the generic type.

   Create a test program that creates two instances of generic class of type String called `strObj` and of type Integer called `intObj`. Set a value for each of these objects. Display these values using the getter method.

3. In a class called `CompareMax`, create a generic static method called maximum where the generic type extends the `Comparable` interface, which receives three parameters. Find the maximum of three values invoked by the main method.

4. Provide a declaration and implementation of the generic method `minmax()` that takes in an array of generic type and returns a string with the following format: Min = <minValue> Max = <maxValue>. For instance, in your main method, create one object of type array for integers and one object of type string:

   ```
   Integer[] intArray = {5,3,7,1,4,9,8,2};
   String[] strArray = {"red", "blue", "orange", "tan"};
   ```

   `minmax()` method returns `“Min = 1  Max = 9”` For `intArray`
   `minmax()` method returns `“Min = blue  Max = tan”` for `strArray`

   ***Hint:** use `Comparable` interface to compare the values
   ***Hint:** `compareTo()` method:
   - if a > b, it returns positive number
   - if a < b, it returns negative number
   - if a == b, it returns 0

5. Create a class called `FindMax` that contains the following:

   Create a `Circle class` that uses the `Comparable interface`. Declare the radius variable and a single parameterized constructor that accepts this variable.

   In your main program, create 3 different objects of type array (a) for integers that stores the following values, 1,2,3; (b) a list of string that stores red, green, blue and (c) a circle object of radius 3, 2.9 and 5.9. Invoke the `max` method as below:

   ```
   public static <E extends Comparable<E>> E max(E[] list)
   ```

The max method above returns the maximum value in an array.

6. In a class called `MinMaxTwoDArray`, write two generic methods:
   a. First method returns the minimum element in a two-dimensional array. Below is the method signature:

      ```
      public static <E extends Comparable<E>> E min(E[][] list)
      ```

   b. Second method returns the maximum element in a two-dimensional array. Below is the method signature:

      ```
      public static <E extends Comparable<E>> E max(E[][] list)
      ```

   c. Create a test program that creates one instance of generic class of type Integer called `numbers` with the elements: {4, 5, 6}, {1, 2, 3}. Display the minimum and maximum elements using the `min` and `max` methods.