Solutions

Quiz 4 - Midtermish

Study Guide / Practice Problems

Topics:

- Inheritance (child classes and overriding)
    - Modules:
        - 09, 12, 13, 14, 15 (practice problems)
    - You should know:
        - How to override equals method multiple times
        - How to override hashCode / toString multiple times
        - How to make a child class extend a parent
- Hashcode
    - Modules:
        - 12, 13
    - You should know:
        - How to override hashCode long way
        - How to override hashCode using Objects.hash
- Old Quizzes / Study guides

Study Material:

- Materials listed above, lab 5 part 2, old study guides.

Practice problems

Overriding Equals, HashCode, toString

1. Inheritance

Given the following classes, fill in the TODOs.

```
public class Contestant
{
   private int airtime; // airtime, in minutes

   public Contestant(int airtime) {this.airtime = airtime;}

   public String toString() { return “Airtime (min): " + airtime;
}

   public boolean equals(  Object o      )
   {
        if (o == null) {return false;}
        if (o.getClass() != this.getClass()) {return false;}
        Contestant c = (Contestant) o;
        return this.airtime == c.airtime;
   }
}
```

```
public class Bachelor extends Contestant
{
   private int rosesLeft;

    /* TODO: Complete the constructor below.  Fill in the
   *     parameter and write the code*/
   public Bachelor(  int airtime, int rosesLeft)
   {
     super(airtime);
     this.rosesLeft = rosesLeft;
   }
   /* TODO: Complete the toString method such that it would
             return a String in the following format:

             Roses left: <rosesLeft>
             Airtime (min): <airtime>

             For example:

             "Roses left: 5
         Airtime (min): 500"
   */
   public String toString()
   {
      return “Roses left: “ + rosesLeft + “\n” + super.toString();
   }


   /* TODO:) Complete the equals method below.  Fill in the
   *    parameter and write the code*/
   public boolean equals(Object o  )
   {
         return super.equals(o) && ((Bachelor)o).rosesLeft ==
rosesLeft;
   }
}
```

2. Overriding hashcode

a. Override the hashCode method for the Student class by hand (assume a Student class has two variables - a double gpa, and a int named studentLoans)

```
    public int hashCode()
    {
        int hash = 1;

        hash = hash * 31 + studentLoans; //can leave since int
       //cannot add double to int
        hash = hash * 31 + ((Double)gpa).hashCode();
        return hash;
    }
```

b. Override the hashCode method for the Student class using any of the Objects methods you want:

```
    public int hashCode()
    {
        Return Objects.hash(studentLoans, gpa);
    }
```