CS110 Assignment #4
Due Wednesday, February 18th

This assignment consists of three programs that use user-defined classes. For the first one, I provide the class and you write the program that utilizes that class. For the second program, I provide the class design in the form of a UML diagram; you complete the class and the program. For the third, you design the class and use it in your program.

Be sure that all code is well-documented. For user-defined classes, comment for Javadocs.

1. At this end of this assignment, you'll find a definition of the `Die` class (a slight modification of the one developed in chapter 6). Using this definition, complete Programming Challenge #8.

Sample Runs:

(user chooses to stop after 2 rolls)

```
Roll the dice? (y/n) : y
Points: 4
Roll the dice? (y/n) : n
Game Over
User's Points: 4
Computer's Points: 8
The computer wins!
```

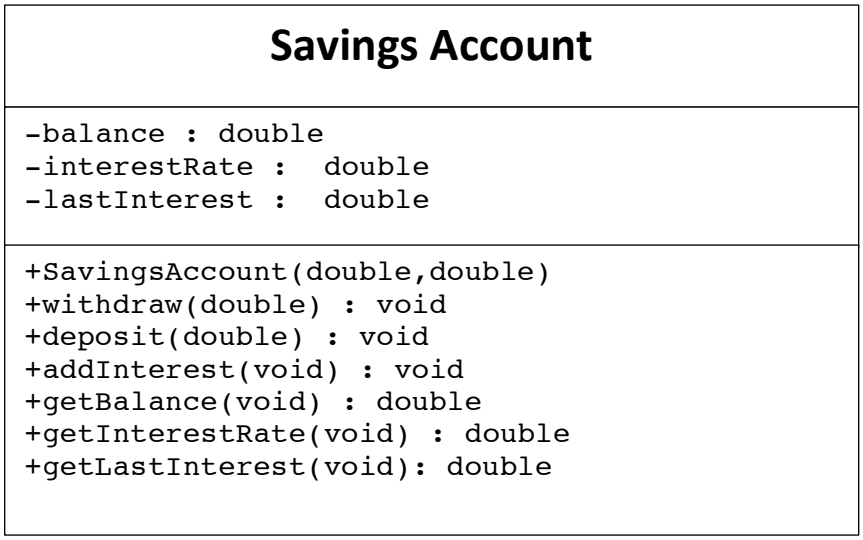
(The game executes to completion)

```
Roll the dice? (y/n) : y
Points: 5
Roll the dice? (y/n) : y
Points: 10
Roll the dice? (y/n) : y
Points: 14
Roll the dice? (y/n) : y
Points: 20
Roll the dice? (y/n) : y
Points: 23
Game Over
User's Points: 23
Computer's Points: 19
The computer wins!
```

(user chooses to stop before any rolls)

```
Roll the dice? (y/n) : n
Goodbye!
```

2. Complete Programming Challenge #11. This requires you to complete Programming Challenge #10. Submit your `SavingsAccount` class as well as a class with the main method that has the main method. Implement your `SavingsAccount` class according to the UML diagram:



3. Complete Programming Challenge #13. This requires you to complete Programming Challenge #12. You are free to design your own Coin class to best suit your needs. You will be graded on design decisions.

Sample Runs:

```
Balance: $1.00
You win.
```

```
Balance: $1.15
You did not win.
```

```java
import java.util.Random;

/**
   Represents one die with faces showing values between 1 and the
   number of faces on the die.
*/

public class Die
{
   private static int MIN_SIDES = 4; // no less than 4 sides on a die
   private int sides   ;   // The number of sides
   private int value;            // The value of the die
   private Random rand;          // Random number generator

   /**
      The Default constructor calls the roll method
      creates the Random object, sets number of sides to 6 and
      sets       the intitial value of the die to a random
      number.
    */

   public Die()
   {
      // create Random object
      rand = new Random();
      // set number of side on die
      sides = 6;
      // Call the roll method to randomly
      // set the value of the die.
      roll();
   }
   /**
      Alternate constructor calls the roll allows user to specify
      number of side son die.  The method creates the Random object,
      sets number of sides to value user specified and sets the intitial
      value of the die to a random number.
      @param numSides The number of sides on the die
   */

    public Die(int numSides)
    {
      // create Random object
      rand = new Random();
      // set number of side on die
      sides = numSides;
      // Call the roll method to randomly
      // set the value of the die.
      roll();
    }


    /**
      The roll method sets the value of the die
      to a random number.
     */
```

```
    public void roll()
    {

      // Set the value to a random number.
      value = rand.nextInt(sides) + 1;
    }

    /**
       The getValue method returns the value
      of the die.
    */

    public int getValue()
    {
      return value;
    }
}
```