CSIS 10B Assignment 1 SOLUTIONS

1) CodingBat problems

```
   /**
   * @author Emerson Christie
   * Copy all of your coding bat methods into this class
   *
   */
   public class CodingBat
   {
   // just copy / paste your methods from coding bat
   // you don't need to demo it, I just want to see
   // your code.
   public String withoutEnd(String str) {
   String newStr = "";
   for(int k = 1; k < str.length()-1; k++)
   newStr = newStr + str.charAt(k);
   return newStr;
   }

   public String withouEnd2(String str) {
   String newStr = "";
   for(int k = 1; k < str.length()-1; k++)
   newStr = newStr + str.charAt(k);
   return newStr;
   }

   public String middleTwo(String str) {
   String newStr = "";
   newStr = newStr + str.charAt((str.length()/2)-1) + str.charAt(str.length()/2);
   return newStr;
   }

   public int[] makeEnds(int[] nums) {
   int[] result = new int[] {nums[0],nums[nums.length-1]};
   return result;
   }

   public boolean sameFirstLast(int[] nums) {
   return (nums.length>0 && nums [0] == nums [nums.length - 1]);
   }

   public boolean double23(int[] nums) {
   if (nums.length == 2)
   return ((nums[0] == 2 && nums[1] == 2) || (nums[0] == 3 && nums[1] == 3 ));
   else
   return false;
   }

}
```

2) GeneralCoin.java

```
import java.util.Random;
/**
 * class GeneralCoin represents a coin that can be flipped.
 *
 * @author David Meyer
 * @version 02/22/13
 */
```

```
public class GeneralCoin
{
    private String sideUp;
    private Random gen;

    /**
     * Constructor for objects of class GeneralCoin
     */
    public GeneralCoin()
    {
        // the random # generator
        gen = new Random();

        // initial toss
        toss();
    }

    /**
     * Setter for String sideUp
     * @param side The string to set sideUp to. Either "Heads" or "Tails"
     */
    public void setSideUp(String side)
    {
        sideUp = side;
    }

    /**
     * Getter for String sideUP
     * @return sideUp Either "Heads" or "Tails"
     */
    public String getSideUp()
    {
        return sideUp;
    }

    /**
     * This method tosses the "coin" and sets sideUp to the corresponding result.
     */
    public void toss()
    {
        if(gen.nextInt(2) == 0) //gen to "flip the coin"
            setSideUp("Heads"); //if 0 heads
        else //then must be tails
            setSideUp("Tails");
    }
}
```

Coin.java

```
/**
 * class Coin inherits from GenericCoin
 * @author David Meyer
 * @version 02/22/13
 */
public class Coin extends GeneralCoin
{   // complete the definitions using inheritance to extend the GenericCoin class
    private int value;
    private String name;

    /**
```

```
     * no-args coin constructor
     */
    public Coin()
    {
        super();  // invoke the GeneralCoin constructor
        value = 0;
        name = "----";
    }

    /**
     * Constructor w/ args.
     */
    public Coin(int value, String name)
    {
        super();
        this.value = value;
        this.name = name;
    }

    /**
     * Sets the value of the coin.
     */
    public void setValue(int newValue)
    {
        value = newValue;
    }

    /**
     * Returns the value of the coin.
     */
    public int getValue()
    {
        return value;
    }

    /**
     * Sets the name of the coin
     */
    public void setName(String newName)
    {
        name = newName;
    }

    /**
     * Returns the name of the coin.
     */
    public String getName()
    {
        return name;
    }
}
```

CoinApp.java

```
/**
 * Write a description of class CoinApp here.
 *
 * @author David Meyer
 * @version 02/22/13
 */
public class CoinApp
{
    public static void main(String [] args)
    {
        /******************** Prob 2b) ***************************************
        // Write a program that tosses two GeneralCoin coins 50 times each.
        // Record and report how many times each coin lands heads up.
        // Also report which coin landed heads up most often.

        GeneralCoin a = new GeneralCoin();
        GeneralCoin b = new GeneralCoin();

        a.toss();
        b.toss();

        System.out.println("Coin A: " + a.getSideUp() + ", Coin B: " + b.getSideUp());

        //initialize the counters
        int countHeadsA = 0;
        int countHeadsB = 0;

        // Now add statements to toss both coins 50 times
        System.out.println("Flipping GeneralCoin 50 times");
        for(int i = 0; i < 50; i++)
        {
            //toss the coins
            a.toss();
            b.toss();

            //check if either landed heads up, if true then increment count
            if(a.getSideUp().equals("Heads"))
                countHeadsA++;
            // *
            if(b.getSideUp().equals("Heads"))
                countHeadsB++;
        }

        //report who landed heads up most often.
        System.out.print("Coin A landed " + countHeadsA + " times up");
        System.out.print(" while coin B landed " + countHeadsB + " times up.");
        System.out.println("\nCoin " + ((countHeadsA > countHeadsB) ? "A" : "B")
                                     + " landed heads up most often.");

        //******************** END Prob 2b) ****************************************/
        /******************** Prob 2D) ****************************************
        // Write a small program that tests your Coin class.
        // (Essentially, repeating the steps from 2B but with Coin objects)
        // Show that your new coins inherit the behaviors of a GeneralCoin object.
        // (i.e. they can also be tossed)

        // Here is an example with one coin
        Coin a = new Coin();
        a.setValue(25);
        a.setName("Quarter");
        a.toss();
        System.out.println("Coin a is a " + a.getName());
        System.out.println("Coin a value is " + a.getValue());
        System.out.println("Coin a side up is " + a.getSideUp());

        //create another test coin
        Coin b = new Coin();
        b.setValue(25);
        b.setName("Quarter");

        // let's toss the coin objects 50 times
        System.out.println("Flipping Coin 50 times");
        //initialize the counters
        int countHeadsA = 0;
        int countHeadsB = 0;
        for(int i = 0; i < 50; i++)
        {
            //toss the coins
            a.toss();
            b.toss();

            //check if either landed heads up, if true then increment count
            if(a.getSideUp().equals("Heads"))
                countHeadsA++;
            // *
            if(b.getSideUp().equals("Heads"))
                countHeadsB++;
        }

        //report who landed heads up most often.
        System.out.println("Coin " + ((countHeadsA > countHeadsB) ? "A" : "B")
                                   + " landed heads up most often.");
        //end coin object test

        /******************** END Prob 2D) ****************************************/
        //******************** Prob 2E) ****************************************
        // Write your coin tossing application here after reading handout step 2E

        Coin theCoin = new Coin();   // we can reuse this one variable for all coins
        ArrayBag<Coin> headCoins = new ArrayBag<Coin>(40);
        Coin[] tailCoins = new Coin[40];

        //since tailCoins could be a partially filled array
        int tailSize = -1; //-1 since the first index is 0

        // use either an array of Coin or an ArrayBag<Coin> for the tailCoins
        for(int i = 1; i <= 40; i++)
        {
            // toss the coin
            theCoin.toss();

            //assign what kind of coin it is
            if(i <= 10)
            {
                theCoin.setName("Penny");
                theCoin.setValue(1);
            }
            else if(i >= 11 && i <= 20)
            {
                theCoin.setName("Nickel");
                theCoin.setValue(5);
            }
            else if(i >= 21 && i <= 30)
            {
                theCoin.setName("Dime");
                theCoin.setValue(10);
            }
            else
            {
                theCoin.setName("Quarter");
                theCoin.setValue(25);
            }

            //add the coin to either the bag or array
            if(theCoin.getSideUp().equals("Heads"))
            {
                headCoins.add(new Coin(theCoin.getValue(), theCoin.getName()));
```

```java
            }
            else
            {
                tailCoins[tailSize+1] = new Coin(theCoin.getValue(), theCoin.getName());
                tailSize++;
            }
        }

        //value counters for heads and tails
        int headValue = 0;
        int tailValue = 0;

        //remove all the coins from headCoins and tally the value
        while(!headCoins.isEmpty())
            headValue += headCoins.remove().getValue();

        //tally the value of the tailCoins
        for(int i = 0; i <= tailSize; i++)
            tailValue += tailCoins[i].getValue();

        // add up the two totals and make sure they equal 410 cents!!
        System.out.println("Coins heads up total value: " + headValue + " cents");
        System.out.println("Coins tails up total value: " + tailValue + " cents");
        System.out.println("For a total of " + (headValue + tailValue) + " cents");
        //******************** END Prob 2E) ****************************************/
    }
}
```