**S11 CS-401 Lab #7: Reverse Engineering the Coin class**

**Lab #7 will be script graded. You have till Saturday 7/14 midnight to handin Lab #7 via the online handin system.**

**You are given the main program and the output produced by it. You must write the Coin class defintion file named Coin.java which will satisfy the class definition required by CoinTester.java. Be sure to write your Coin.java in the same directory with CoinTester.java**

**Look at the main and see how the coin class is constructed and what methods are called. From those observations, you can deduce the private data members and  public methods with their exact naming and data type. Note that when a coin object is created, it's internal head and tail counts are initialized to zero. Each time a coin is .flip()'d that flip increments that coin's heads or tails counts with a 50% each probability. Thus, you must use a Random variable inside your Coin class to get a 50/50 chance of getting a head or tail. Once the flip method decided that a head or tail was rolled, the flip method increments the appropriate counter and then returns "H" or "T". The reset() method starts that coin's counters back at 0.**

**Start with**[**CoinTester.java**](http://www.cs.pitt.edu/~hoffmant/401/lab-07/CoinTester.java)***(warning: some browsers will choke displaying the angle brackets in the Java code)***

|  |
| --- |
| /\*  CoinTester.java - tests the Coin class by  constructing variables and calling it's methods  \*/  public class CoinTester  {  public static void main( String args[] )  {  Coin coin1 = new Coin();  Coin coin2 = new Coin();  // FLIP COIN1, PRINT RESULTS  System.out.println("\nFlipping Coin1 20 times.");  for (int i=0 ; i<20 ; ++i)  System.out.print( coin1.flip() + " " ); // Equal chance of head or tail  System.out.println();  System.out.println("heads=" + coin1.getNumHeads() +  ", tails=" + coin1.getNumTails() );  coin1.reset(); // sets numHeads and numTails back to zero;  // FLIP COIN2, PRINT RESULTS  System.out.println("\nFlipping Coin2 10 times.");  for (int i=0 ; i<10 ; ++i)  System.out.print( coin2.flip() + " " ); // Equal chance of head or tail  System.out.println();  System.out.println("heads=" + coin2.getNumHeads() +  ", tails=" + coin2.getNumTails() );  coin2.reset(); // sets numHeads and numTails back to zero;  // FLIP COIN2 AGAIN, PRINT RESULTS  System.out.println("\nFlipping Coin1 again 35 times.");  for (int i=0 ; i<35 ; ++i)  System.out.print( coin1.flip() + " " ); // Equal chance of head or tail  System.out.println();  System.out.println("heads=" + coin1.getNumHeads() +  ", tails=" + coin1.getNumTails() );  coin1.reset(); // sets numHeads and numTails back to zero;  }// END main  }//EOF |

**I SEEDED the Random object in my Coin.java class definition file with the number 17. As a result, this program and yours should give IDENTICAL results every time. You should see approximately equal numbers of heads an tails if the coin is flipped many times.**

**Random r = new Random(17); // guarantees same random numbers each run of program**

**Here is the output of a sample run of CoinTester.java**

|  |
| --- |
| Flipping Coin1 20 times.  T T T T H H T T H T H T T H T H T T H T  heads=7, tails=13  Flipping Coin2 10 times.  T T T T H H T T H T  heads=3, tails=7  Flipping Coin1 again 35 times.  H H T H T T T H H H T H T H T H T T H H H H T T T T T T T T H H H H H  heads=18, tails=17 |