

\*\*\*\*\* OUTPUT \*\*\*\*\*

UNCOMPILED APPLICATION – ALL AREAS CAUSING ERRORS OUTLINED  
IN THE CODE  
ALSO CREATED PSEUDO CODE TO OUTLINE MY INTENDED LOGIC

# CODE

```
/*
Online Java - IDE, Code Editor, Compiler
Online Java is a quick and easy tool that helps you to build, compile, test your programs online.
*/
import java.util.Random;
import java.util.*;
public class Main
{
public static void main(String[] args)
{
System.out.println("Program to test flipping coins multiple times");
int n=4; // this is number of coins
Flipcoin fc= new Flipcoin(n); // ****instantiate Flipcoin. PROBLEM OCCURS HERE****

System.out.println("Number of rounds is: " + fc.getRounds());
}
}
class Flipcoin
{
private int numberCoins;
int[] coins = new int[numberCoins]; //array for number coins
Flipcoin fc;
int exections=0;
public Flipcoin(int numberCoins)
{
this.numberCoins=numberCoins;
System.out.println("number coins right now is:" + numberCoins);
Flipcoin fc;
int i;
int heads=0;
int tails=0;
for (i=0; i<numberCoins;i++) // this will process each coins
{
Random random = new Random();
int rand = random.nextInt(2) + 1; // This will generate numbers 1 & 2.
```

```

// 1 WILL BE TAKEN TO BE HEADS
// 2 WILL BE TAKEN TO BE TAILS
coins[i]=rand; //PROBLEM OCCURS HERE AND ANY REFERENCES USING VARIABLE I FAILS
System.out.println(rand); //SINCE IT IS LINKED TO NUMBERCOINS VARIABLE

if (coins[i]==1) //ALL HEADS ARE COUNTED
{
System.out.println ("heads");
heads++;
}
if (coins[i]==2) //ALL TAILS ARE COUNTED
{
System.out.println("tails");
tails++;
}
}
if (tails==numberCoins) // THIS IS OUT OF THE FOR LOOP ONCE ALL COINS ARE TOSSED. IT WILL
CHECK IF ALL ARE TAILS
{ // THIS IS A VOID FLIP SINCE IT WILL ELIMINATE ALL COINS.
//CODE REQUIRES 1 COIN REMAINING. SO A RE-FLIP OF ALL COINS IS UNDERTAKEN
executions++;
flipCoinsAgain(numberCoins,executions);
}

do // THIS ENTIRE SECTION OF CODE HAS NOT BE TESTED SINCE THERE IS AN ERROR
{ // IN PASSING NUMBER COINS TO THE CONSTRUCTOR
if (tails>=1)
{
numberCoins=numberCoins-tails; //THIS WILL REDUCE NUMBER COINS BY NUMBER TAILS IF A
TAIL IS TOSSED
//flipCoinsAgain(numberCoins);
}
if (heads>=1)
{
flipCoinsAgain(numberCoins); // ANY HEADS THAT ARE TOSSED WILL TRIGGER COINS TOSSED
AGAIN WITH REVISED NUMBER COINS
executions++; //THIS INCREMENTS SINCE ALL APPLICABLE COINS ARE TOSSED AGAIN.
}
}while((numberCoins!=1) && (heads==1));
//THE WHOLE CYCLE WILL CONTINUE UNTIL 1 COIN IS LEFT WHICH IS HEADS.
// IT IS NOT KNOWN IF THE PROBLEM SUGGESTS THAT LAST COIN LEFT CAN BE EITHER HEADS
OR TAILS
// IF IT CAN BE EITHER, THEN WHILE STATEMENT IS AS FOLLOWS:
//while((numberCoins==1)

```

```
}  
void flipCoinsAgain(int numberCoins, int executions)  
{  
    System.out.println("new value of n:" + numberCoins);  
    new Flipcoin(numberCoins); //THIS WILL INVOKE FLIP OF REMAINING COINS  
    //return executions;  
}  
int getRounds()  
{  
    return executions;  
}  
}
```