WEEK 11

EXTRA PROGRAMS

/\*Write a program to create a thread and find the sum of odd numbers from 1 to 100 in

this thread. Find the sum of even numbers for the same range in the main thread.\*/

class Threads implements Runnable

{

Thread t;

int sum=0;

Threads()

{

t=new Thread(this, "OddThread");

System.out.println("THREAD:"+t);

t.start();

}

public void run()

{

try

{

for(int n=1;n<=100;n=n+2)

{

System.out.print("ODD THREAD: "+n+" ");

sum=sum+n;

Thread.sleep(100);

}

}

catch(InterruptedException ie)

{

System.out.println("ODD THREAD Interrupted");

}

System.out.println("ODD THREAD quitting");

System.out.println("SUM OF ODD NUMBERS="+sum);

}

}

class threadmain

{

public static void main(String ss[])

{

int sum=0;

Threads n1=new Threads();

try

{

for(int n=2;n<=100;n=n+2)

{

System.out.print("EVEN Thread: "+n+" ");

sum=sum+n;

Thread.sleep(100);

}

}

catch(InterruptedException ie)

{

System.out.println("EVEN Thread interrupted");

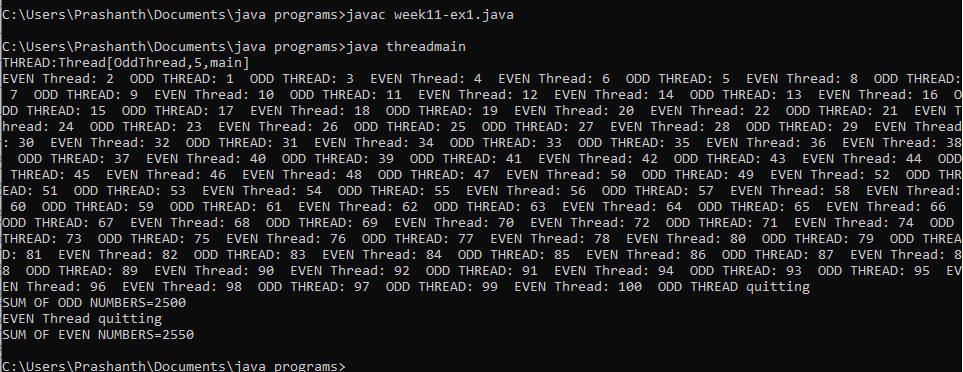
}

System.out.println("EVEN Thread quitting");

System.out.println("SUM OF EVEN NUMBERS="+sum);

}

}



/\*Develop a multithreaded Java program to create three threads. First thread generates

random integer for every second and if the value is even, second thread computes the

square of number and prints. If the value is odd, the third thread will print the value of

cube of number.\*/

import java.util.Random;

class Square extends Thread

{

int num;

Square(int n)

{

num = n;

}

public void run()

{

System.out.println("Square of " + num + " = " + (num\*num) );

}

}

class Cube extends Thread

{

int num;

Cube(int n)

{

num = n;

}

public void run()

{

System.out.println("Cube of " + num + " = " + (num\*num\*num) );

}

}

class Number extends Thread

{

public void run()

{

Random rn = new Random();

for(int i =0; i<10; i++)

{

int random\_value = rn.nextInt(100);

System.out.println("Random Integer generated : " + random\_value);

if(random\_value%2==0)

{

Square s = new Square(random\_value);

s.start();

}

else

{

Cube c = new Cube(random\_value);

c.start();

}

try {

Thread.sleep(1000);

}

catch (InterruptedException e) {

System.out.println("interrupted");

}

}

}

}

class Main

{

public static void main(String args[])

{

Number n = new Number();

n.start();

}

}

