/\*Assignment based on Collections and Multithreading in Java\*/

Program1-

import java.util.\*;

public class ConcurrentDemo extends Thread {

static ArrayList l = new ArrayList();

public void run()

{

try {

Thread.sleep(2000);

}

catch (InterruptedException e) {

System.out.println("Child Thread" +

" going to add element");

}

l.add("D");

}

public static void main(String[] args)

throws InterruptedException

{

l.add("A");

l.add("B");

l.add("c");

ConcurrentDemo t = new ConcurrentDemo();

t.start();

Iterator itr = l.iterator();

while (itr.hasNext()) {

String s = (String)itr.next();

System.out.println(s);

Thread.sleep(6000);

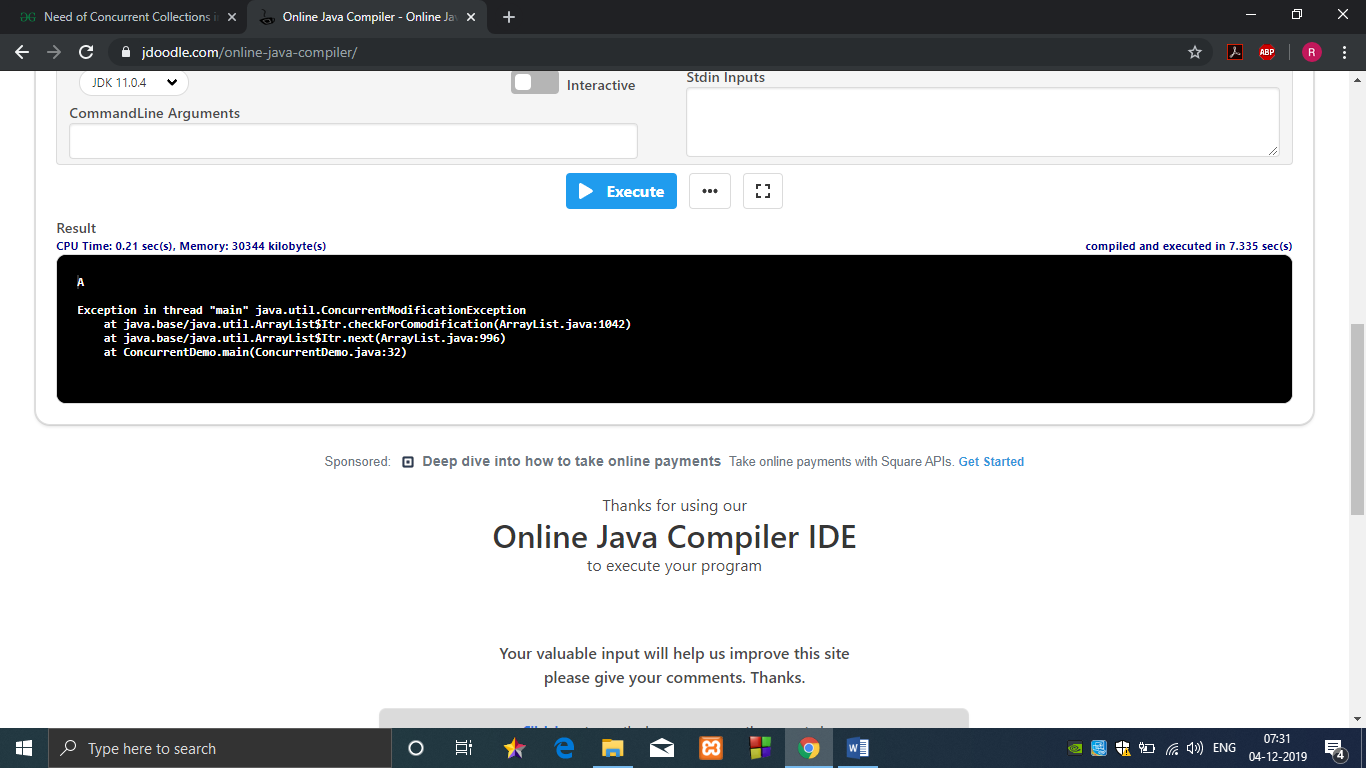
}

System.out.println(l);

}

}

Output-



Program 2-

class MultithreadingDemo extends Thread

{

public void run()

{

try

{

// Displaying the thread that is running

System.out.println ("Thread " +

Thread.currentThread().getId() +

" is running");

}

catch (Exception e)

{

// Throwing an exception

System.out.println ("Exception is caught");

}

}

}

public class Multithread

{

public static void main(String[] args)

{

int n = 8; // Number of threads

for (int i=0; i<8; i++)

{

MultithreadingDemo object = new MultithreadingDemo();

object.start();

}

}

}

Output-

