***Ripunjay Narula 19BCE0470***

***Java Lab Digital Assignment***

**Packages**:

1.package pack1;

public interface inter1{

public float add(float a, float b);

public float sub(float a, float b);

}

package pack2;

public interface inter2{

public float mul(float a, float b);

public float div(float a, float b);

}

import pack1.inter1;

import pack2.inter2;

import java.util.\*;

public class calc implements inter1,inter2{

public float add(float a, float b){

return (a+b);

}

public float sub(float a, float b){

return (a-b);

}

public float mul(float a, float b){

return (a\*b);

}

public float div(float a, float b){

if(b!=0){

return (a/b);

}

else{

System.out.println("Denominator cannot be 0");

return 0;

}

}

public static void main(){

calc obj=new calc();

System.out.println("Enter the 2 numbers:");

Scanner sc= new Scanner(System.in);

float a=sc.nextFloat();

float b= sc.nextFloat();

System.out.println(obj.add(a,b));

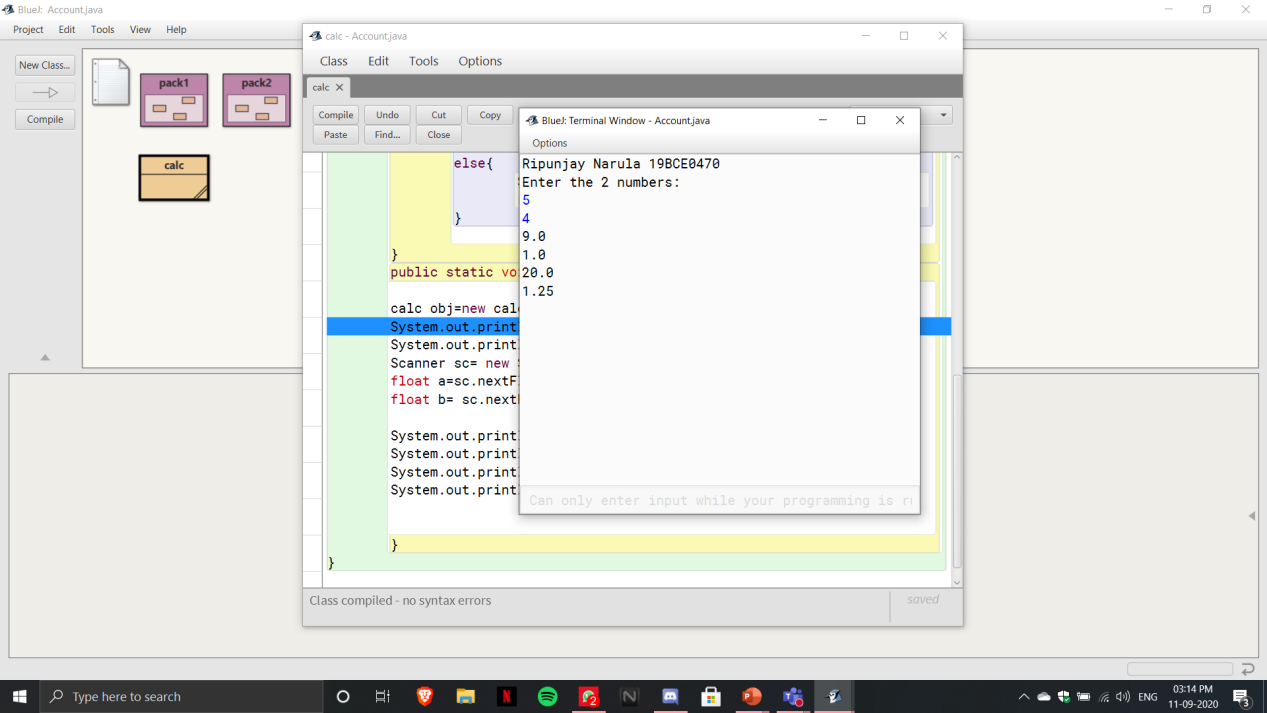
System.out.println(obj.sub(a,b));

System.out.println(obj.mul(a,b));

System.out.println(obj.div(a,b));

}

}



2.package primespackage;

public class Primes{

public boolean checkForPrime(int n){

int c=0,i;

for(i=1;i<n;i++)

{

if(n%i==0)

c++;

}

return (c==1);

}

}

import primespackage.Primes;

import java.util.\*;

public class TwinPrimes{

public static void main(){

Primes obj=new Primes();

for(int i=2;i<11;i++){

if (obj.checkForPrime(i) && obj.checkForPrime(i+2)){

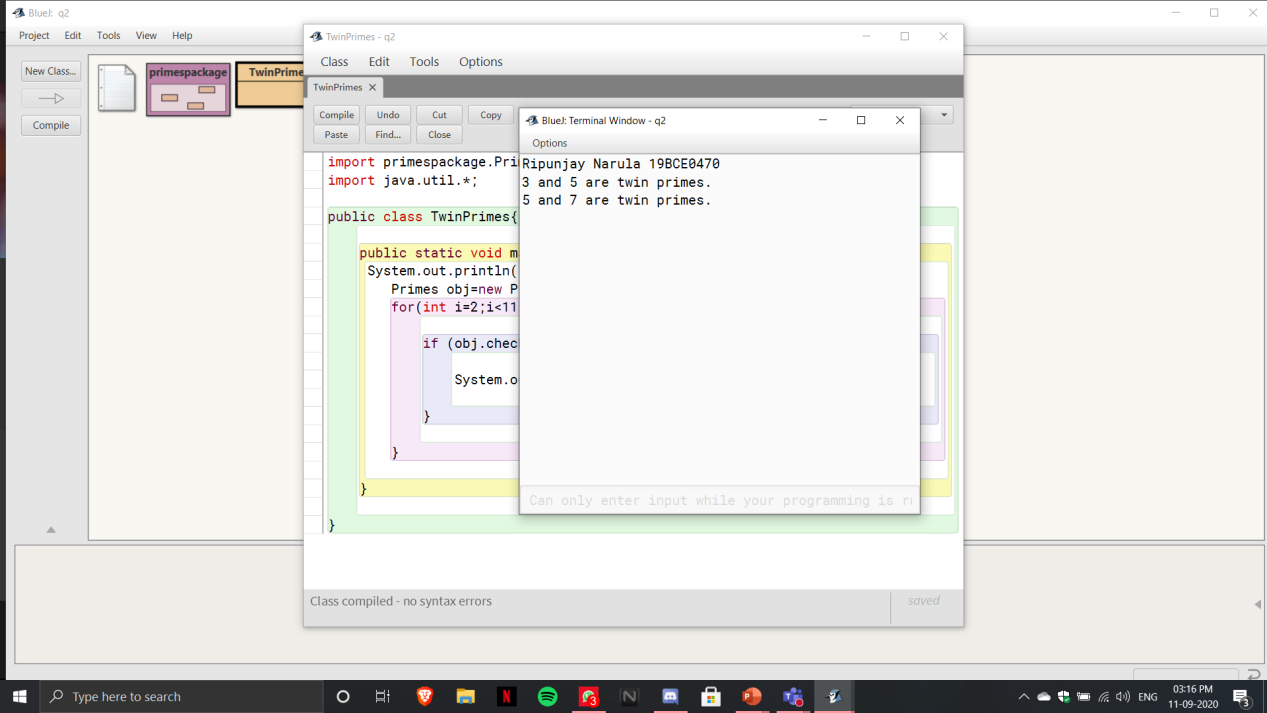
System.out.println(i+" and "+(i+2)+" are twin primes. ");

}

}

}

}



**Exception Handling:**

1.import java.util.Scanner;

class IllegalArgumentException extends Exception

{

IllegalArgumentException(String s)

{

super(s);

}

}

class NumberFormatException extends Exception

{

NumberFormatException(String s)

{

super(s);

}

}

class NoSuchElementException extends Exception

{

NoSuchElementException(String s)

{

super(s);

}

}

public class stud\_exception

{

String RegNo;

String Phn\_No;

public boolean onlyDigits(String str)

{

int count=0;

for (int i = 0; i < str.length(); i++)

{

if (!Character.isDigit(str.charAt(i)))

{

return false;

}

}

return true;

}

public void LengthCheck(String RegNo, String Phn\_No) throws IllegalArgumentException

{

if((this.RegNo.length()!=9)||(this.Phn\_No.length()!=10))

{

throw new IllegalArgumentException("Invalid");

}

else

{

System.out.println("Valid");

}

}

public void CheckNumber(String Phn\_No) throws NumberFormatException

{

if(!onlyDigits(this.Phn\_No))

{

throw new NumberFormatException("Invalid");

}

else

{

System.out.println("Valid");

}

}

public void RegNoPattern(String RegNo) throws NoSuchElementException

{

if(!RegNo.matches("[A-Za-z0-9]+"))

{

throw new NoSuchElementException("Invalid");

}

else

{

System.out.println("Valid");

}

}

stud\_exception(String RegNo, String Phn\_No)

{

this.RegNo =RegNo;

this.Phn\_No=Phn\_No;

try{

LengthCheck(this.RegNo,this.Phn\_No);

CheckNumber(this.Phn\_No);

RegNoPattern(this.RegNo);

}

catch(NoSuchElementException e)

{

System.out.println("Exception occured: "+e);

}

catch(NumberFormatException e)

{

System.out.println("Exception occured: "+e);

}

catch(IllegalArgumentException e)

{

System.out.println("Exception occured: "+e);

}

}

public static void main(String args[])

{

Scanner input= new Scanner(System.in);

System.out.println("Reg No:");

String RegNo= input.nextLine();

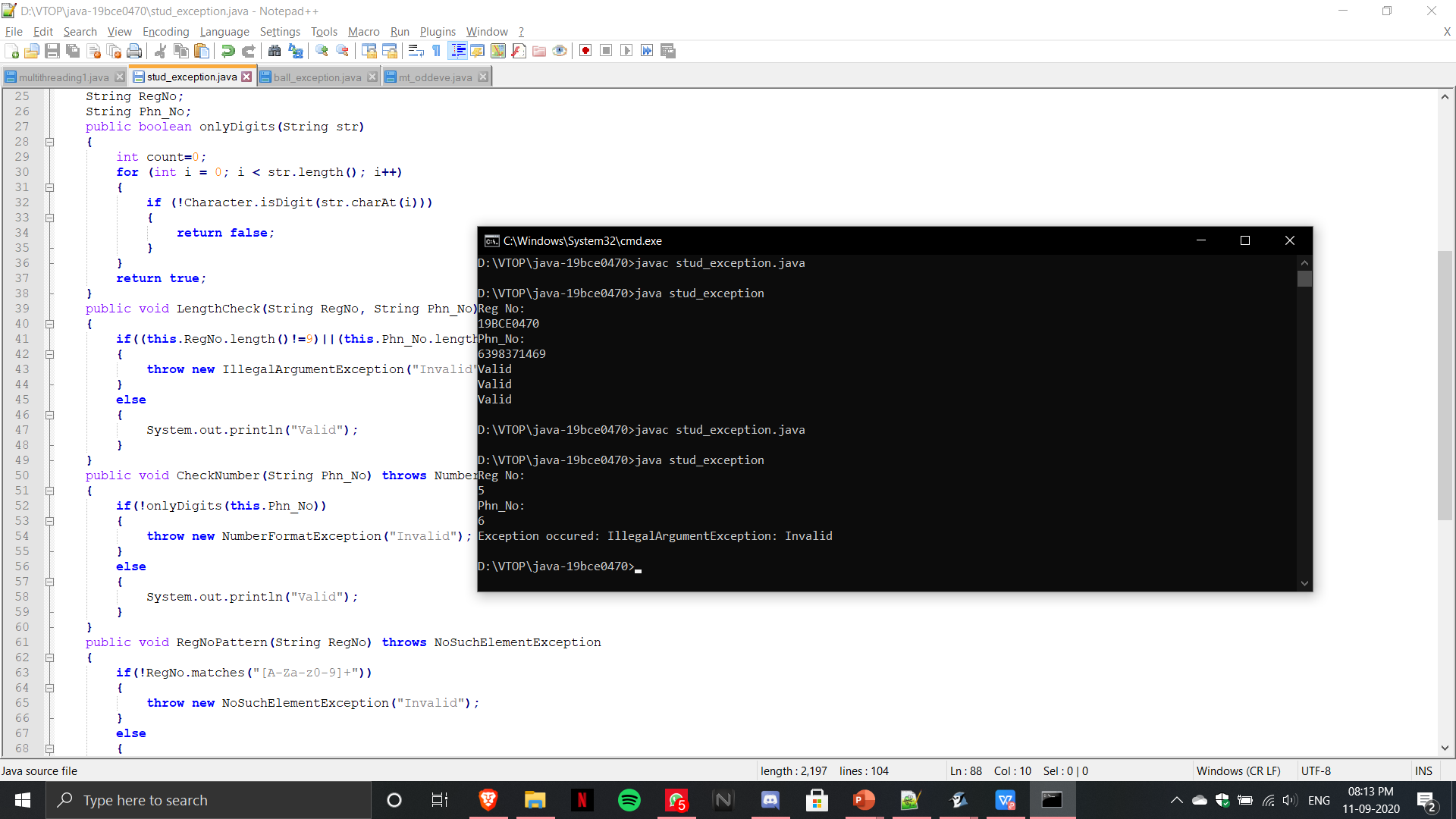
System.out.println("Phn\_No:");

String Phn\_No= input.nextLine();

stud\_exception s= new stud\_exception(RegNo,Phn\_No);

}

}



2.import java.util.Scanner;

class SameColorBallException extends Exception

{

SameColorBallException (String s)

{

super(s);

}

}

public class ball\_exception

{

public void RandomPickGenerator()

{

int count=0;

int arr[]=new int[10];

Scanner input=new Scanner(System.in);

while(count<10)

{

System.out.println("Enter ball red:0, green:1, blue:2, yellow:3");

int num=input.nextInt();

arr[count]=num;

try

{

if(count>2)

{

if((arr[count-3]==arr[count-2])&&(arr[count-2]==arr[count-1])&&(arr[count-1]==arr[count]))

{

throw new SameColorBallException("Invalid");

}

else

{

count++;

}

}

else

{

count++;

}

}

catch(SameColorBallException e)

{

System.out.println("Do not enter the same colored ball more than thrice ");

}

}

int r=0,b=0,y=0,g=0;

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

{

if(arr[i]==0)

{

r++;

}

else if(arr[i]==1)

{

g++;

}

else if(arr[i]==2)

{

b++;

}

else

{

y++;

}

}

System.out.println("Red balls: "+r);

System.out.println("Blue balls: "+b);

System.out.println("Green balls: "+g);

System.out.println("Yellow balls: "+y);

}

public static void main(String args[])

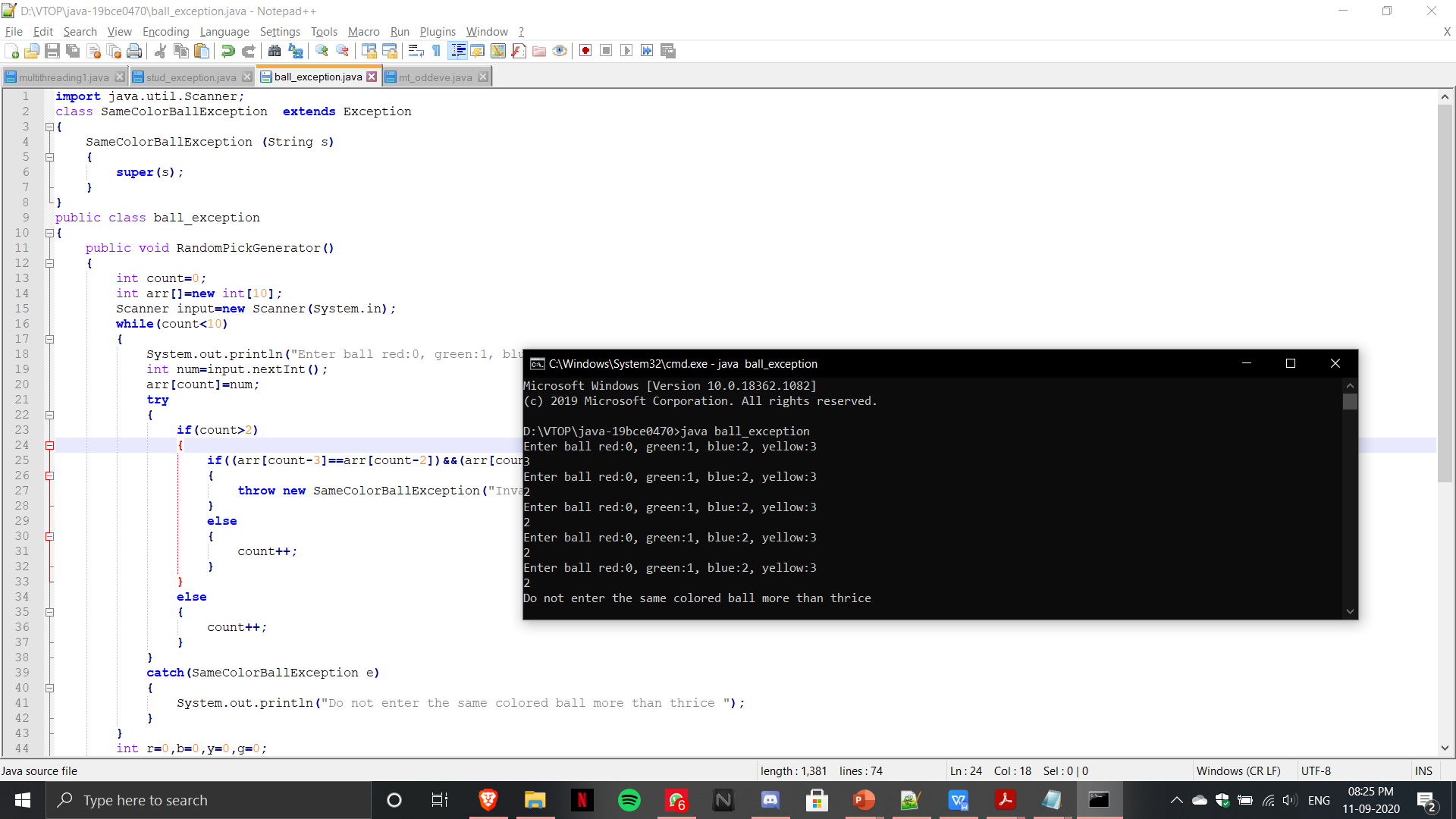
{

ball\_exception r= new ball\_exception();

r.RandomPickGenerator();

}

}



**Multithreading**:

1.import java.util.Scanner;

class EvenThread extends Thread

{

int n;

EvenThread(int n)

{

this.n=n;

}

public void run()

{

System.out.println("Even Nubers are: ");

for(int i=1;i<=n;i++)

{

if(i%2==0)

{

System.out.println(i);

}

}

}

}

class OddThread extends Thread

{

int n;

OddThread(int n)

{

this.n=n;

}

public void run()

{

System.out.println("Odd Numbers are: ");

for(int i=1;i<=n;i++)

{

if(i%2!=0)

{

System.out.println(i);

}

}

}

}

public class mt\_oddeve

{

public static void main(String arg[])

{

Scanner input=new Scanner(System.in);

System.out.println("Enter Range: ");

int n=input.nextInt();

EvenThread e= new EvenThread(n);

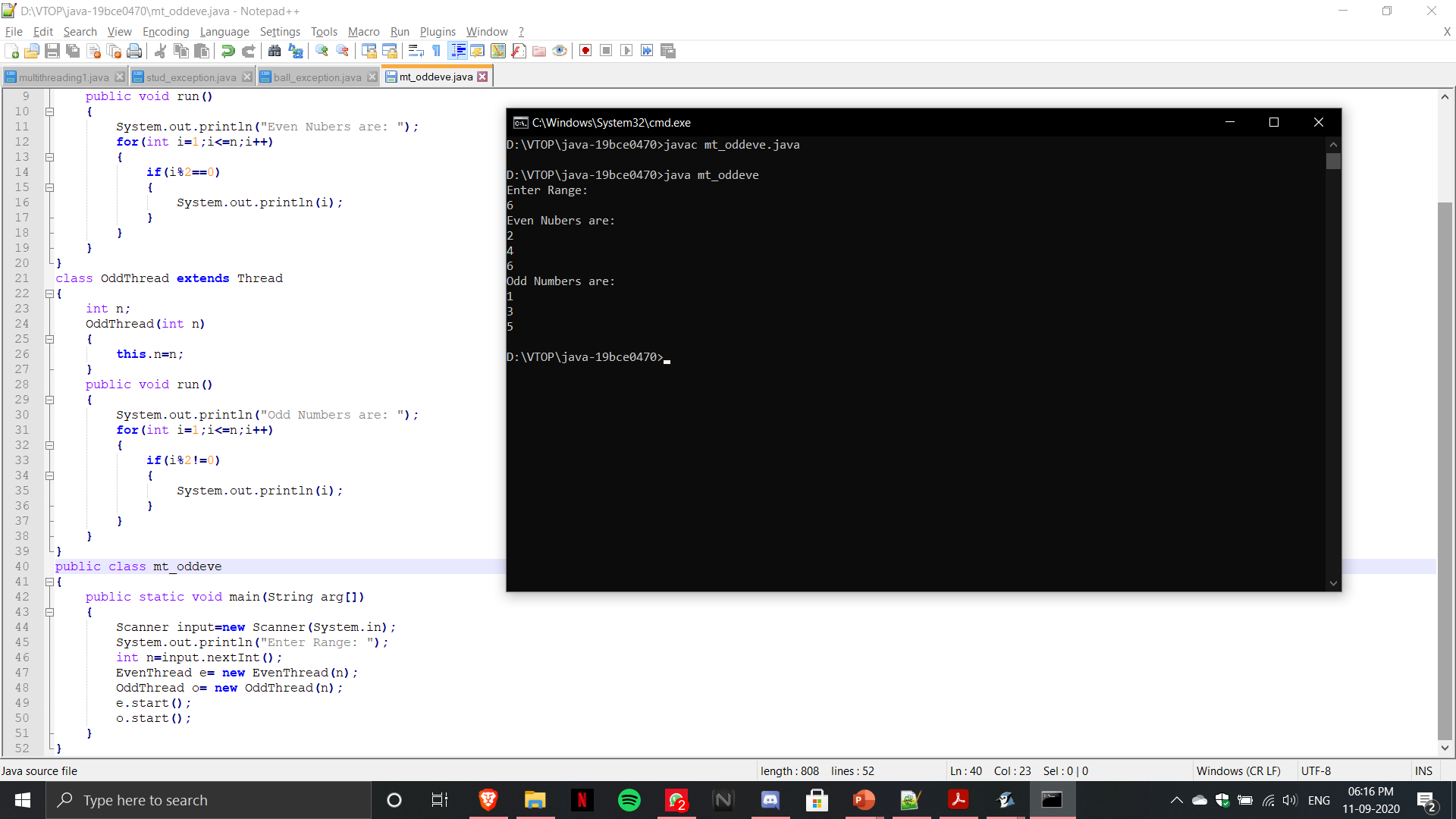
OddThread o= new OddThread(n);

e.start();

o.start();

}

}



2.import java.util.\*;

import java.awt.\*;

class One implements Runnable

{

One()

{

new Thread(this,"one").start();

}

public void run()

{

for(int i=0;i<6;i++)

{

try

{

Thread.sleep(1000);

}

catch(InterruptedException e)

{

System.out.println("Hello");

}

System.out.println("Hello");

}

}

}

class Two implements Runnable

{

Two(){

new Thread(this,"two").start();

}

public void run()

{

for(int i=0;i<6;i++)

{

try

{

Thread.sleep(3000);

}

catch(InterruptedException e)

{

System.out.println("Welcome to VIT");

}

System.out.println("Welcome to VIT");

}

}

}

public class multithreading1

{

public static void main(String args[])

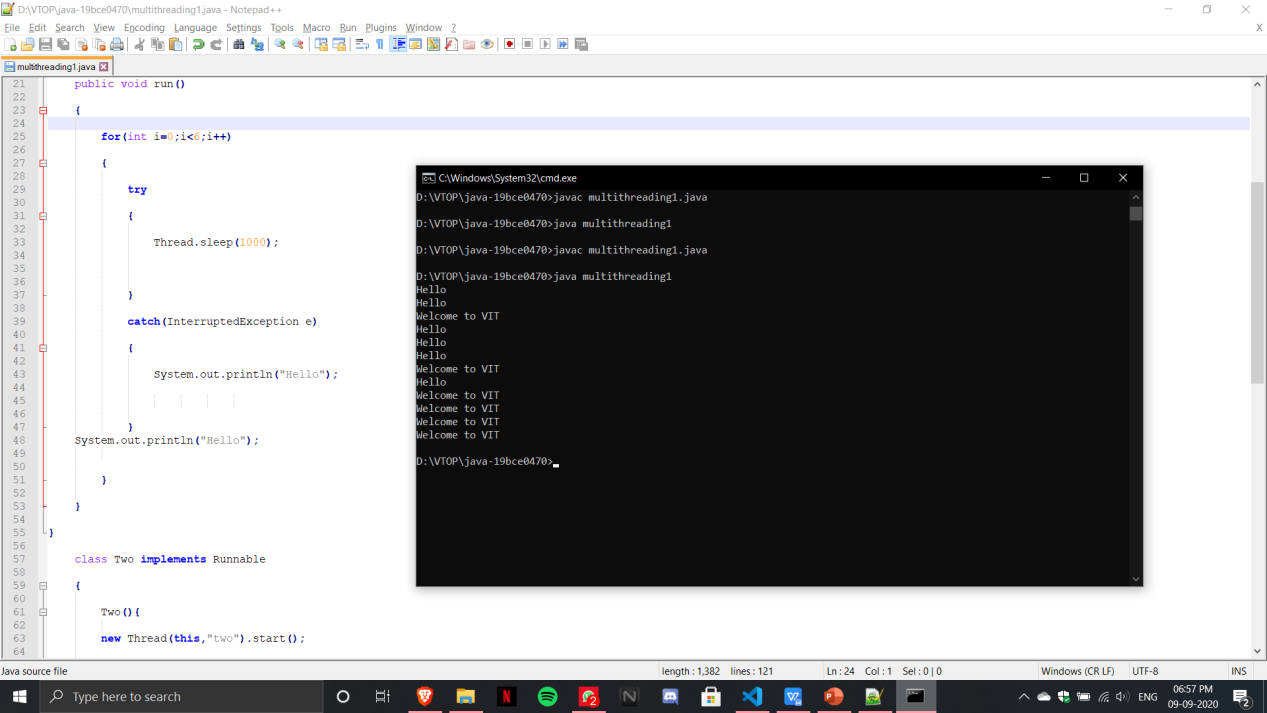
{

One o1 = new One();

Two t1 = new Two();

}

}



3.import java.util.Random;

class Vote {

int[] arr;

int A\_vote;

int B\_vote;

int C\_vote;

public void GenerateVotes() {

Random r = new Random();

this.arr = new int[240];

for (int i = 0; i < 240; i++) {

int a = r.nextInt(3);

this.arr[i] = a + 1;

}

this.A\_vote = 0;

this.B\_vote = 0;

this.C\_vote = 0;

}

synchronized public void VoteCount(int start, int fin) {

int A = 0, B = 0, C = 0;

for (int i = start; i < fin; i++) {

if (this.arr[i] == 1) {

A++;

} else if (this.arr[i] == 2) {

B++;

} else {

C++;

}

}

this.A\_vote += A;

this.B\_vote += B;

this.C\_vote += C;

}

}

class ThreadA extends Thread {

Vote v;

ThreadA(Vote v) {

this.v = v;

}

public void run() {

this.v.VoteCount(0, 60);

}

}

class ThreadB extends Thread {

Vote v;

ThreadB(Vote v) {

this.v = v;

}

public void run() {

this.v.VoteCount(60, 120);

}

}

class ThreadC extends Thread {

Vote v;

ThreadC(Vote v) {

this.v = v;

}

public void run() {

this.v.VoteCount(120, 180);

}

}

class ThreadD extends Thread {

Vote v;

ThreadD(Vote v) {

this.v = v;

}

public void run() {

this.v.VoteCount(180, 240);

}

}

public class election\_mt {

public static void main(String args[]) {

Vote v = new Vote();

v.GenerateVotes();

ThreadA a = new ThreadA(v);

ThreadB b = new ThreadB(v);

ThreadC c = new ThreadC(v);

ThreadD d = new ThreadD(v);

a.start();

b.start();

c.start();

d.start();

try {

a.join();

b.join();

c.join();

d.join();

} catch (Exception e) {

System.out.println("Exception has " + e);

}

if (v.A\_vote >= v.B\_vote && v.A\_vote >= v.C\_vote) {

System.out.println("A is the winner with " + v.A\_vote + " votes");

} else if (v.B\_vote >= v.A\_vote && v.B\_vote >= v.C\_vote) {

System.out.println("B is the winner with " + v.B\_vote + " votes");

} else {

System.out.println("C is the winner with " + v.C\_vote + " votes");

}

System.out.println("A:" + v.A\_vote + " B: " + v.B\_vote + " C: " + v.C\_vote);

}

}

