**Program 1**

**Arithmetic Exception**

public class MultipleCatchBlock1 {

public static void main(String[] args) {

try{

int a[]=new int[5];

a[5]=30/0; }

catch(ArithmeticException e) {

System.out.println("Arithmetic Exception occurs");

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("ArrayIndexOutOfBounds Exception occurs");

}

catch(Exception e) {

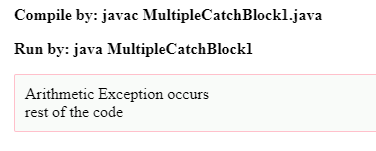
System.out.println("Parent Exception occurs");

}

System.out.println("rest of the code");

}

}



**Array Index out of Bounds**

public class MultipleCatchBlock2 {

public static void main(String[] args) {

try{

int a[]=new int[5];

System.out.println(a[10]);

}

catch(ArithmeticException e)

{

System.out.println("Arithmetic Exception occurs");

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("ArrayIndexOutOfBounds Exception occurs");

}

catch(Exception e)

{

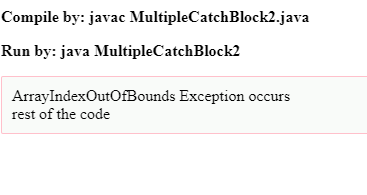
System.out.println("Parent Exception occurs");

}

System.out.println("rest of the code");

}

}



**Null pointer Exception**

import java.io.\*;

class ne

{

public static void main (String[] args)

{

// Initializing String variable with null value

String ptr = null;

// Checking if ptr.equals null or works fine.

try

{

if (ptr.equals("gfg"))

System.out.print("Same");

else

System.out.print("Not Same");

}

catch(NullPointerException e)

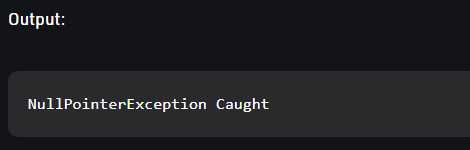
{

System.out.print("NullPointerException Caught");

}

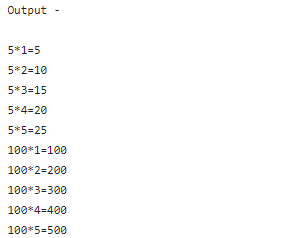
}

}



**Program 2**

|  |
| --- |
| class Table |
|  | { |
|  | void printTable(int n) |
|  | { |
|  | synchronized(this) |
|  | { |
|  | for(int i=1;i<=5;i++) |
|  | { |
|  | System.out.println(+n+"\*"+i+"="+(n\*i)); |
|  | try |
|  | { |
|  | Thread.sleep(400); |
|  | } |
|  | catch(Exception e) |
|  | { |
|  | System.out.println(e); |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  |  |
|  | class Mythread1 extends Thread |
|  | { |
|  | Table t; |
|  | Mythread1(Table t) |
|  | { |
|  | this.t=t; |
|  | } |
|  | public void run() |
|  | { |
|  | t.printTable(5); |
|  | } |
|  | } |
|  |  |
|  |  |
|  | class Mythread2 extends Thread |
|  | { |
|  | Table t; |
|  | Mythread2(Table t) |
|  | { |
|  | this.t=t; |
|  | } |
|  | public void run() |
|  | { |
|  | t.printTable(100); |
|  | } |
|  | } |
|  |  |
|  | class Use |
|  | { |
|  | public static void main(String args[]) |
|  | { |
|  | Table obj = new Table(); |
|  | Mythread1 th1 = new Mythread1(obj); |
|  | Mythread2 th2 = new Mythread2(obj); |
|  | th1.start(); |
|  | th2.start(); |
|  | } |
|  | } |

****

**Program 3**

import java.util.\*;

import java.io.\*;

public class ugly {

public static void main(String args[]) {

int inputNumber;

Scanner sc=new Scanner(System.in);

System.out.println("Enter the number :");

inputNumber=sc.nextInt();

boolean check = true;

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

if(i!=2&&i!=3&&i!=5) {

if(inputNumber%i==0&&checkPrime(i)) {

check = false;

break;

}

}

}

if(check) {

System.out.println(inputNumber+" is an ugly number");

} else {

System.out.println(inputNumber+" is Not an ugly number");

}

}

static boolean checkPrime(int number) {

boolean flag = true;

for(int i = 2; i<=number/2; i++) {

if(number%i==0) {

flag = false;

break;

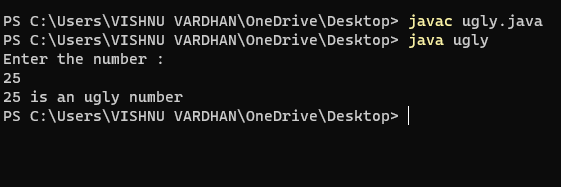
}

}

return flag;

}

}

****

**Program 4**

import java.io.\*;

import java.util.\*;

class fibo {

static int fib(int n)

{

if (n==0||n==1)

return 0;

else if(n==2)

return 1;

return fib(n - 1) + fib(n - 2);

}

public static void main(String args[])

{

int n;

Scanner sc=new Scanner(System.in);

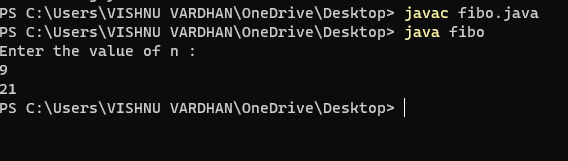
System.out.println("Enter the value of n : ");

n=sc.nextInt();

System.out.println(fib(n));

}

}



**Program 5**

**import java.io.\*;**

**import java.util.\*;**

**class duplicate {**

**static int removeDuplicates(int arr[], int n) {**

**if (n == 0 || n == 1)**

**return n;**

**int[] temp = new int[n];**

**int j = 0;**

**for (int i = 0; i < n-1; i++) {**

**if (arr[i] != arr[i+1])**

**temp[j++] = arr[i];**

**}**

**temp[j++] = arr[n-1];**

**for (int i = 0; i < j; i++) {**

**arr[i] = temp[i];**

**}**

**return j;**

**}**

**public static void main(String[] args) {**

**int arr[] = {10, 20, 20, 30, 40, 40, 40, 50, 50};**

**int n = arr.length;**

**n = removeDuplicates(arr, n);**

**for (int i = 0; i < n; i++) {**

**System.out.print(arr[i]+" ");**

**}**

**}**

**}**

