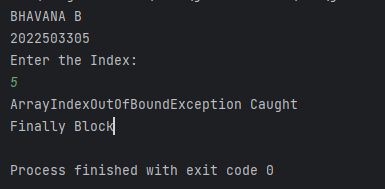
**1. WRITE A PROGRAM TO PERFORM UNCHECKED EXCEPTION. USE APPROPRIATE TRY -CATCH BLOCKS TO HANDLE THESE EXCEPTIONS AND PROVIDE MEANINGFUL ERROR MESSAGES**

**A**

**PROGRAM**

import java.util.\*;  
public class Assignment7\_1\_3305 {  
 public static void main(String[] args){  
 Scanner scan = new Scanner(System.*in*);  
 try {  
 System.*out*.println("BHAVANA B\n2022503305");  
 int[] arr = {1, 2, 3, 4};  
 int index;  
 System.*out*.println("Enter the Index: ");  
 index = scan.nextInt();  
 System.*out*.println("The value is: " + arr[index]);  
 }  
 catch (ArrayIndexOutOfBoundsException ae){  
 System.*out*.println("ArrayIndexOutOfBoundException Caught");  
 }  
 finally{  
 System.*out*.println("Finally Block");  
 scan.close();  
 }  
 }  
}

**OUTPUT**

’

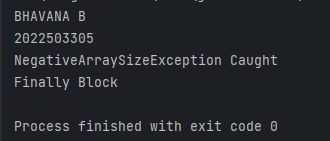
**B.**

**PROGRAM**

import java.util.\*;  
public class Assignment7\_1\_3305 {  
 public static void main(String[] args){  
 Scanner scan = new Scanner(System.*in*);  
 try {

int[] negativeArray = new int[-1];  
 }  
 catch (NegativeArraySizeException nase) {  
 System.*out*.println("NegativeArraySizeException Caught");  
 }  
 finally{  
 System.*out*.println("Finally Block");  
 scan.close();  
 }  
 }  
}

**OUTPUT**



**C.**

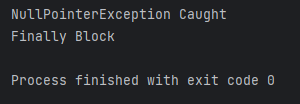
**PROGRAM**

import java.util.\*;  
public class Assignment7\_1\_3305 {  
 public static void main(String[] args){  
 Scanner scan = new Scanner(System.*in*);  
 try {

String str = null

}  
 catch (NullPointerException npe) {  
 System.*out*.println("NullPointerException Caught");  
 }  
 finally{  
 System.*out*.println("Finally Block");  
 scan.close();  
 }  
 }  
}

**OUTPUT**

****

**D.**

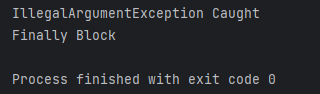
**PROGRAM**

import java.util.\*;  
public class Assignment7\_1\_3305 {  
 public static void main(String[] args){  
 Scanner scan = new Scanner(System.*in*);  
 try {

int num = Integer.*parseInt*("abc");

}  
 catch (NullPointerException npe) {  
 System.*out*.println("NullPointerException Caught");  
 }  
 finally{  
 System.*out*.println("Finally Block");  
 scan.close();  
 }  
 }  
}

**OUTPUT**

****

**E**

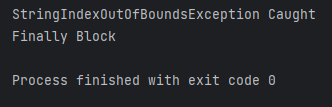
**PROGRAM**

import java.util.\*;  
public class Assignment7\_1\_3305 {  
 public static void main(String[] args){  
 Scanner scan = new Scanner(System.*in*);  
 try {

String s = "test";  
System.*out*.println(s.charAt(10));

}  
 catch (NullPointerException npe) {  
 System.*out*.println("NullPointerException Caught");  
 }  
 finally{  
 System.*out*.println("Finally Block");  
 scan.close();  
 }  
 }  
}

**OUTPUT**

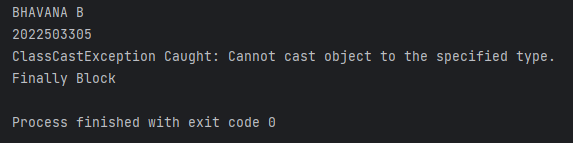
****

**F**

**PROGRAM**

import java.util.\*;  
public class Assignment7\_1\_3305 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 try {  
 System.*out*.println("BHAVANA B\n2022503305");  
 Object obj = "Hello, World!";  
 Integer num = (Integer) obj;  
 System.*out*.println("Value: " + num);  
 } catch (ClassCastException cce) {  
 System.*out*.println("ClassCastException Caught: Cannot cast object to the specified type.");  
 } finally {  
 System.*out*.println("Finally Block");  
 scan.close();  
 }  
 }  
}

**OUTPUT**

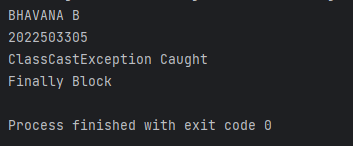
****

**G**

**PROGRAM**

import java.util.\*;  
public class Assignment7\_1\_3305 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("BHAVANA B\n2022503305");  
 List<Object> objList = new ArrayList<>();  
 objList.add("String Element");  
 objList.add(42);  
 String[] strArray = new String[objList.size()];  
 for (int i = 0; i < objList.size(); i++) {  
 strArray[i] = (String) objList.get(i);   
 }  
 System.*out*.println("Values: " + Arrays.*toString*(strArray));  
 } catch (ArrayStoreException ase) {  
 System.*out*.println("ArrayStoreException Caught");  
 } catch (ClassCastException cce) {  
 System.*out*.println("ClassCastException Caught");  
 } finally {  
 System.*out*.println("Finally Block");  
 }  
 }  
}

**OUTPUT**

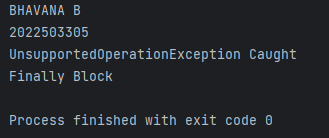


**H**

**PROGRAM**

import java.util.\*;  
public class Assignment7\_1\_3305 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("BHAVANA B\n2022503305");  
 List<String> originalList = new ArrayList<>();  
 originalList.add("Element 1");  
 originalList.add("Element 2");  
  
 List<String> unmodifiableList = Collections.*unmodifiableList*(originalList);  
 unmodifiableList.add("Element 3");  
  
 System.*out*.println("List: " + unmodifiableList);  
 } catch (UnsupportedOperationException uoe) {  
 System.*out*.println("UnsupportedOperationException Caught");  
 } finally {  
 System.*out*.println("Finally Block");  
 }  
 }  
}

**OUTPUT**

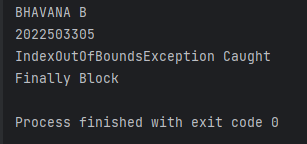
****

**I**

**PROGRAM**

import java.util.\*;  
public class Assignment7\_1\_3305 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("BHAVANA B\n2022503305");  
 List<String> list = new ArrayList<>();  
 list.add("Element 1");  
 list.add("Element 2");  
 list.add("Element 3");  
 System.*out*.println("Accessing Element at Index 5: " + list.get(5));  
  
 } catch (IndexOutOfBoundsException ioobe) {  
 System.*out*.println("IndexOutOfBoundsException Caught");  
 } finally {  
 System.*out*.println("Finally Block");  
 }  
 }  
}

**OUTPUT**

****

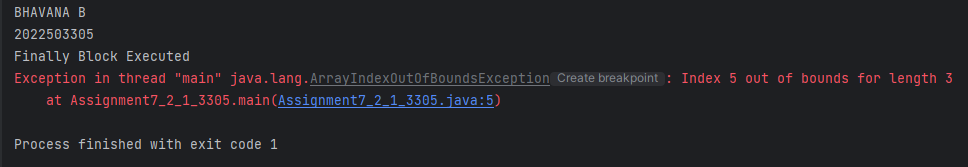
**2. 2. WRITE A PROGRAM THAT DEMONSTRATES DIFFERENT TRY-CATCH-FINALLY BLOCK COMBINATIONS**

**A. TRY WITHOUT CATCH BLOCK**

**PROGRAM**

public class Assignment7\_2\_1\_3305 {  
 public static void main(String[] args) {  
 try {  
 int[] arr = {1, 2, 3};  
 System.*out*.println(arr[5]);  
 } finally {  
 System.*out*.println("BHAVANA B\n2022503305");  
 System.*out*.println("Finally Block Executed");  
 }  
 }  
}

**OUTPUT**

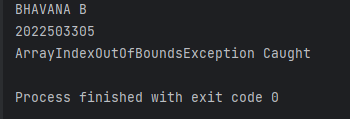
****

**B. TRY WITHOUT FINALLY BLOCK**

**PROGRAM**

import java.sql.\*;  
public class Assignment7\_2\_2\_3305 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("BHAVANA B\n2022503305");  
 int[] arr = {1, 2, 3};  
 System.*out*.println(arr[5]);  
 } catch (ArrayIndexOutOfBoundsException e) {  
  
 System.*out*.println("ArrayIndexOutOfBoundsException Caught");  
 }  
 }  
}

**OUTPUT**

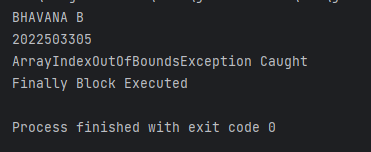
****

**C. TRY WITH CATCH AND FINALLY BLOCK**

**PROGRAM**

public class Assignment7\_2\_3\_3305 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("BHAVANA B\n2022503305");  
 int[] arr = {1, 2, 3};  
 System.*out*.println(arr[5]); // This will cause ArrayIndexOutOfBoundsException  
 } catch (ArrayIndexOutOfBoundsException e) {  
 System.*out*.println("ArrayIndexOutOfBoundsException Caught: Invalid index access.");  
 } finally {  
 System.*out*.println("Finally Block Executed");  
 }  
 }  
}

**OUTPUT**

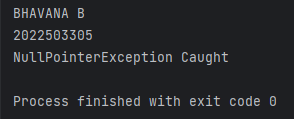
****

**D. TRY WITH MULTIPLE CATCH BLOCK**

**PROGRAM**

public class Assignment7\_2\_4\_3305 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("BHAVANA B\n2022503305");  
 String[] arr = new String[2];  
 arr[0] = "Hello";  
 System.*out*.println(arr[1].length());  
 int num = Integer.*parseInt*("abc");  
 } catch (NullPointerException e) {  
 System.*out*.println("NullPointerException Caught");  
 } catch (NumberFormatException e) {  
 System.*out*.println("NumberFormatException Caught");  
 } catch (ArrayIndexOutOfBoundsException e) {  
 System.*out*.println("ArrayIndexOutOfBoundsException Caught");  
 }  
 }  
}

**OUTPUT**

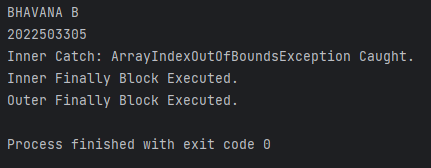
****

**E. NESTED TRY CATCH FINALLY BLOCK**

**PROGRAM**

public class Assignment7\_2\_5\_3305 {  
 public static void main(String[] args) {  
 try {  
 System.*out*.println("BHAVANA B\n2022503305");  
 int[] outerArray = {1, 2, 3};  
 try {  
 System.*out*.println(outerArray[5]); // This will cause ArrayIndexOutOfBoundsException  
 } catch (ArrayIndexOutOfBoundsException e) {  
 System.*out*.println("Inner Catch: ArrayIndexOutOfBoundsException Caught.");  
 } finally {  
 System.*out*.println("Inner Finally Block Executed.");  
 }  
 } catch (Exception e) {  
 System.*out*.println("Outer Catch: General Exception Caught.");  
 } finally {  
 System.*out*.println("Outer Finally Block Executed.");  
 }  
 }  
}

**OUTPUT**

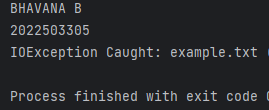
****

**F**.

**PROGRAM**

import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.IOException;  
public class Assignment7\_2\_6\_3305 {  
 public static void main(String[] args) {  
 String filePath = "example.txt";  
 System.*out*.println("BHAVANA B\n2022503305");  
 try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {  
 String line;  
 while ((line = br.readLine()) != null) {  
 System.*out*.println(line);  
 }  
 } catch (IOException e) {  
 System.*out*.println("IOException Caught: " + e.getMessage());  
 }  
 }  
}

**OUTPUT**



**3 CREATE A CUSTOM EXCEPTION CLASS CALLED INVALIDMARKEXCEPTION THAT EXTENDS EXCEPTION. THEN, WRITE A STUDENT CLASS WITH A METHOD TO SET MARKS THAT THROWS THIS CUSTOM EXCEPTION IF THE MARK IS OUT OF RANGE (E.G., LESS THAN 0 OR GREATER THAN 100).**

**PROGRAM**

import java.util.Scanner;

class InvalidMarkException extends Exception {

public InvalidMarkException(String message) {

super(message);

}

}

class Student {

private int marks;

public void setMarks(int marks) throws InvalidMarkException {

if (marks < 0 || marks > 100) {

throw new InvalidMarkException("Marks must be between 0 and 100.");

}

this.marks = marks;

}

public int getMarks() {

return marks;

}

}

public class Assignment7\_3\_3305 {

public static void main(String[] args) {

Student student = new Student();

try {

student.setMarks(105);

} catch (InvalidMarkException e) {

System.out.println("Error: " + e.getMessage());

}

try {

System.out.println("BHAVANA B\n2022503305");

Scanner scan = new Scanner(System.in);

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

int marks;

marks = scan.nextInt();

student.setMarks(marks);

System.out.println("Marks set successfully: " + student.getMarks());

} catch (InvalidMarkException e) {

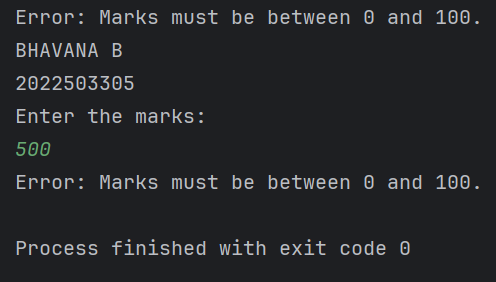
System.out.println("Error: " + e.getMessage());

}

}

}

**OUTPUT**

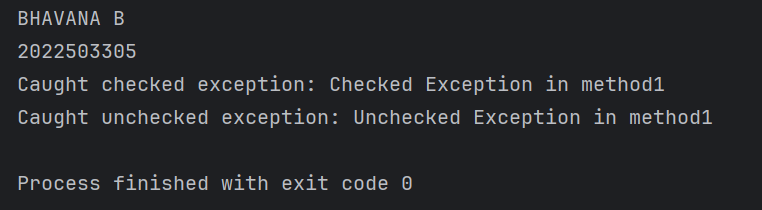
****

**4. WRITE A PROGRAM TO ILLUSTRATE THE PROPAGATION OF CHECKED AND UNCHECKED EXCEPTION.**

**PROGRAM**

class CheckedException {  
 public void method1() throws java.io.IOException {  
 throw new java.io.IOException("Checked Exception in method1");  
 }  
  
 public void method2() throws java.io.IOException {  
 method1();  
 }  
  
 public void method3() {  
 try {  
 method2();  
 } catch (java.io.IOException e) {  
 System.*out*.println("Caught checked exception: " + e.getMessage());  
 }  
 }  
}  
  
class UnCheckedException {  
 public void method1() {  
 throw new ArithmeticException("Unchecked Exception in method1");  
 }  
  
 public void method2() {  
 method1();  
 }  
  
 public void method3() {  
 try {  
 method2();  
 } catch (ArithmeticException e) {  
 System.*out*.println("Caught unchecked exception: " + e.getMessage());  
 }  
 }  
}  
public class Assignment7\_4\_3305 {  
 public static void main(String[] args) {  
 System.*out*.println("BHAVANA B\n2022503305");  
 CheckedException checked = new CheckedException();  
 checked.method3();  
 UnCheckedException unchecked = new UnCheckedException();  
 unchecked.method3();  
 }  
}

**OUTPUT**

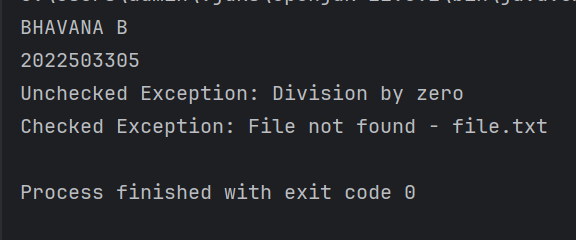
****

**5.**

**PROGRAM**

class ExceptionHandlingOverloading {  
 public void handleException(String fileName) throws java.io.IOException {  
 throw new java.io.IOException("Checked Exception: File not found - " + fileName);  
 }  
 public void handleException(int number) {  
 if (number == 0) {  
 throw new ArithmeticException("Unchecked Exception: Division by zero");  
 } else {  
 System.*out*.println("Result: " + (10 / number));  
 }  
 }  
}  
public class Assignment7\_5\_3305 {  
 public static void main(String[] args) {  
 System.*out*.println("BHAVANA B\n2022503305");  
 ExceptionHandlingOverloading obj = new ExceptionHandlingOverloading();  
 try {  
 obj.handleException(0);  
 } catch (ArithmeticException e) {  
 System.*out*.println(e.getMessage());  
 }  
 try {  
 obj.handleException("file.txt");  
 } catch (java.io.IOException e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

**OUTPUT**

****

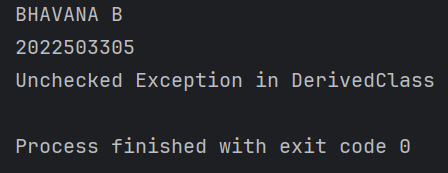
**6.**

**A. OVERRIDING A METHOD THAT THROWS AN UNCHECKED EXCEPTION**

**PROGRAM**

class BaseClass {  
 public void display() {  
 throw new ArithmeticException("Unchecked Exception in BaseClass");  
 }  
}  
  
class DerivedClass extends BaseClass {  
 @Override  
 public void display() {  
 throw new ArithmeticException("Unchecked Exception in DerivedClass");  
 }  
}  
  
public class Assignment7\_6\_1\_3305 {  
 public static void main(String[] args) {  
 System.*out*.println("BHAVANA B\n2022503305");  
 BaseClass obj = new DerivedClass();  
 try {  
 obj.display();  
 } catch (ArithmeticException e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

**OUTPUT**

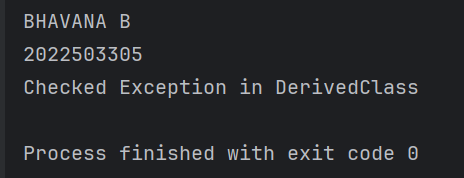
****

**B. OVERRIDING A METHOD THAT THROWS A CHECKED EXCEPTION**

**PROGRAM**

class BaseClass1 {  
 public void display() throws java.io.IOException {  
 throw new java.io.IOException("Checked Exception in BaseClass");  
 }  
}  
class DerivedClass1 extends BaseClass1 {  
 @Override  
 public void display() throws java.io.IOException {  
 throw new java.io.IOException("Checked Exception in DerivedClass");  
 }  
}  
public class Assignment7\_6\_2\_3305 {  
 public static void main(String[] args) {  
 System.*out*.println("BHAVANA B\n2022503305");  
 BaseClass1 obj = new DerivedClass1();  
 try {  
 obj.display();  
 } catch (java.io.IOException e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

**OUTPUT**

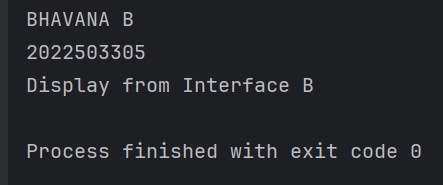
****

**DIAMOND PROBLEM**

**PROGRAM**

interface A {  
 void display();  
}  
interface B extends A {  
 @Override  
 default void display() {  
 System.*out*.println("Display from Interface B");  
 }  
}  
interface C extends A {  
 @Override  
 default void display() {  
 System.*out*.println("Display from Interface C");  
 }  
}  
class D implements B, C {  
 @Override  
 public void display() {  
 B.super.display();  
 }  
}  
public class Assignment7\_7\_3305 {  
 public static void main(String[] args) {  
 System.*out*.println("BHAVANA B\n2022503305");  
 D obj = new D();  
 obj.display();  
 }  
}

**OUTPUT**

****