**Assisted Practice: 3.5 Keywords and Custom Exceptions**

This section will guide you to:

* Write a program in Java to demonstrate try and catch
* Use Eclipse (the popular text editor for Java programs)
* Push code to Git

This lab has six sub-sections, namely:

* + 1. Creating a new project in Eclipse
    2. Writing a program in Java to demonstrate the **throw** keyword
    3. Writing a program in Java to demonstrate the **throws** keyword
    4. Writing a program in Java to demonstrate the **finally** keyword
    5. Writing a program in Java to demonstrate custom exceptions
    6. Pushing the code to your GitHub repositories

**Step 3.5.1:** Creating a new project in Eclipse

* Open Eclipse
* Go to File -> New -> Project -> Java Project -> Next.
* Type in any project name and click on “Finish.”
* Select your project and go to File -> New -> Class.
* Enter **ThrowDemo** in class name, check the checkbox “public static void main(String[] args)”, and click on “Finish.”

**Step 3.5.2:** Writing a program in Java to demonstrate the **throw** keyword

public class TestThrow1 {

//function to check if person is eligible to vote or not

public static void validate(int age) {

if(age<18) {

//throw Arithmetic exception if not eligible to vote

throw new ArithmeticException("Person is not eligible to vote");

}

else {

System.out.println("Person is eligible to vote!!");

}

}

//main method

public static void main(String args[]){

//calling the function

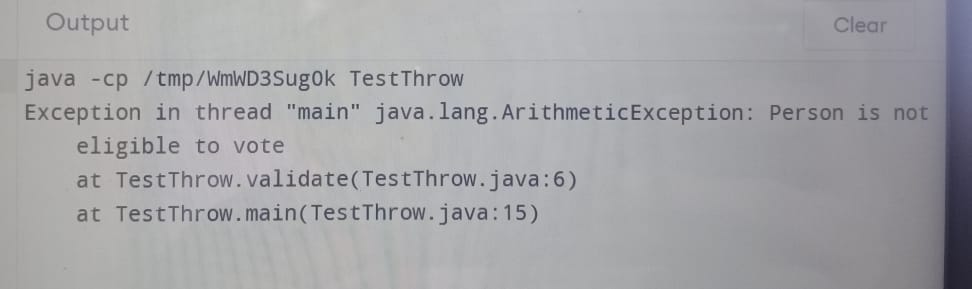
validate(13);

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

}

}

**Output:**

****

**Step 3.5.3:** Writing a program in Java to demonstrate the **throws** keyword

import java.io.IOException;

class Testthrows1{

void m()throws IOException{

throw new IOException("device error");//checked exception

}

void n()throws IOException{

m();

}

void p(){

try{

n();

}catch(Exception e){System.out.println("exception handled");}

}

public static void main(String args[]){

Testthrows1 obj=new Testthrows1();

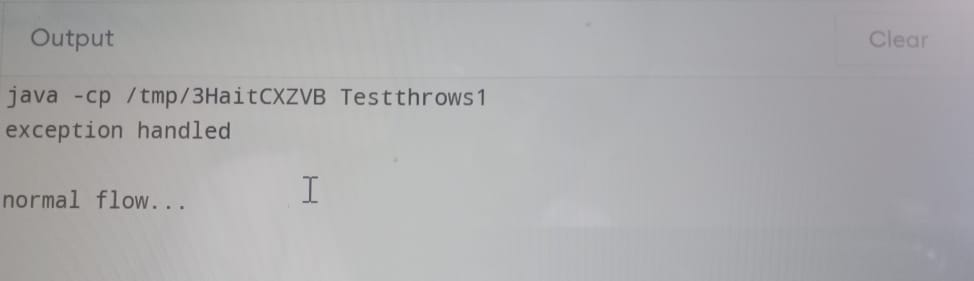
obj.p();

System.out.println("normal flow...");

}

}

**Output:**

****

**Step 3.5.4:** Writing a program in Java to demonstrate the **finally** keyword

class TestFinallyBlock {

 public static void main(String args[]){

  try{

//below code do not throw any exception

   int data=25/5;

   System.out.println(data);

  }

//catch won't be executed

  catch(NullPointerException e){

System.out.println(e);

}

//executed regardless of exception occurred or not

 finally {

System.out.println("finally block is always executed");

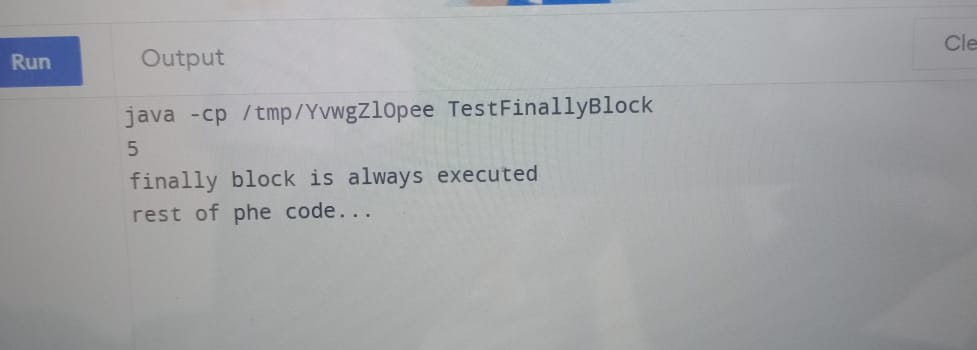
}

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

 }

}

**Output:**

****

**Step 3.5.5:** Writing a program in Java to demonstrate custom exceptions

class MyException extends Exception

{

public MyException(String s)

{

super(s);

}

}

public class Main

{

public static void main(String args[])

{

try

{

throw new MyException("temp");

}

catch (MyException ex)

{

System.out.println("Caught");

System.out.println(ex.getMessage());

}

}

}

**Output:**

