VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Computer Programming Java

MOODLE LAB TASK 6

Faculty: Sathya Raj R

Slot: L3+L4+L29+L30

Venue: SJT 515

NAME: Soumyojyoti Saha

Reg ID: 21BCE4007

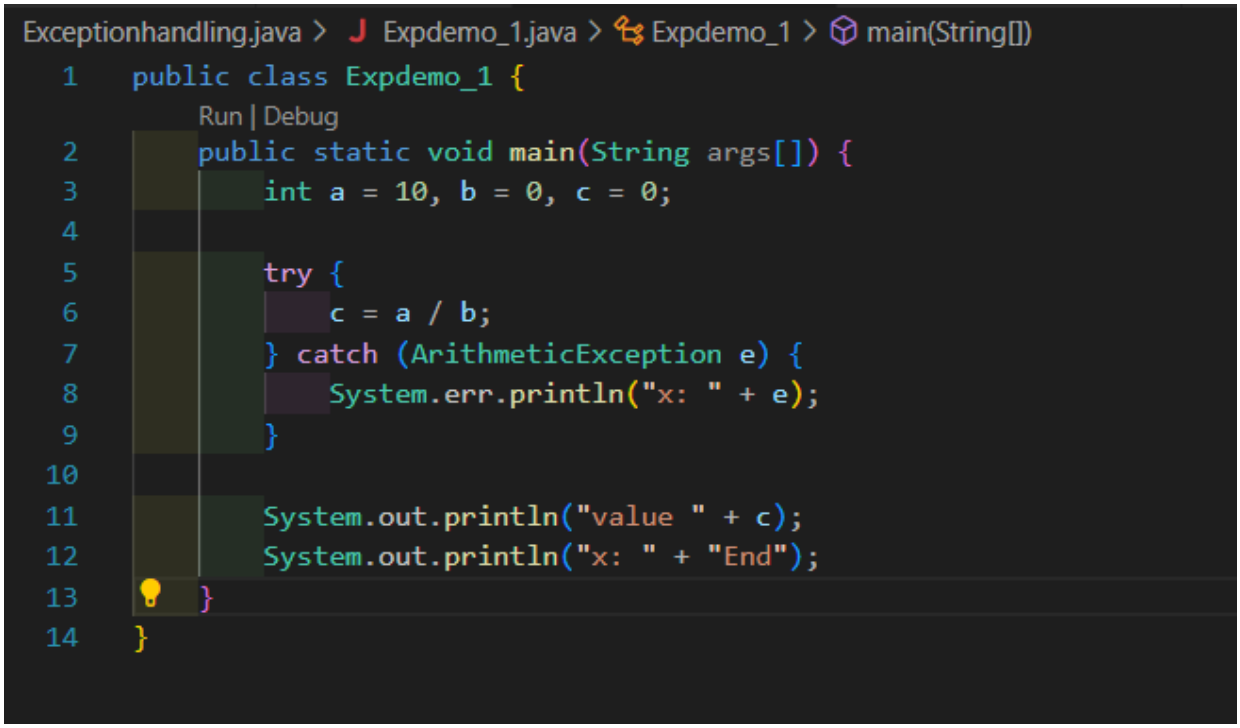
Exception Handling

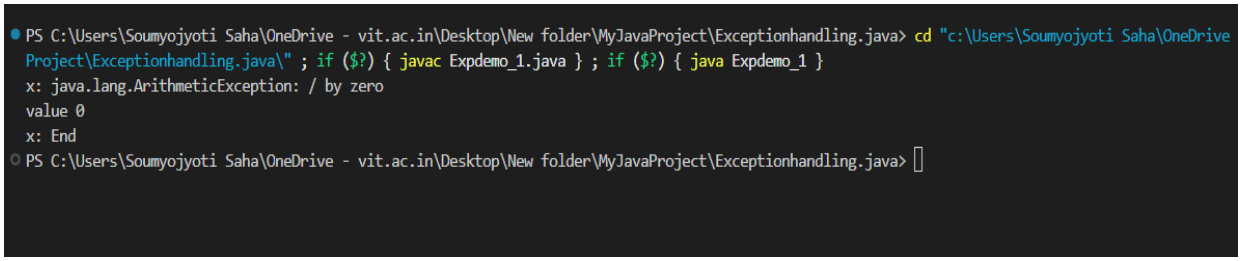
Q1. How can you modify Code 1 to handle a potential ArrayIndexOutOfBoundsException in addition to the existing ArithmeticException?

CODE:

```
public class Expdemo_1 {
    public static void main(String args[]) {
        int a = 10, b = 0, c = 0;

        try {
            c = a / b;
        } catch (ArithmeticException e) {
            System.err.println("x: " + e);
        }

        System.out.println("value " + c);
        System.out.println("x: " + "End");
    }
}
```

OUTPUT:





Q2. How can you modify Code 2 to handle a potential ArrayIndexOutOfBoundsException in addition to the existing ArithmeticException and NullPointerException?

CODE:

```
public class Expdemo_2 {
    public static void main(String[] args) {
        int a = 10, b = 0, c = 0;
```

```
        int[] arr = null;

        try {
            c = a / b;
            System.out.println(arr[1]);
        } catch (ArithmeticException e) {
            System.out.println("x: Arithmetic Exception Error");
        } catch (NullPointerException e) {
            System.out.println("x: Null Pointer Exception Error");
        } catch (Exception e) {
            System.out.println("x: ERROR");
        } finally {
            System.out.println("x: FINAL");
        }

        System.out.println("value " + c);
        System.out.println("x: End");
    }
}
```

OUTPUT:

Exceptionhandling.java > Expdemo_2.java > Expdemo_2 > main(String[])

```java
public class Expdemo_2 {
    Run | Debug
    public static void main(String[] args) {
        int a = 10, b = 0, c = 0;
        int[] arr = null;

        try {
            c = a / b;
            System.out.println(arr[1]);
        } catch (ArithmeticException e) {
            System.out.println(x:"x: Arithmetic Exception Error");
        } catch (NullPointerException e) {
            System.out.println(x:"x: Null Pointer Exception Error");
        } catch (Exception e) {
            System.out.println(x:"x: ERROR");
        } finally {
            System.out.println(x:"x: FINAL");
        }

        System.out.println("value " + c);
        System.out.println(x:"x: End");
    }
}
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\New folder\MyJavaProject> cd "c:\Users\Soumyojyoti Saha\OneDrive - Exceptionhandling.java\" ; if ($?) { javac Expdemo_2.java } ; if ($?) { java Expdemo_2 }
x: Arithmetic Exception Error
x: FINAL
value 0
x: End
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\New folder\MyJavaProject\Exceptionhandling.java>
```

Q3. Create a user-defined exception and catch it.

CODE:

```
class MyException extends Exception {
    public MyException(String s) {
        // Call constructor of parent Exception
        super(s);
    }
}

// A Class that uses the above MyException
public class Main {
    // Driver Program
    public static void main(String[] args) {
        try {
            // Throw an object of user defined exception
            throw new MyException("GeeksGeeks");
        }
        catch (MyException ex) {
            System.out.println("Caught");
            // Print the message from MyException object
            System.out.println(ex.getMessage());
        }
    }
}
```

Example1.java | ExceptionExample1.java | Main.java

Exceptionhandling > Main.java > Main > main(String[])

```java
// User-Defined Exception Handling
// Example 1:
// Question: Create a user-defined exception and catch it.

// A Class that represents user-defined exception
class MyException extends Exception {
    public MyException(String s) {
        // Call constructor of parent Exception
        super(s);
    }
}

// A Class that uses the above MyException
public class Main {
    // Driver Program
    Run | Debug
    public static void main(String[] args) {
        try {
            // Throw an object of user defined exception
            throw new MyException(s:"GeeksGeeks");
        }
        catch (MyException ex) {
            System.out.println(x:"Caught");
            // Print the message from MyException object
            System.out.println(ex.getMessage());
        }
    }
}
```

OUTPUT:

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

```
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java sem 7> cd "c:\Users\Soumyojyoti Saha\OneDrive
g\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
Caught
GeeksGeeks
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java sem 7\Exceptionhandling>
```

Q4. Throw and catch a user-defined exception without a message.

CODE:

```
class MyException extends Exception {}

// A Class that uses the above MyException
public class setText {
    // Driver Program
    public static void main(String[] args) {
        try {
            // Throw an object of user defined exception
            throw new MyException();
        }
        catch (MyException e) {
            System.out.println("Caught");
        }
    }
}
```

Exceptionhandling > setText.java > ...

```java
// Example 2:
// Question: Throw and catch a user-defined exception without a message.

// A Class that represents user-defined exception
class MyException extends Exception {}

// A Class that uses the above MyException
public class setText {
    // Driver Program
    Run | Debug
    public static void main(String[] args) {
        try {
            // Throw an object of user defined exception
            throw new MyException();
        }
        catch (MyException e) {
            System.out.println(x:"Caught");
        }
    }
}

```

OUTPUT:

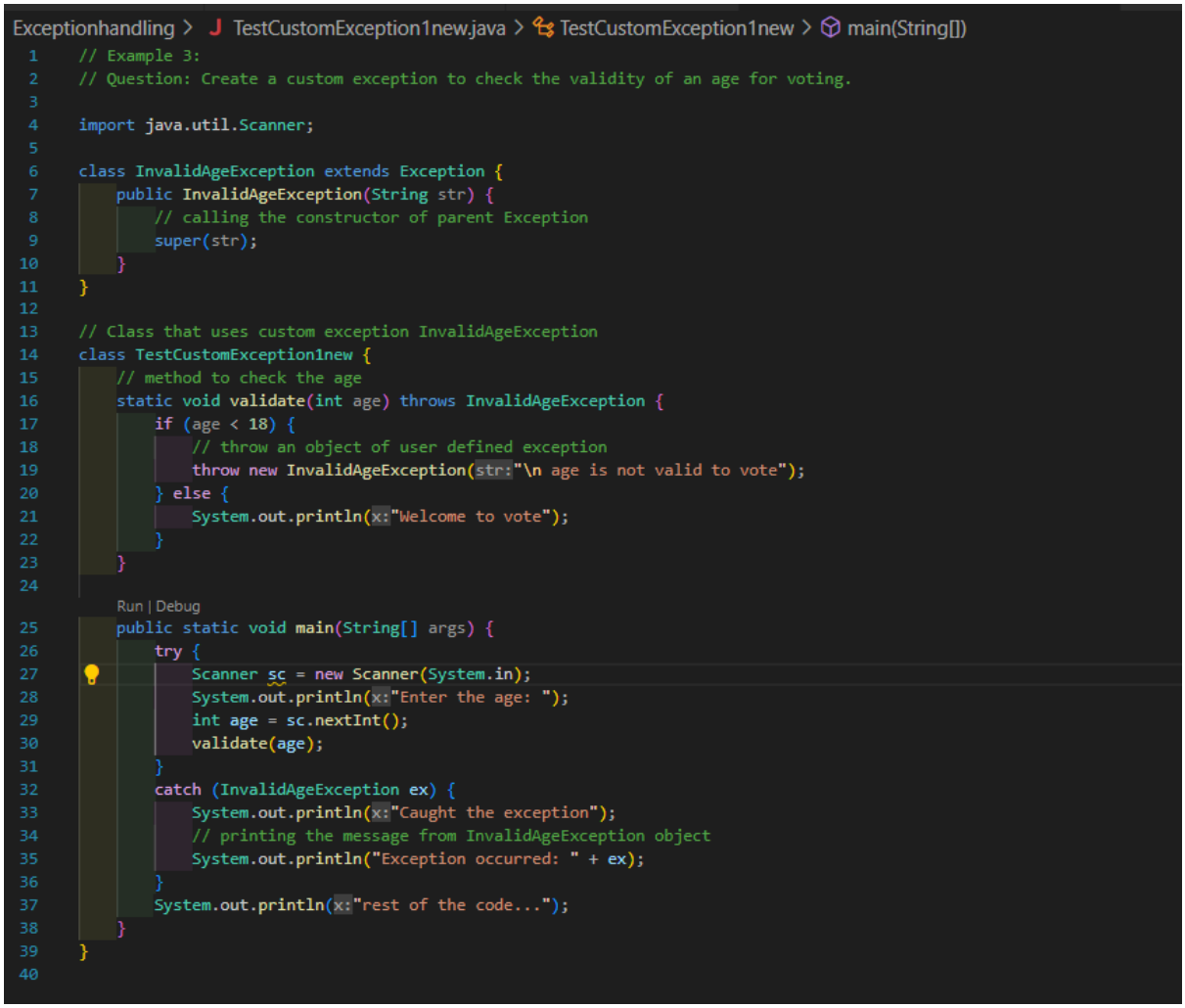
```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java sem 7> cd "c:\Users\Soumyojyoti Saha\OneDrive -
$?) { java setText }
Caught
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java sem 7\Exceptionhandling>
```

Q5. Create a custom exception to check the validity of an age for voting.

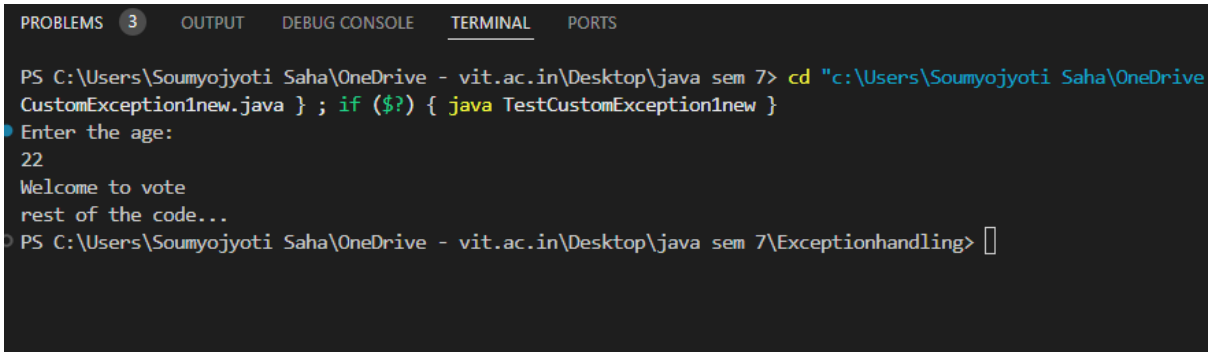
CODE:

```
import java.util.Scanner;

class InvalidAgeException extends Exception {
    public InvalidAgeException(String str) {
        // calling the constructor of parent Exception
        super(str);
    }
}

// Class that uses custom exception InvalidAgeException
class TestCustomException1new {
    // method to check the age
    static void validate(int age) throws InvalidAgeException {
        if (age < 18) {
            // throw an object of user defined exception
            throw new InvalidAgeException("\n age is not valid to vote");
        } else {
            System.out.println("Welcome to vote");
        }
    }

    public static void main(String[] args) {
        try {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter the age: ");
```

```
            int age = sc.nextInt();
            validate(age);
        }
        catch (InvalidAgeException ex) {
            System.out.println("Caught the exception");
            // printing the message from InvalidAgeException object
            System.out.println("Exception occurred: " + ex);
        }
        System.out.println("rest of the code...");
    }
}
```



OUTPUT:



Q6. Handle multiple types of exceptions within one try block.

CODE:

```
class ExceptionExample1 {
    public static void main(String[] args) {
        try {
            int i = 8;
            int j = 0;
            int a[] = new int[5];
            int k = i / j; // may throw arithmetic exception
            a[6] = 10; // may throw array index out of bounds exception
            System.out.println("K value is: " + k);
        }
        catch (ArithmeticException e) {
            System.out.println("Error in arithmetic");
        }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Array limit exceeds..");
        }
        catch (Exception e) {
            System.out.println("Error");
```

```
        }
        finally {
            System.out.println("\nProgram completed");
        }
    }
}
```

Example1.java | ExceptionExample1.java | Expdemo_1.java | TestCustomException1new.java

Exceptionhandling > ExceptionExample1.java > ExceptionExample1 > main(String[])

```java
// Example 4:
// Question: Handle multiple types of exceptions within one try block.

class ExceptionExample1 {
    Run | Debug
    public static void main(String[] args) {
        try {
            int i = 8;
            int j = 0;
            int a[] = new int[5];
            int k = i / j; // may throw arithmetic exception
            a[6] = 10; // may throw array index out of bounds exception
            System.out.println("K value is: " + k);
        }
        catch (ArithmeticException e) {
            System.out.println(x:"Error in arithmetic");
        }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println(x:"Array limit exceeds..");
        }
        catch (Exception e) {
            System.out.println(x:"Error");
        }
        finally {
            System.out.println(x:"\nProgram completed");
        }
    }
}

```

OUTPUT:

PROBLEMS 3 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

```
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java sem 7> cd "c:\Users\Soumyojyoti Saha\OneDrive -
ptionExample1.java } ; if ($?) { java ExceptionExample1 }
Error in arithmetic

Program completed
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java sem 7\Exceptionhandling>
```

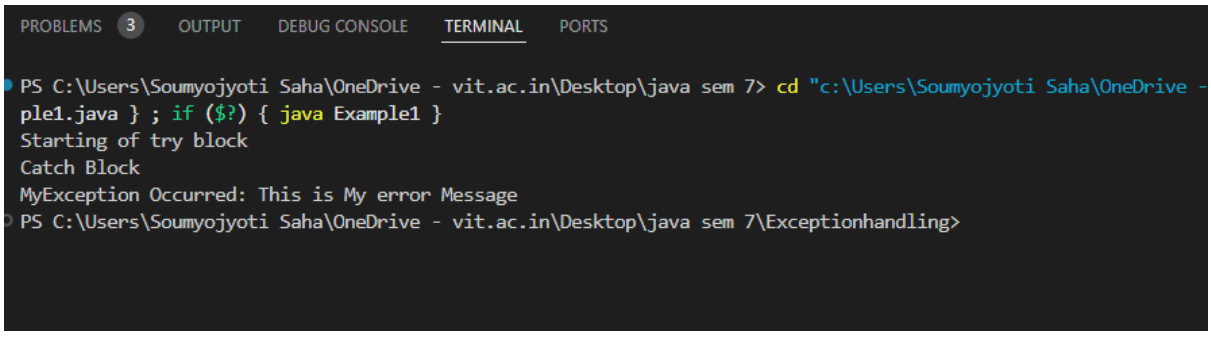
Q7. Create and throw a custom exception with a string message.

CODE:

```
class MyException extends Exception {
    String str1;
    // Constructor of custom exception class
    MyException(String str2) {
        str1 = str2;
    }

    public String toString() {
        return ("MyException Occurred: " + str1);
    }
}

class Example1 {
    public static void main(String[] args) {
        try {
            System.out.println("Starting of try block");
            // Throw the custom exception
            throw new MyException("This is My error Message");
        }
        catch (MyException exp) {
            System.out.println("Catch Block");
            System.out.println(exp);
        }
    }
}
```

```java
class MyException extends Exception {
    String str1;
    // Constructor of custom exception class
    MyException(String str2) {
        str1 = str2;
    }

    public String toString() {
        return ("MyException Occurred: " + str1);
    }
}

class Example1 {
    Run | Debug
    public static void main(String[] args) {
        try {
            System.out.println(x:"Starting of try block");
            // Throw the custom exception
            throw new MyException(str2:"This is My error Message");
        }
        catch (MyException exp) {
            System.out.println(x:"Catch Block");
            System.out.println(exp);
        }
    }
}
```

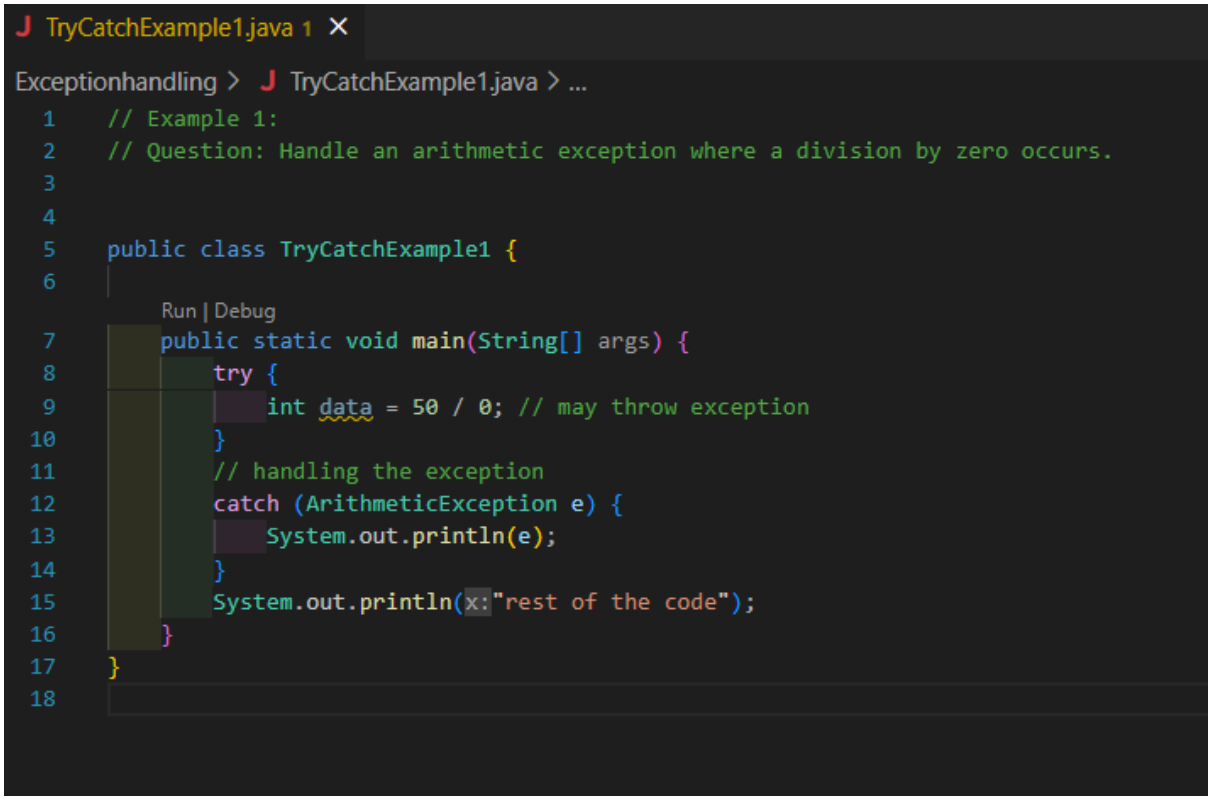
OUTPUT:



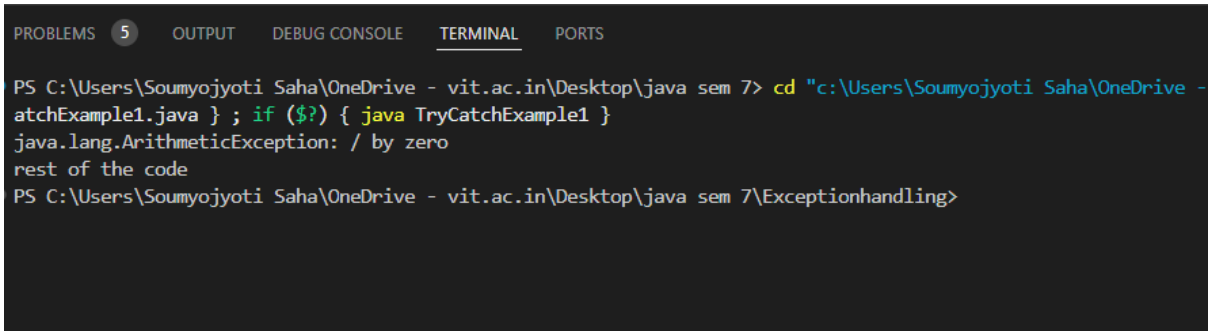
Q8. Handle an arithmetic exception where a division by zero occurs.

CODE:

```
public class TryCatchExample1 {


    public static void main(String[] args) {
        try {
            int data = 50 / 0; // may throw exception
        }
        // handling the exception
        catch (ArithmeticException e) {
            System.out.println(e);
        }
        System.out.println("rest of the code");
    }
}
```



OUTPUT:

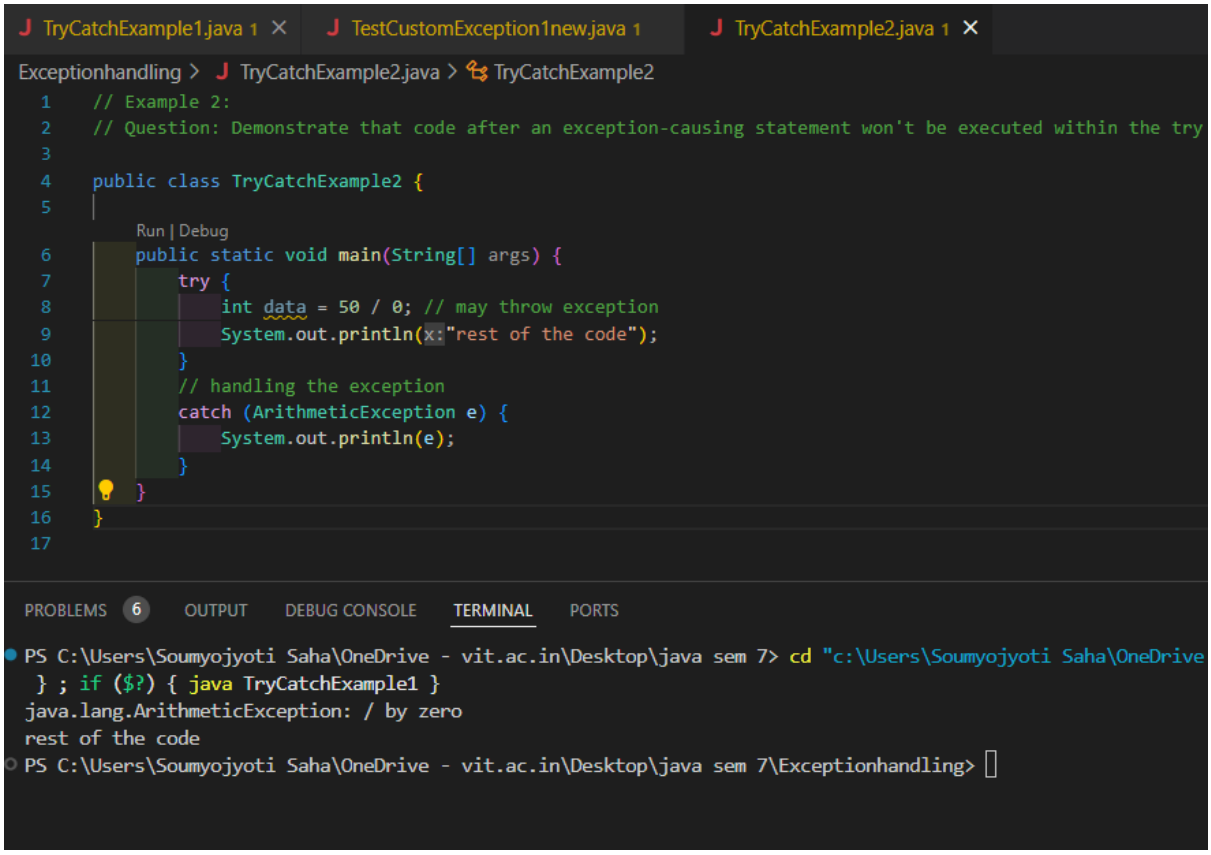


Q9. Demonstrate that code after an exception-causing statement won't be executed within the try block.

CODE:

```
public class TryCatchExample2 {


    public static void main(String[] args) {
        try {
            int data = 50 / 0; // may throw exception
            System.out.println("rest of the code");
        }
        // handling the exception
        catch (ArithmeticException e) {
            System.out.println(e);
        }
    }
}
```

OUTPUT:



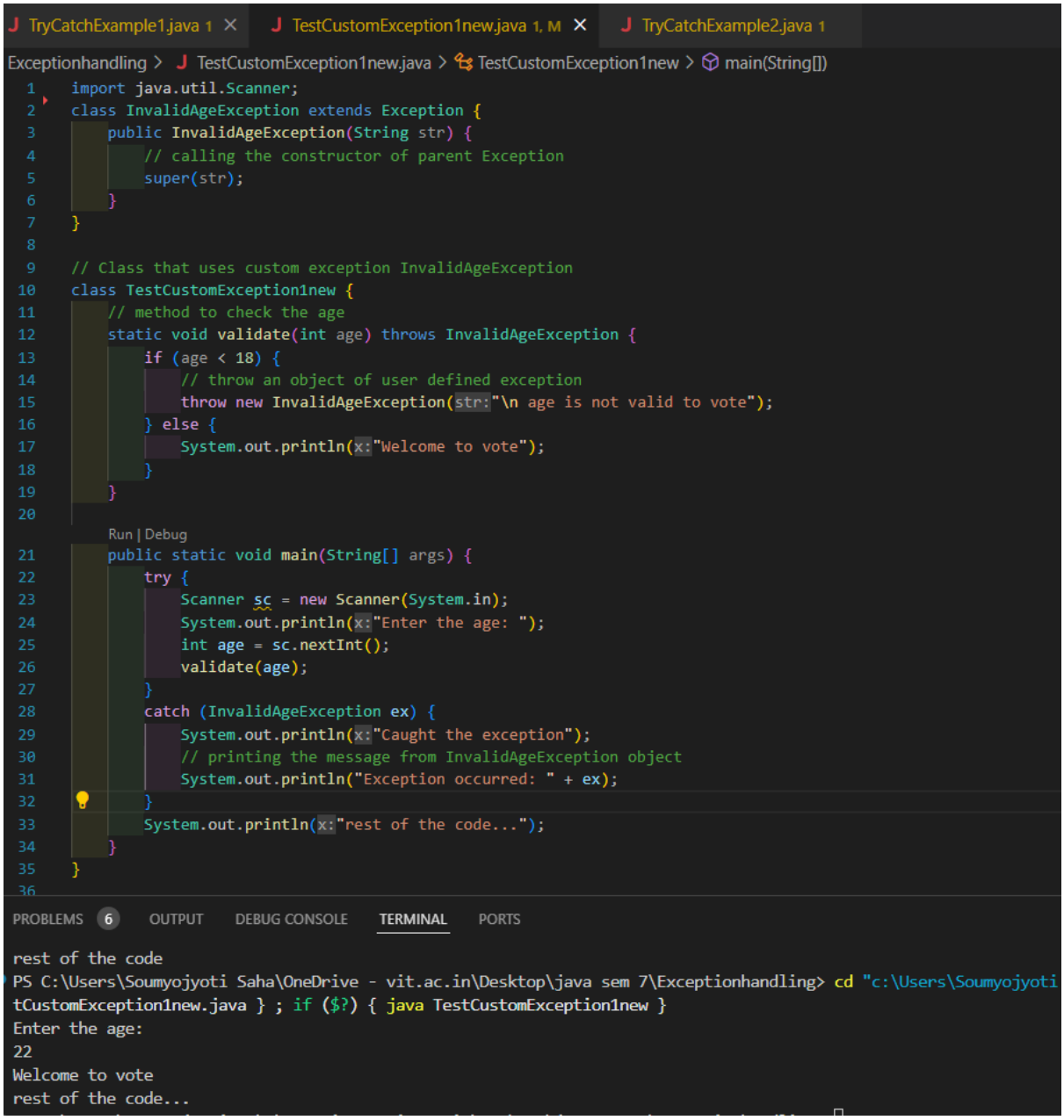
Q10. Create a custom exception to check the validity of an age for voting.

CODE:

```
import java.util.Scanner;


class InvalidAgeException extends Exception {
    public InvalidAgeException(String str) {
        // calling the constructor of parent Exception
        super(str);
    }
}


// Class that uses custom exception InvalidAgeException
class TestCustomException1new {
```

```
    // method to check the age
    static void validate(int age) throws InvalidAgeException {
        if (age < 18) {
            // throw an object of user defined exception
            throw new InvalidAgeException("\n age is not valid to vote");
        } else {
            System.out.println("Welcome to vote");
        }
    }

    public static void main(String[] args) {
        try {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter the age: ");
            int age = sc.nextInt();
            validate(age);
        }
        catch (InvalidAgeException ex) {
            System.out.println("Caught the exception");
            // printing the message from InvalidAgeException object
            System.out.println("Exception occurred: " + ex);
        }
        System.out.println("rest of the code...");
    }
}
```

OUTPUT:

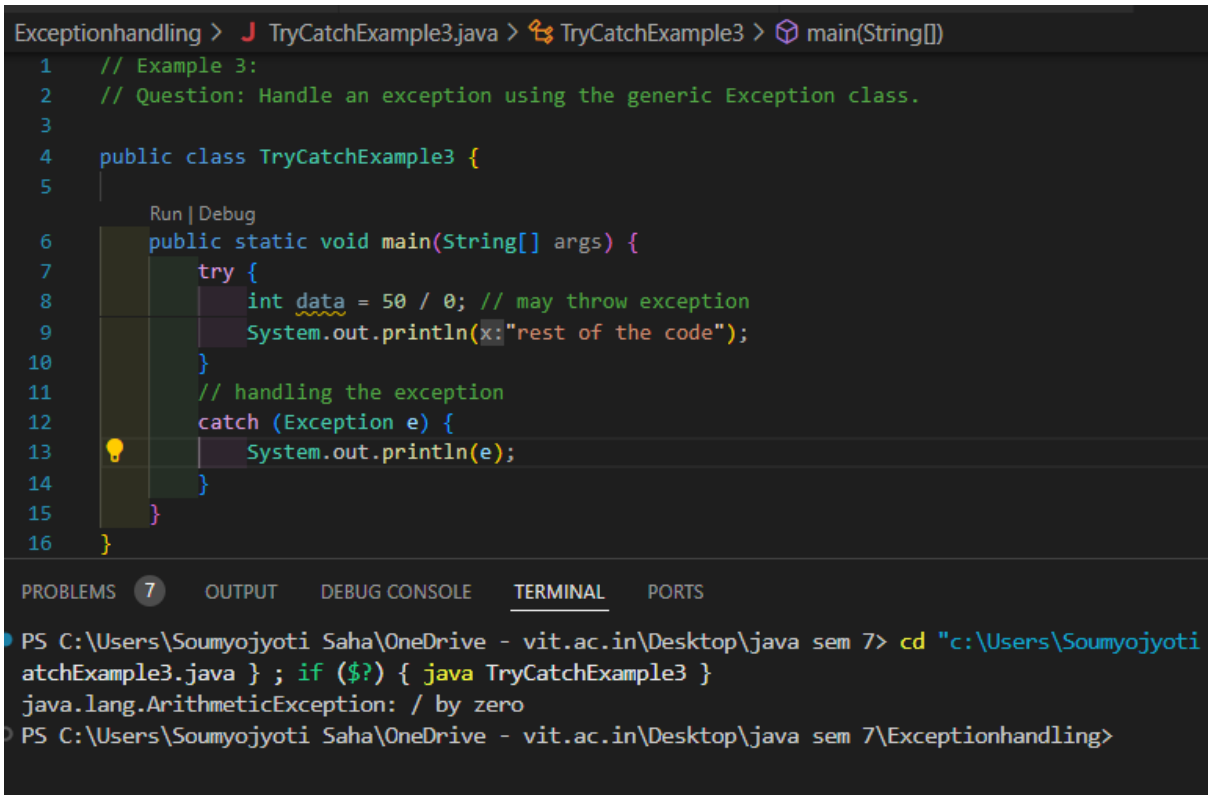

TryCatchExample1.java 1
TestCustomException1new.java 1, M
TryCatchExample2.java 1
Exceptionhandling > TestCustomException1new.java > TestCustomException1new > main(String[])
import java.util.Scanner;
class InvalidAgeException extends Exception {
public InvalidAgeException(String str) {
// calling the constructor of parent Exception
super(str);
}
}
// Class that uses custom exception InvalidAgeException
class TestCustomException1new {
// method to check the age
static void validate(int age) throws InvalidAgeException {
if (age < 18) {
// throw an object of user defined exception
throw new InvalidAgeException(str:"\n age is not valid to vote");
} else {
System.out.println(x:"Welcome to vote");
}
}
Run | Debug
public static void main(String[] args) {
try {
Scanner sc = new Scanner(System.in);
System.out.println(x:"Enter the age: ");
int age = sc.nextInt();
validate(age);
}
catch (InvalidAgeException ex) {
System.out.println(x:"Caught the exception");
// printing the message from InvalidAgeException object
System.out.println("Exception occurred: " + ex);
}
System.out.println(x:"rest of the code...");
}
}
PROBLEMS 6
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
rest of the code
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\java sem 7\Exceptionhandling> cd "c:\Users\Soumyojyoti
tCustomException1new.java } ; if ($?) { java TestCustomException1new }
Enter the age:
22
Welcome to vote
rest of the code...

Q11. Handle an exception using the generic Exception class.

CODE:

```
public class TryCatchExample3 {


    public static void main(String[] args) {
        try {
            int data = 50 / 0; // may throw exception
            System.out.println("rest of the code");
        }
        // handling the exception
        catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

OUTPUT:

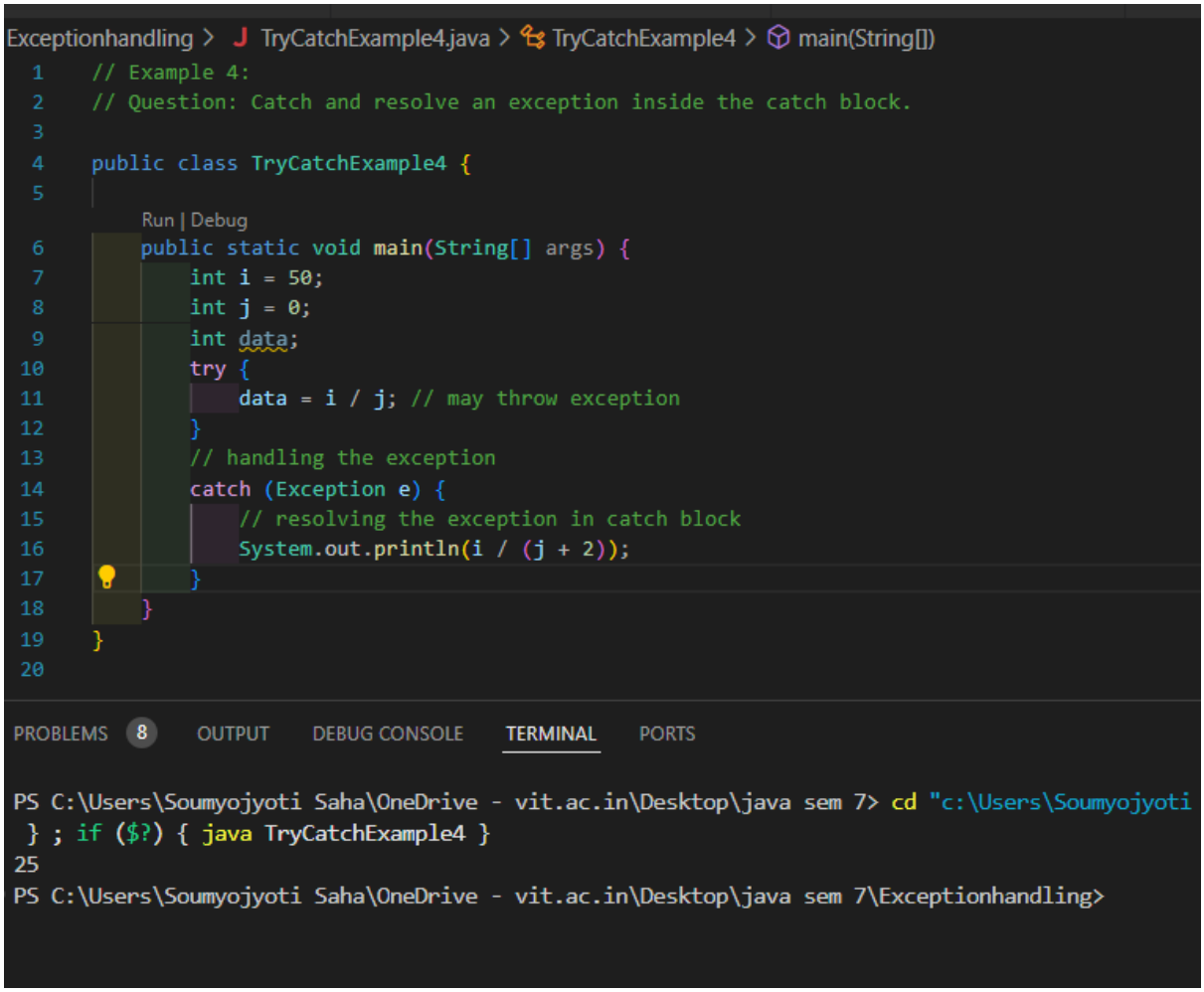


Q12. Catch and resolve an exception inside the catch block.

CODE:

```
public class TryCatchExample4 {


    public static void main(String[] args) {
        int i = 50;
        int j = 0;
        int data;
        try {
            data = i / j; // may throw exception
        }
        // handling the exception
        catch (Exception e) {
            // resolving the exception in catch block
            System.out.println(i / (j + 2));
        }
    }
}
```

OUTPUT:



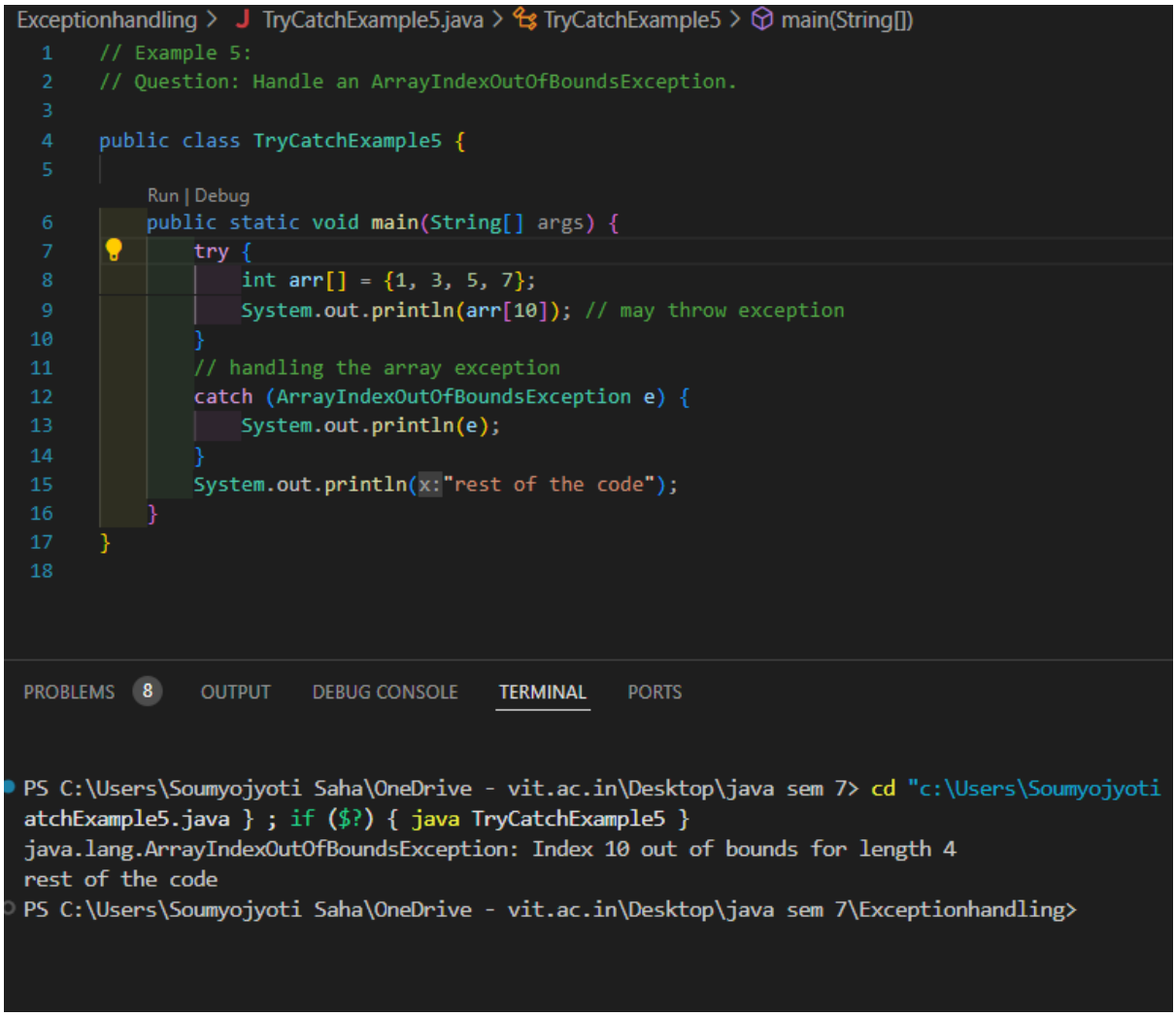
Q13. Handle an ArrayIndexOutOfBoundsException.

CODE:

```
public class TryCatchExample5 {


    public static void main(String[] args) {
        try {
            int arr[] = {1, 3, 5, 7};
            System.out.println(arr[10]); // may throw exception
        }
        // handling the array exception
        catch (ArrayIndexOutOfBoundsException e) {
```

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

OUTPUT:

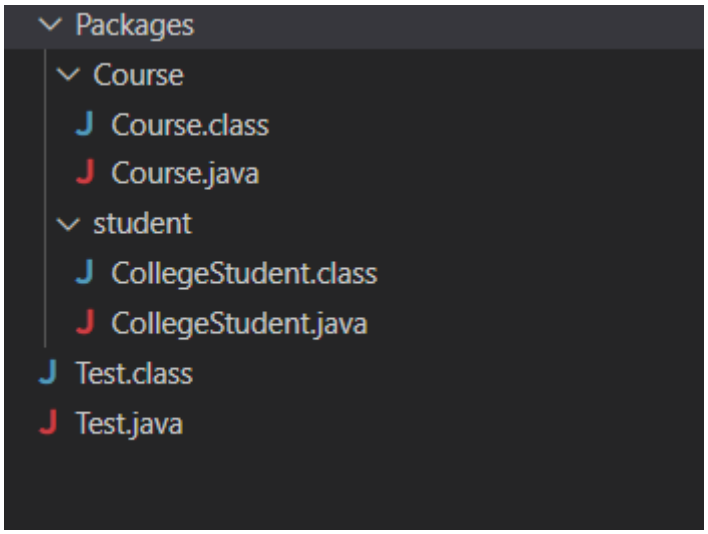


Packages

Write a Java program that demonstrates the following functionality:

1. CollegeStudent Class:
    - Create a class CollegeStudent in the package Packages.student.
    - The class should have the following attributes:
        - name (String): The student's name.
        - regno (String): The student's registration number.
        - age (int): The student's age.
        - courseEnrolled (String): The course the student is enrolled in.
        - contactNumber (String): The student's contact number.
        - email (String): The student's email address.
    - Implement getter and setter methods for each attribute.
    - Include a method displayStudentDetails to print all the student's details.
2. Course Class:
    - Create a class Course in the package Packages.Course.
    - The class should have the following attributes:
        - courseName (String): The name of the course.
        - courseCode (String): The code of the course.
        - courseDescription (String): A brief description of the course.
        - credits (int): The number of credits for the course.
        - instructorName (String): The name of the instructor.
    - Implement getter and setter methods for each attribute.
    - Include a method displayCourseDetails to print all the course's details.
3. Main Class:
    - Create a Test class in the src folder that will:

- Create an instance of the CollegeStudent class, providing all relevant details (name, registration number, age, course enrolled, contact number, and email).
- Create an instance of the Course class, providing details such as course name, course code, description, credits, and instructor name.
- Call the respective methods to display the student's and course's details.



Course/Course.java

```
package Packages.Course;


public class Course {
    private String courseName;
    private String courseCode;
    private String courseDescription;
    private int credits;
    private String instructorName;


    // Constructor to initialize course details
```

```
    public Course(String courseName, String courseCode, String
courseDescription, int credits, String instructorName) {
        this.courseName = courseName;
        this.courseCode = courseCode;
        this.courseDescription = courseDescription;
        this.credits = credits;
        this.instructorName = instructorName;
    }

    // Getter and Setter for Course Name
    public String getCourseName() {
        return courseName;
    }

    public void setCourseName(String courseName) {
        this.courseName = courseName;
    }

    // Getter and Setter for Course Code
    public String getCourseCode() {
        return courseCode;
    }

    public void setCourseCode(String courseCode) {
        this.courseCode = courseCode;
    }

    // Getter and Setter for Course Description
```

```
public String getCourseDescription() {
    return courseDescription;
}

public void setCourseDescription(String courseDescription) {
    this.courseDescription = courseDescription;
}

// Getter and Setter for Credits
public int getCredits() {
    return credits;
}

public void setCredits(int credits) {
    this.credits = credits;
}

// Getter and Setter for Instructor Name
public String getInstructorName() {
    return instructorName;
}

public void setInstructorName(String instructorName) {
    this.instructorName = instructorName;
}

// Display course details
```

```
    public void displayCourseDetails() {
        System.out.println("Course Name: " + courseName);
        System.out.println("Course Code: " + courseCode);
        System.out.println("Course Description: " + courseDescription);
        System.out.println("Credits: " + credits);
        System.out.println("Instructor Name: " + instructorName);
    }
}
```

Course.java

Packages > Course > Course.java > Course > Course(String, String, String, int, String)

```java
package Packages.Course;

public class Course {
    private String courseName;
    private String courseCode;
    private String courseDescription;
    private int credits;
    private String instructorName;

    // Constructor to initialize course details
    public Course(String courseName, String courseCode, String courseDescription, int credits, String
    instructorName) {
        this.courseName = courseName;
        this.courseCode = courseCode;
        this.courseDescription = courseDescription;
        this.credits = credits;
        this.instructorName = instructorName;
    }

    // Getter and Setter for Course Name
    public String getCourseName() {
        return courseName;
    }

    public void setCourseName(String courseName) {
        this.courseName = courseName;
    }

    // Getter and Setter for Course Code
    public String getCourseCode() {
        return courseCode;
    }

    public void setCourseCode(String courseCode) {
        this.courseCode = courseCode;
    }

    // Getter and Setter for Course Description
    public String getCourseDescription() {
        return courseDescription;
    }
```

```java

    public void setCourseDescription(String courseDescription) {
        this.courseDescription = courseDescription;
    }

    // Getter and Setter for Credits
    public int getCredits() {
        return credits;
    }

    public void setCredits(int credits) {
        this.credits = credits;
    }

    // Getter and Setter for Instructor Name
    public String getInstructorName() {
        return instructorName;
    }

    public void setInstructorName(String instructorName) {
        this.instructorName = instructorName;
    }

    // Display course details
    public void displayCourseDetails() {
        System.out.println("Course Name: " + courseName);
        System.out.println("Course Code: " + courseCode);
        System.out.println("Course Description: " + courseDescription);
        System.out.println("Credits: " + credits);
        System.out.println("Instructor Name: " + instructorName);
    }
}

```

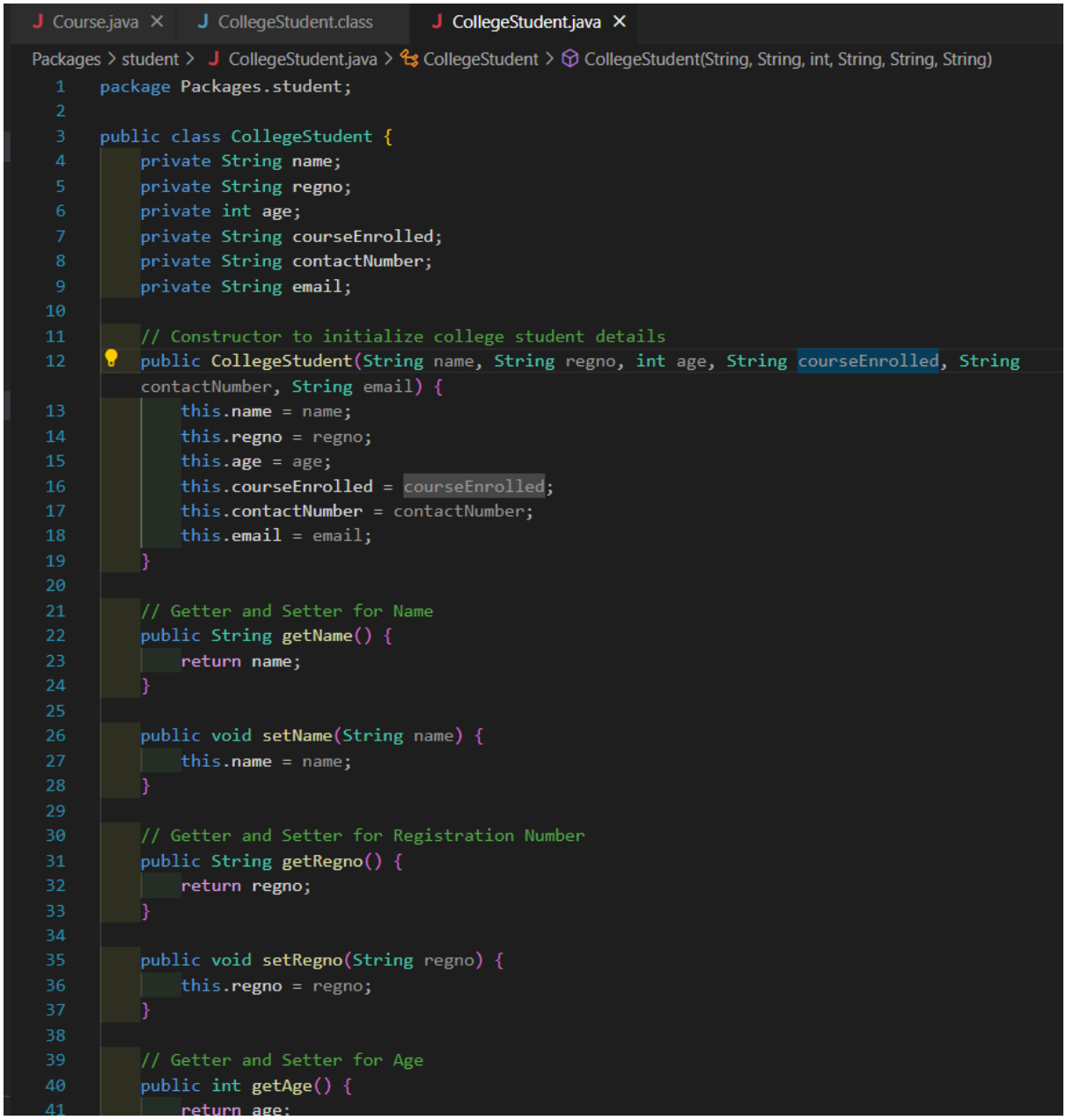
Student/CollegeStudent.java

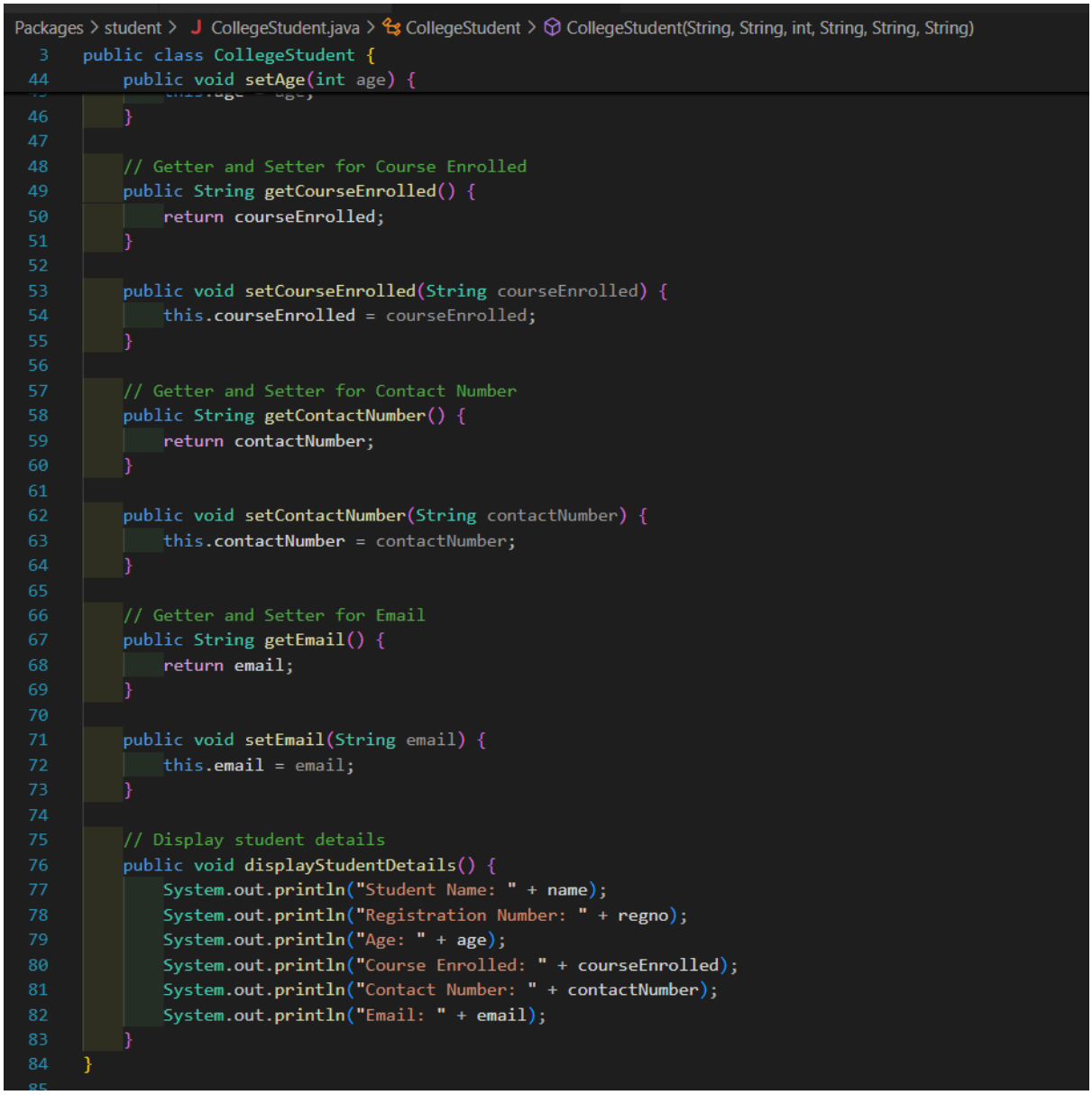
```
package Packages.student;


public class CollegeStudent {
    private String name;
    private String regno;
    private int age;
    private String courseEnrolled;
    private String contactNumber;
    private String email;
```

```
    // Constructor to initialize college student details
    public CollegeStudent(String name, String regno, int age, String
courseEnrolled, String contactNumber, String email) {
        this.name = name;
        this.regno = regno;
        this.age = age;
        this.courseEnrolled = courseEnrolled;
        this.contactNumber = contactNumber;
        this.email = email;
    }

    // Getter and Setter for Name
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    // Getter and Setter for Registration Number
    public String getRegno() {
        return regno;
    }

    public void setRegno(String regno) {
        this.regno = regno;
    }
```

```
// Getter and Setter for Age
public int getAge() {
    return age;
}

public void setAge(int age) {
    this.age = age;
}

// Getter and Setter for Course Enrolled
public String getCourseEnrolled() {
    return courseEnrolled;
}

public void setCourseEnrolled(String courseEnrolled) {
    this.courseEnrolled = courseEnrolled;
}

// Getter and Setter for Contact Number
public String getContactNumber() {
    return contactNumber;
}

public void setContactNumber(String contactNumber) {
    this.contactNumber = contactNumber;
}
```

```
    // Getter and Setter for Email
    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    // Display student details
    public void displayStudentDetails() {
        System.out.println("Student Name: " + name);
        System.out.println("Registration Number: " + regno);
        System.out.println("Age: " + age);
        System.out.println("Course Enrolled: " + courseEnrolled);
        System.out.println("Contact Number: " + contactNumber);
        System.out.println("Email: " + email);
    }
}
```


Course.java
CollegeStudent.class
CollegeStudent.java
Packages > student > CollegeStudent.java > CollegeStudent > CollegeStudent(String, String, int, String, String, String)
1 package Packages.student;
2
3 public class CollegeStudent {
4 private String name;
5 private String regno;
6 private int age;
7 private String courseEnrolled;
8 private String contactNumber;
9 private String email;
10
11 // Constructor to initialize college student details
12 public CollegeStudent(String name, String regno, int age, String courseEnrolled, String contactNumber, String email) {
13 this.name = name;
14 this.regno = regno;
15 this.age = age;
16 this.courseEnrolled = courseEnrolled;
17 this.contactNumber = contactNumber;
18 this.email = email;
19 }
20
21 // Getter and Setter for Name
22 public String getName() {
23 return name;
24 }
25
26 public void setName(String name) {
27 this.name = name;
28 }
29
30 // Getter and Setter for Registration Number
31 public String getRegno() {
32 return regno;
33 }
34
35 public void setRegno(String regno) {
36 this.regno = regno;
37 }
38
39 // Getter and Setter for Age
40 public int getAge() {
41 return age;



Test.java

```
import Packages.student.CollegeStudent;
import Packages.Course.Course;


public class Test {
    public static void main(String[] args) {
        // Create a CollegeStudent with more information
```

CollegeStudent collegeStudent = new CollegeStudent("Soumyojyoti Saha", "21BCE4007", 21, "B.Tech in CSE", "7001813062", "soumyojyoti.saha2021@vitstudent.ac.in");

// Create a course

Course course = new Course("JAVA", "BCSE206", "Introduction to Java Programming", 3, "Dr. Sathyaraj");

// Display college student details

System.out.println("\nCollegeStudent Details:");

collegeStudent.displayStudentDetails();

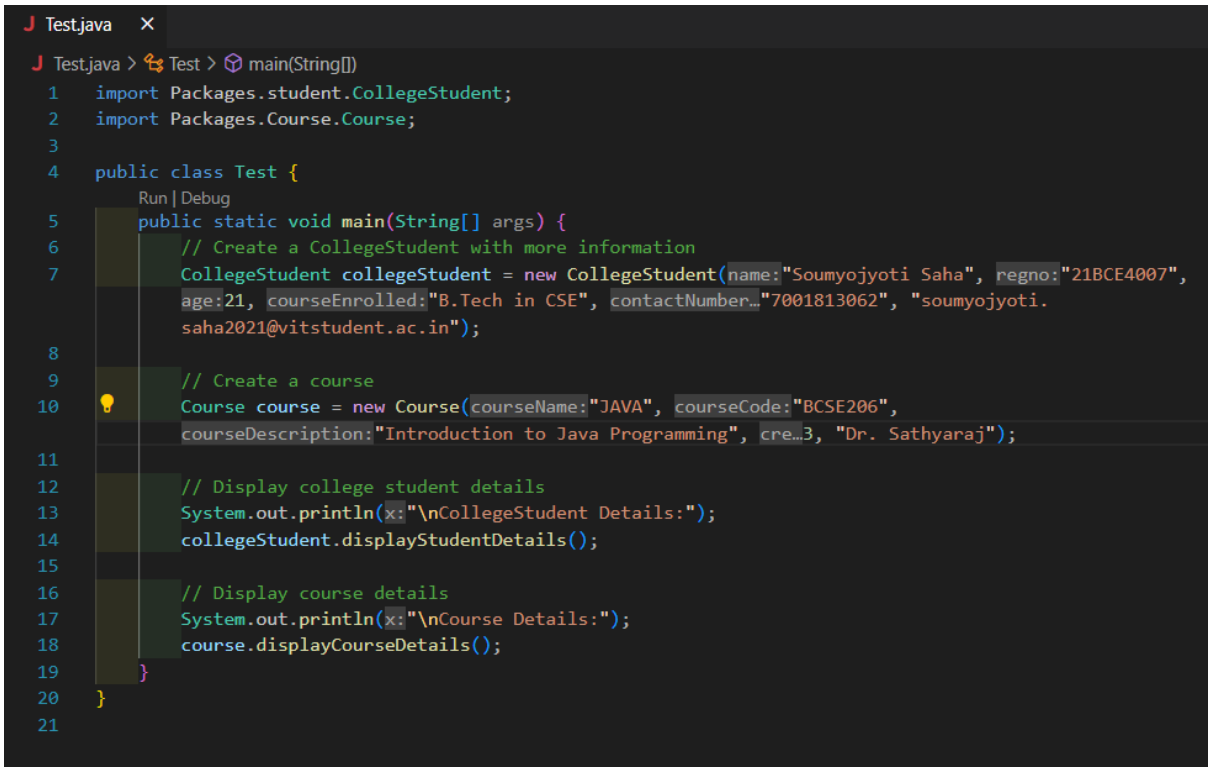
// Display course details

System.out.println("\nCourse Details:");

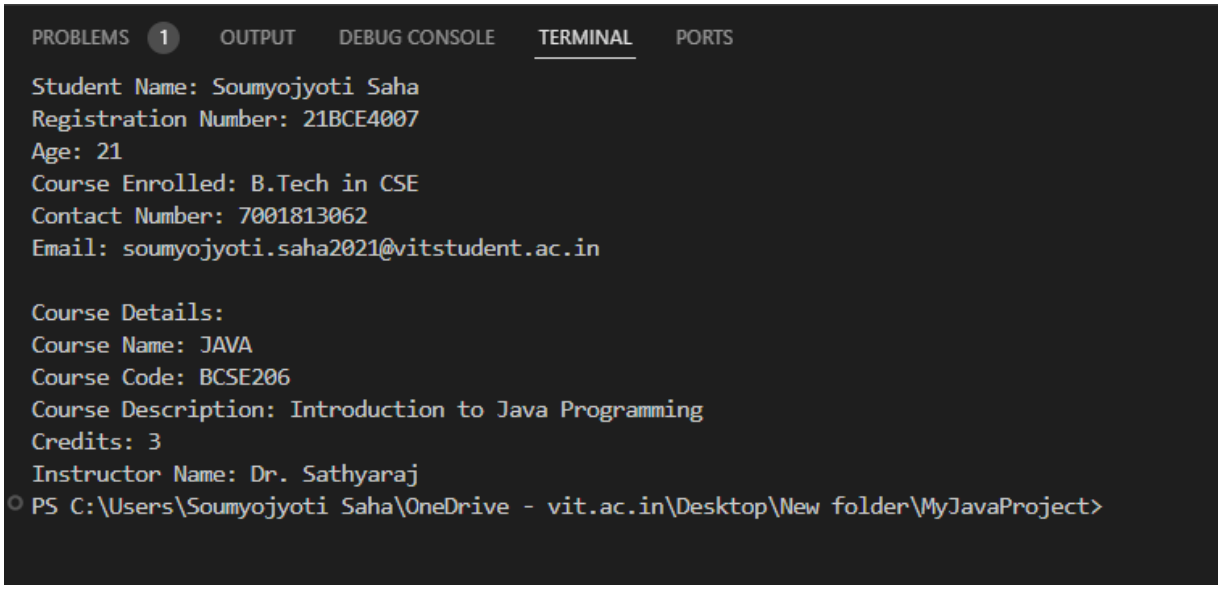
course.displayCourseDetails();

}

}



OUTPUT:


PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Student Name: Soumyojyoti Saha
Registration Number: 21BCE4007
Age: 21
Course Enrolled: B.Tech in CSE
Contact Number: 7001813062
Email: soumyojyoti.saha2021@vitstudent.ac.in
Course Details:
Course Name: JAVA
Course Code: BCSE206
Course Description: Introduction to Java Programming
Credits: 3
Instructor Name: Dr. Sathyaraj
PS C:\Users\Soumyojyoti Saha\OneDrive - vit.ac.in\Desktop\New folder\MyJavaProject>

END