VIT®

Vellore Institute of Technology

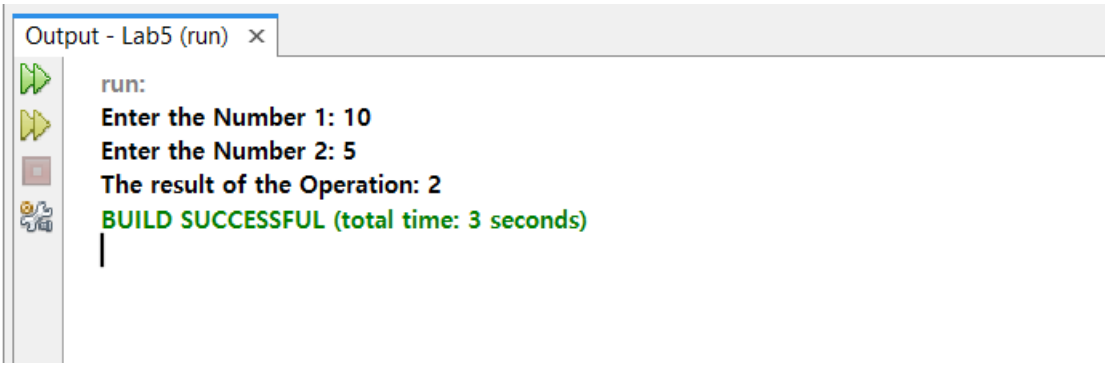
(Deemed to be University under section 3 of UGC Act, 1956)

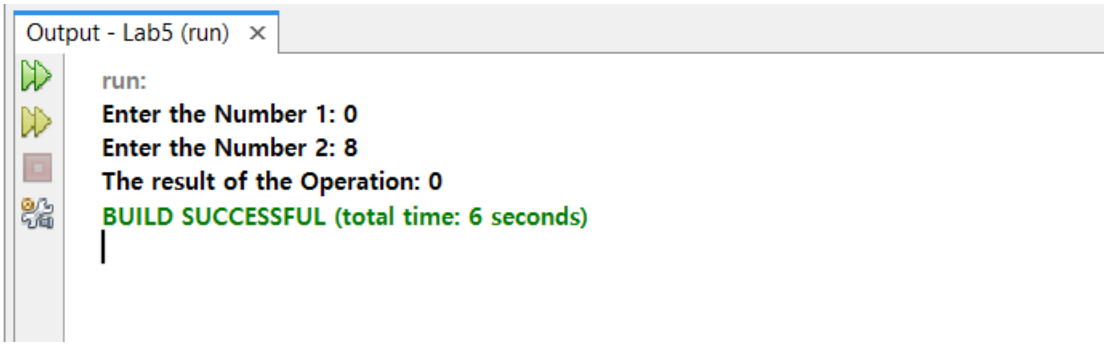
Programme	:	**BTech. CSE Core**	Semester	:	**Win 2021-22**
Course	:	**Java Programming**	Code	:	**CSE1007**
Faculty	:	**Dr. Pradeep K**	Slot	:	**L9+L10**
Name	:	**Hariket Sukesh Kumar Sheth**	Register No.	:	**20BCE1975**

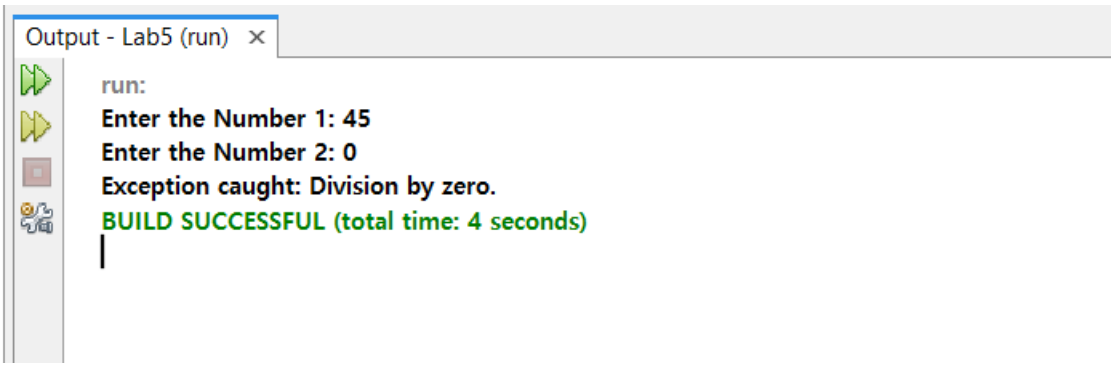
1. Complete the code segment to catch the ArithmeticException in the following, if any. On the occurrence of such an exception, your program should print "Exception caught: Division by zero." If there is no such exception, it will print the result of division operation on two integer values.

```
package lab5;
import java.util.*;

public class Lab5 {
    public static void main(String[] args) {
        int num1, num2;
        Scanner input = new Scanner(System.in);

        int result;
        System.out.print("Enter the Number 1: ");
        num1 = input.nextInt();
        System.out.print("Enter the Number 2: ");
        num2 = input.nextInt();

        try {
            result = num1 / num2;
            System.out.println("The result of the Operation: "+result);
        } catch (ArithmeticException e) {
            System.out.println("Exception caught: Division by zero.");
        }

    }
}
```

OUTPUT:





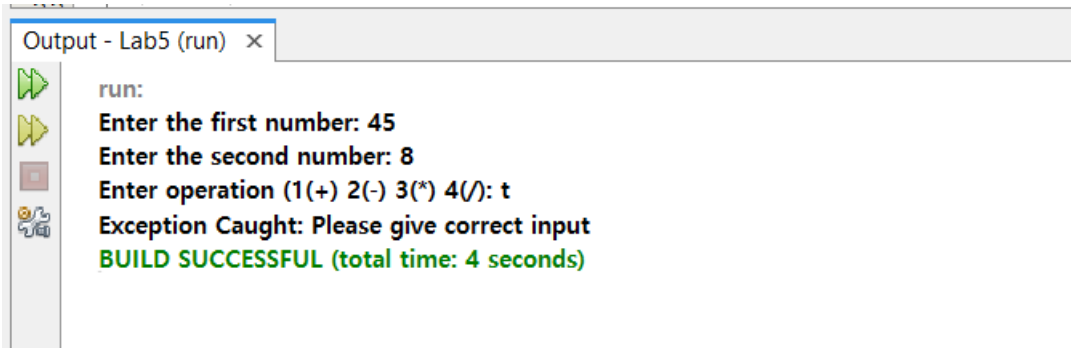


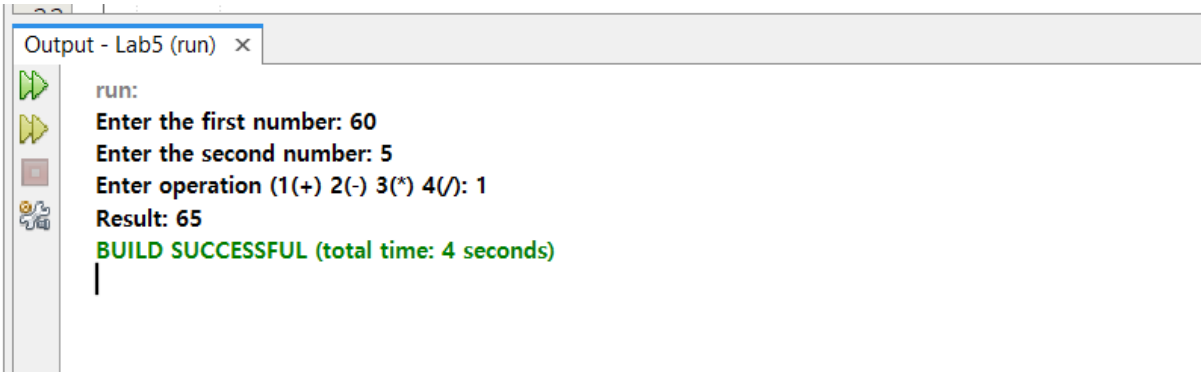
2. Write a calculator program. Note that the program terminates if any operand is nonnumeric. Write a program with an exception handler that deals with nonnumeric operands; your program should display a message that informs the user of the wrong operand type before exiting (hint : NumberFormatException)

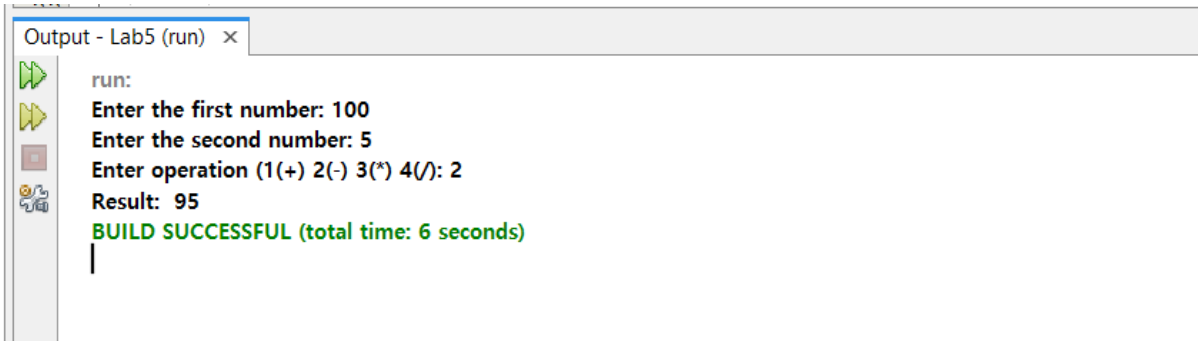
```
package lab5;
import java.util.*;

public class Lab5 {
    public static void main(String[] args) {
        int num1 = 0, num2 = 0;
        int operation;
        try {
            Scanner input = new Scanner(System.in);
            System.out.print("Enter the first number: ");
            num1 = input.nextInt();
            System.out.print("Enter the second number: ");
            num2 = input.nextInt();
            System.out.print("Enter operation (1(+) 2(-) 3(*) 4(/): ");
            operation = input.nextInt();
            switch (operation) {
                case 1:
                    System.out.println("Result: " + (num1 + num2));
                    break;
                case 2:
                    System.out.println("Result:  " + (num1 - num2));
                    break;
                case 3:
                    System.out.println("Result: " + (num1 / num2));
                    break;
                case 4:
                    System.out.println("Result: " + (num1 * num2));
                    break;
            }
        } catch (Exception e) {
```

```
            System.out.println("Exception Caught: Please give correct input");
        }

    }
}
```

OUTPUT:



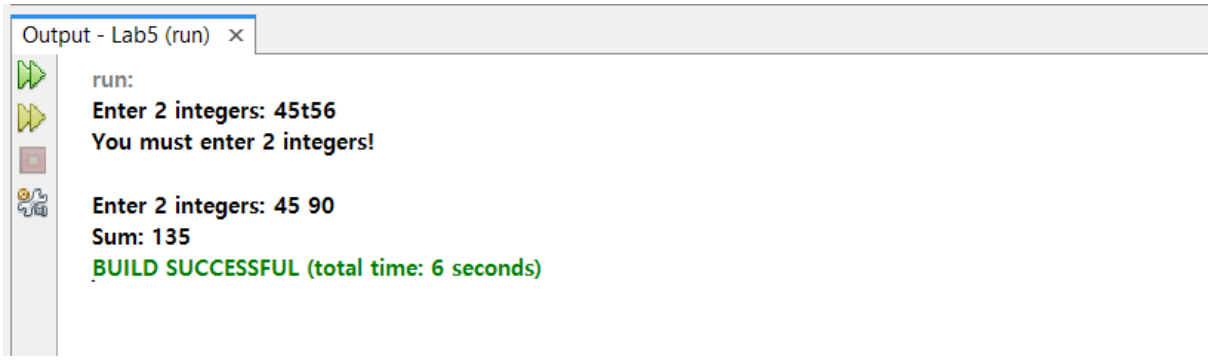




3. Write a program that prompts the user to read two integers and displays their sum. Your program should prompt the user to read the number again if the input is incorrect. Hint- InputMismatchException)

```java
package lab5;
import java.util.*;

public class Lab5 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num1 = 0, num2 = 0;
        while (true) {
            System.out.print("Enter 2 integers: ");
            try {
                num1 = sc.nextInt();
                num2 = sc.nextInt();
                System.out.println("Sum: " + (num1 + num2));
                break;
            } catch (InputMismatchException e) {
                System.out.println("You must enter 2 integers! \n");
                sc.nextLine();
            }
        }
    }
}
```

OUTPUT:

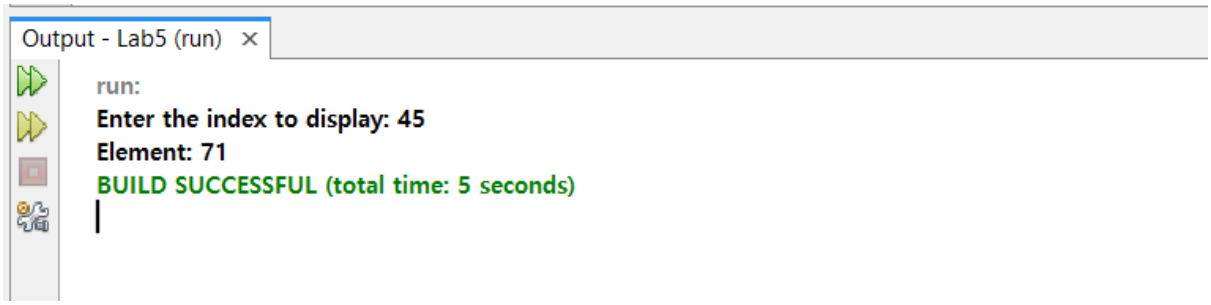


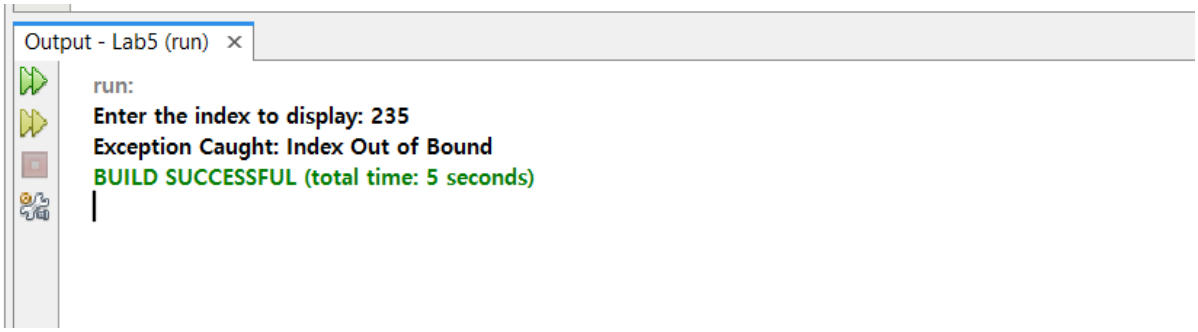
4. Write a program that meets the following requirements:

    1. Creates an array with 100 randomly chosen integers.
    2. Prompts the user to enter the index of the array, and then displays the corresponding element value. If the specified index is out of bounds, display the message Out of Bounds. (hint: ArrayIndexOutOfBoundsException)

```java
package lab5;
import java.util.*;

public class Lab5 {
    public static int[] getArray() {
        int[] array = new int[100];
        for (int i = 0; i < array.length; i++) {
            array[i] = (int)(Math.random() * 100) + 1;
        }
        return array;
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int[] array = getArray();
        System.out.print("Enter the index to display: ");
        try {
            System.out.println("Element: " + array[input.nextInt()]);
        } catch (ArrayIndexOutOfBoundsException ex) {
            System.out.println("Exception Caught: Index Out of Bound");
        }
    }
}
```

OUTPUT:



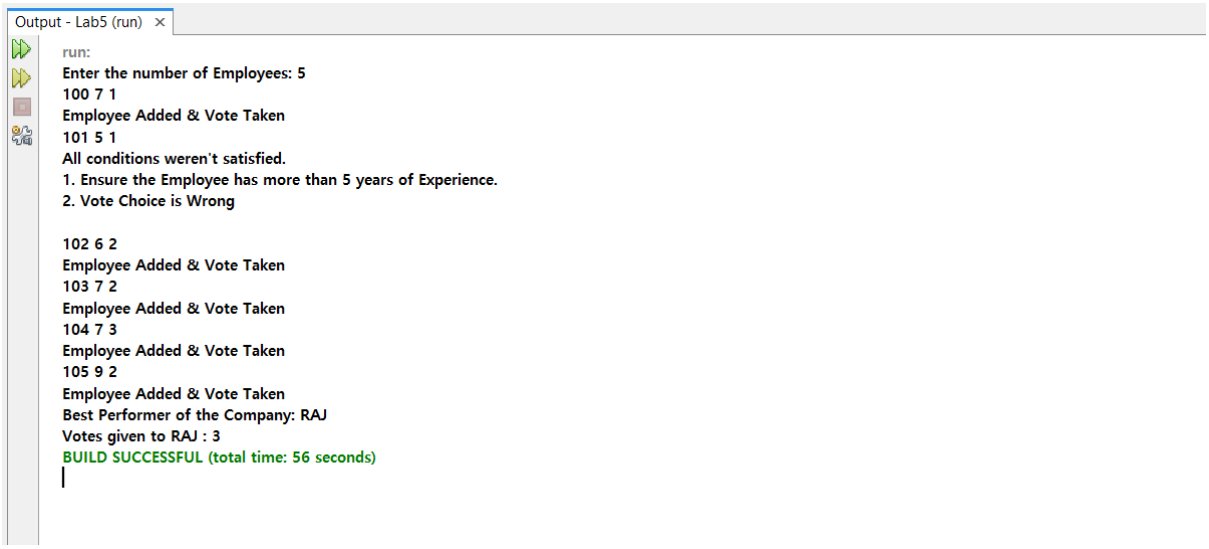


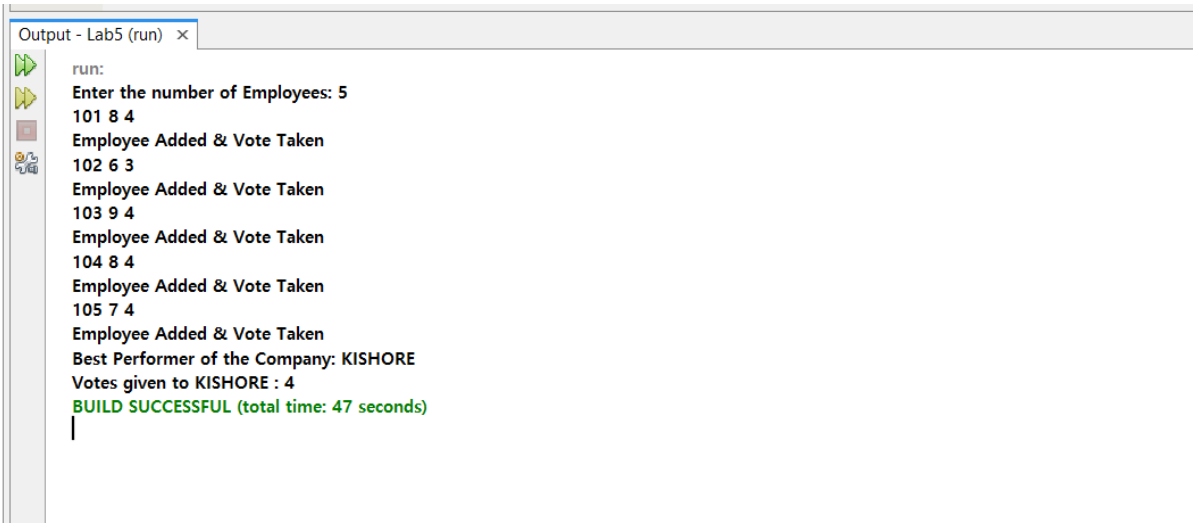
5. Develop a program that announces the best performer of a company among four contestants namely ARUN, RAJ, KUMAR, KISHORE. Accept the, the number of members, employee id, and experience of the team members as input. The team member can vote if the experience is greater than 5 years. Raise an exception if the experience is less than or equal to 5. Display the number of votes earned by best performer.

```java
package lab5;
import java.util.*;

class ExperienceException extends RuntimeException {
    ExperienceException(String str) {
        super(str);
    }
}

class Votes {
    public static int c_arun = 0, c_raj = 0, c_kishore = 0, c_kumar = 0;
    Votes(int vote) {
        switch (vote) {
            case 1:
                c_arun += 1;
                break;
            case 2:
                c_raj += 1;
                break;
            case 3:
                c_kumar += 1;
                break;
            case 4:
                c_kishore += 1;
                break;
        }
    }
    static String[] getResult() {
        String[] result = new String[2];
        if (c_arun > c_raj && c_arun > c_kishore && c_arun > c_kumar) {
            result[0] = "ARUN";
            result[1] = String.valueOf(c_arun);
        }
        if (c_kishore > c_raj && c_kishore > c_arun && c_kishore > c_kumar) {
            result[0] = "KISHORE";
            result[1] = String.valueOf(c_kishore);
        }
        if (c_kumar > c_raj && c_kumar > c_kishore && c_kumar > c_arun) {
            result[0] = "KUMAR";
            result[1] = String.valueOf(c_kumar);
        }
        if (c_raj > c_arun && c_raj > c_kishore && c_raj > c_kumar) {
            result[0] = "RAJ";
            result[1] = String.valueOf(c_raj);
        }
        return result;
    }
}
class Employee {
    int emp_id, experience;
    public Employee(int emp_id, int experience, int vote) throws ExperienceException {
        if (experience <= 5 || (vote < 1 || vote > 4)) {
            throw new ExperienceException("All conditions weren't satisfied.\n1. Ensure
the Employee has more than 5 years of Experience.\n" +
                "2. Vote Choice is Wrong\n");
        }
        this.emp_id = emp_id;
        this.experience = experience;
        Votes temp = new Votes(vote);
        System.out.println("Employee Added & Vote Taken");
    }
}

public class Lab5 {
```

```
    public static void main(String[] args) throws ExperienceException {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of Employees: ");
        int n = sc.nextInt();
        Employee e[] = new Employee[n];
        int emp_id, experience, vote, temp = 0;
        while (temp < n) {
            emp_id = sc.nextInt();
            experience = sc.nextInt();
            vote = sc.nextInt();
            try {
                e[temp] = new Employee(emp_id, experience, vote);
                temp += 1;
            } catch (Exception ex) {
                System.out.println(ex.getMessage());
            }
        }
        String[] result = Votes.getResult();
        System.out.println("Best Performer of the Company: " + result[0]);
        System.out.println("Votes given to " + result[0] + " : " + result[1]);
    }
}
```

OUTPUT:

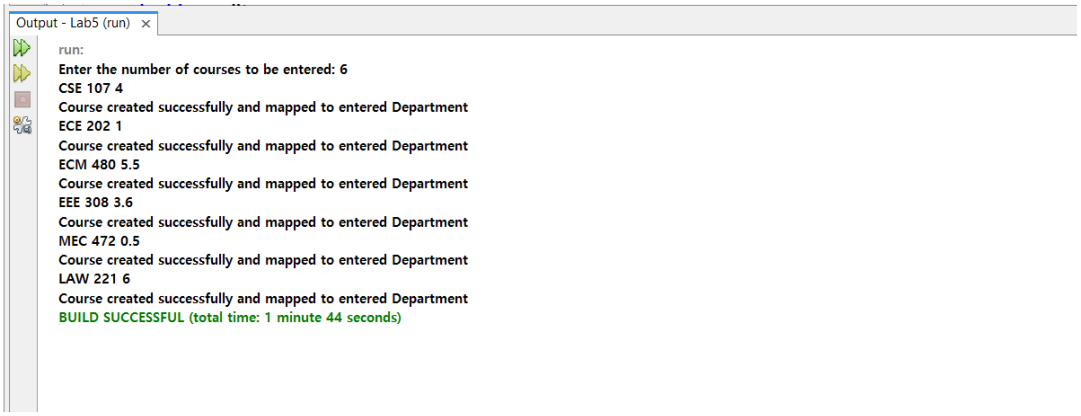


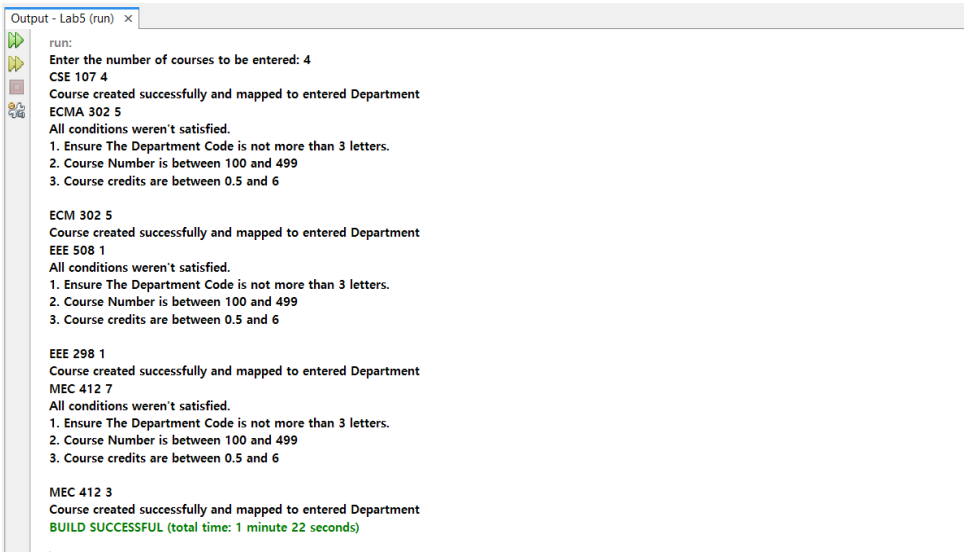


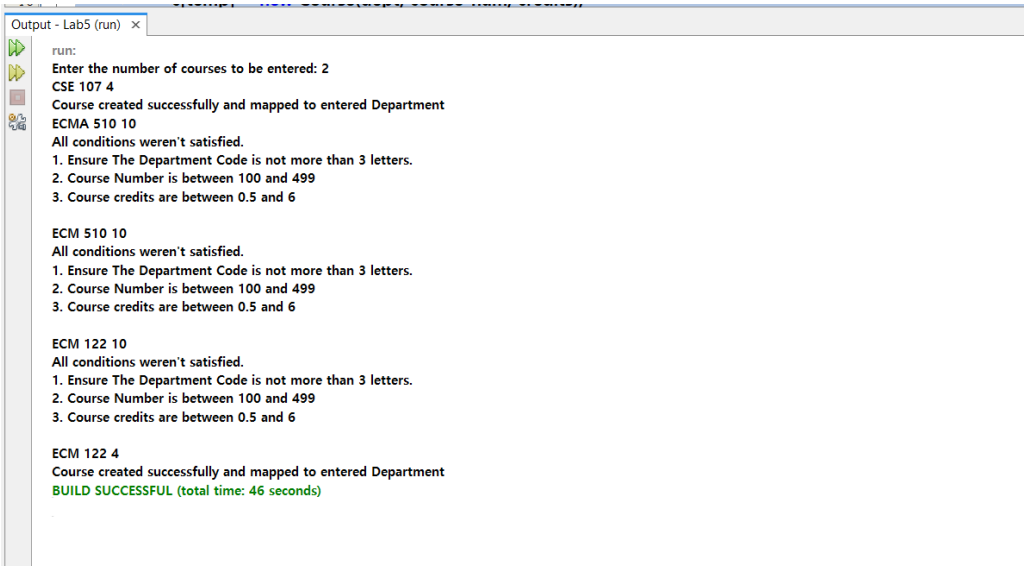
6. Create a **CourseException** class that extends Exception. Create a Course class with String that holds a college course's department (for example, CSE), a course number (for example, 101), and a number of credits (for example, 3) and whose getdata() member function requires values for each field., throw a CourseException if the department does not consist of three letters, if the course number does not consist of three digits between 100 and 499 inclusive, or if the credits are less than 0.5 or more than 6. Write an application and display an appropriate message when a Course object is created.

```
package lab5;
import java.util.*;

class CourseException extends RuntimeException {
    CourseException(String str) {
        super(str);
    }
}

class Course {
    String dept;
    int course_num;
    double credits;
    public Course(String dept, int course_num, double credits) throws CourseException {
        if (dept.length() != 3 || (course_num < 100 || course_num > 499) || (credits < 0.5
|| credits > 6)) {
            throw new CourseException("All conditions weren't satisfied.\n1. Ensure The
Department Code is not more than 3 letters.\n" +
                "2. Course Number is between 100 and 499\n3. Course credits are between 0.5
and 6\n");
        }
        this.dept = dept;
        this.course_num = course_num;
        this.credits = credits;
        System.out.println("Course created successfully and mapped to entered Department");
    }
}

public class Lab5 {
    public static void main(String[] args) throws CourseException {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of courses to be entered: ");
        int n = sc.nextInt();
        Course c[] = new Course[n];
        String dept;
        int course_num, temp=0;
        double credits;
        while(temp<n){
            dept = sc.next();
            course_num = sc.nextInt();
            credits = sc.nextDouble();
            try{
                c[temp] = new Course(dept, course_num, credits);
                temp +=1;
            }catch(Exception e){
                System.out.println(e.getMessage());
            }
        }
    }
}
```

OUTPUT:


Output - Lab5 (run)
run:
Enter the number of courses to be entered: 6
CSE 107 4
Course created successfully and mapped to entered Department
ECE 202 1
Course created successfully and mapped to entered Department
ECM 480 5.5
Course created successfully and mapped to entered Department
EEE 308 3.6
Course created successfully and mapped to entered Department
MEC 472 0.5
Course created successfully and mapped to entered Department
LAW 221 6
Course created successfully and mapped to entered Department
BUILD SUCCESSFUL (total time: 1 minute 44 seconds)


Output - Lab5 (run)
run:
Enter the number of courses to be entered: 4
CSE 107 4
Course created successfully and mapped to entered Department
ECMA 302 5
All conditions weren't satisfied.
1. Ensure The Department Code is not more than 3 letters.
2. Course Number is between 100 and 499
3. Course credits are between 0.5 and 6
ECM 302 5
Course created successfully and mapped to entered Department
EEE 508 1
All conditions weren't satisfied.
1. Ensure The Department Code is not more than 3 letters.
2. Course Number is between 100 and 499
3. Course credits are between 0.5 and 6
EEE 298 1
Course created successfully and mapped to entered Department
MEC 412 7
All conditions weren't satisfied.
1. Ensure The Department Code is not more than 3 letters.
2. Course Number is between 100 and 499
3. Course credits are between 0.5 and 6
MEC 412 3
Course created successfully and mapped to entered Department
BUILD SUCCESSFUL (total time: 1 minute 22 seconds)


Output - Lab5 (run)
run:
Enter the number of courses to be entered: 2
CSE 107 4
Course created successfully and mapped to entered Department
ECMA 510 10
All conditions weren't satisfied.
1. Ensure The Department Code is not more than 3 letters.
2. Course Number is between 100 and 499
3. Course credits are between 0.5 and 6
ECM 510 10
All conditions weren't satisfied.
1. Ensure The Department Code is not more than 3 letters.
2. Course Number is between 100 and 499
3. Course credits are between 0.5 and 6
ECM 122 10
All conditions weren't satisfied.
1. Ensure The Department Code is not more than 3 letters.
2. Course Number is between 100 and 499
3. Course credits are between 0.5 and 6
ECM 122 4
Course created successfully and mapped to entered Department
BUILD SUCCESSFUL (total time: 46 seconds)