Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering
Semester: (Spring, 2023), B.Sc. in CSE (Day)

LAB REPORT NO: 08
Course Title: Object Oriented Programing Lab
Course Code: CSE 202 Section: DE

Student Details

Name	ID
Md. Moshiur Rahman	221902324

Submission Date : 30/05/2023
Course Teacher's Name : **Dr. Muhammad Aminur Rahaman**

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …… **Signature:.....**
Comments:..... **Date:.....**

1. TITLE OF THE LAB EXPERIMENT:

I. Create an exception class named AgeOutOfRangeException extended from the class Exception. This class should contain a constructor which takes the user's age (ex. 40) as parameter. Will print following message when called, "You are older than the requested age (25 years), you are 40!!!".

II. Create an exception class named LowGpaException extended from the class Exception. This class should contain a constructor, with no parameter. The constructor will call the Exception class constructor with the message "Your GPA is not sufficient to apply for this job (2.5)".

III. Create a main class named GPA to prompt the user to enter his/her age and his GPA. The user application for a job will be rejected either if his age is greater than 25 years or his GPA is less than 2.5. You should declare two try-throw-catch blocks; one to handle the AgeOutOfRangeException and the other to handle the LowGpaException. If the user enters acceptable age and GPA, display the message "Your application is accepted and is under study".

2. OBJECTIVES

- ✓ Exception and Use of exception handling keywords.
- ✓ To build a User defined exception.

3. ALGORITHM

Step-1 : Define a class named AgeOutOfRangeException that extends the Exception class:

- Define a class named AgeOutOfRangeException that extends the Exception class:
- Inside the constructor, call the superclass constructor and pass a formatted message that indicates the age is older than the requested age.
- Override the toString() method to return the exception message.

Step-2 : . Define a class named LowGpaException that extends the

Exception class:

- Declare a constructor with no parameters.
- Inside the constructor, call the superclass constructor and pass a message indicating the GPA is not sufficient for the job.

Step-3 : Define the main class named GPA:

- Inside the main method, create an instance of the Scanner class to read user input.
- Prompt the user to enter their age.
- Read the user's age from the input.
- Use a try-catch block to handle the “AgeOutOfRangeException” ; Inside the try block, check if the user's age is greater than 25.
- If the age is greater than 25, throw an instance of AgeOutOfRangeException with the user's age as the parameter.
- If the age is acceptable, continue to the next step.
- Inside the catch block, catch the AgeOutOfRangeException and print the exception message.
- Prompt the user to enter their GPA.
- Read the user's GPA from the input.
- Use another try-catch block to handle the LowGpaException:
- Inside the try block, check if the user's GPA is less than 2.5.
- If the GPA is less than 2.5, throw an instance of LowGpaException.
- If the GPA is acceptable, continue to the next step.
- Inside the catch block, catch the LowGpaException and print the exception message.
- If both the age and GPA are acceptable, print the message "Your application is accepted and is under study".

Step-4 : End the main method and the class..

4. IMPLEMENTATION

```
class AgeOutOfRangeException extends Exception {
    private int age;

    public AgeOutOfRangeException(int age) {
        this.age = age;
    }

    @Override
    public String toString() {
        return "You are older than the requested age (25 years), you are " + age + "!!!";
    }
}

class LowGpaException extends Exception {
    public LowGpaException() {
        super("Your GPA is not sufficient to apply for this job (2.5)");
    }
}

public class GPA {
    public static void main(String[] args) {
        try {
            int age = getUserAge();
```

```
            double gpa = getUserGPA();

            if (age > 25) {
                throw new AgeOutOfRangeException(age);
            }
            if (gpa < 2.5) {
                throw new LowGpaException();
            }

            System.out.println("Your application is accepted and is under study.");
        } catch (AgeOutOfRangeException e) {
            System.out.println("Exception caught: " + e);
        } catch (LowGpaException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }

    private static int getUserAge() {

        return 40;
    }

    private static double getUserGPA() {
        return 2.0;
    }
}
```

5. TEST RESULT

If the user enters an age greater than 25:

```
Enter your age: 30
AgeOutOfRangeException: You are older than the requested age (25 years
    ), you are 30!!!
```

User enters an age older than 25

```
Exception caught: You are older than the requested age (25 years), you
    are 40!!!
```

If the user enters an age less than or equal to 25 and a GPA greater than or equal to 2.5:

```
Enter your age: 22
Enter your GPA: 3.2
Your application is accepted and is under study.
```

6. ANALYSIS & DISCUSSION

The program demonstrates the concept of custom exception classes by extending the base Exception class to create specialized exceptions for specific scenarios.
It showcases the use of try-catch blocks to handle exceptions gracefully and provide appropriate error messages to the user.
By using custom exceptions, the program improves code readability and maintainability by separating and handling specific error conditions in a structured manner.
The program allows for input validation and rejection of applications based on age and GPA criteria.
Overall, the program effectively handles exceptions related to age and GPA and provides informative error messages to the user, ensuring the application process follows the specified requirements.