Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering
Semester:(Spring, Year:2024), B.Sc. in CSE (Day)

Lab Report NO: 06
Course Title: Object-Oriented Programming Lab
Course Code: CSE 202 Section: 223 D9

Lab Experiment Name: Exception, Custom Exception, and use of Exception Handling keywords

Student Details

Name		ID
1.	Promod Chandra Das	231002005

Lab Date :
Submission Date :
Course Teacher's Name : Noyan Ali

Lab Report Status

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

TITLE OF THE LAB REPORT EXPERIMENT: Exception, Custom Exception, and use of Exception Handling keywords.

OBJECTIVES/AIM :

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

Problem analysis:

The java.lang.Throwable class is the superclass of all errors and exceptions in the Java language.

Errors

Errors represent irrecoverable conditions such as Java virtual machine (JVM) running out of memory, memory leaks, stack overflow errors, library incompatibility, infinite recursion, etc. Errors are usually beyond the controlof the programmer and we should not try to handle errors.

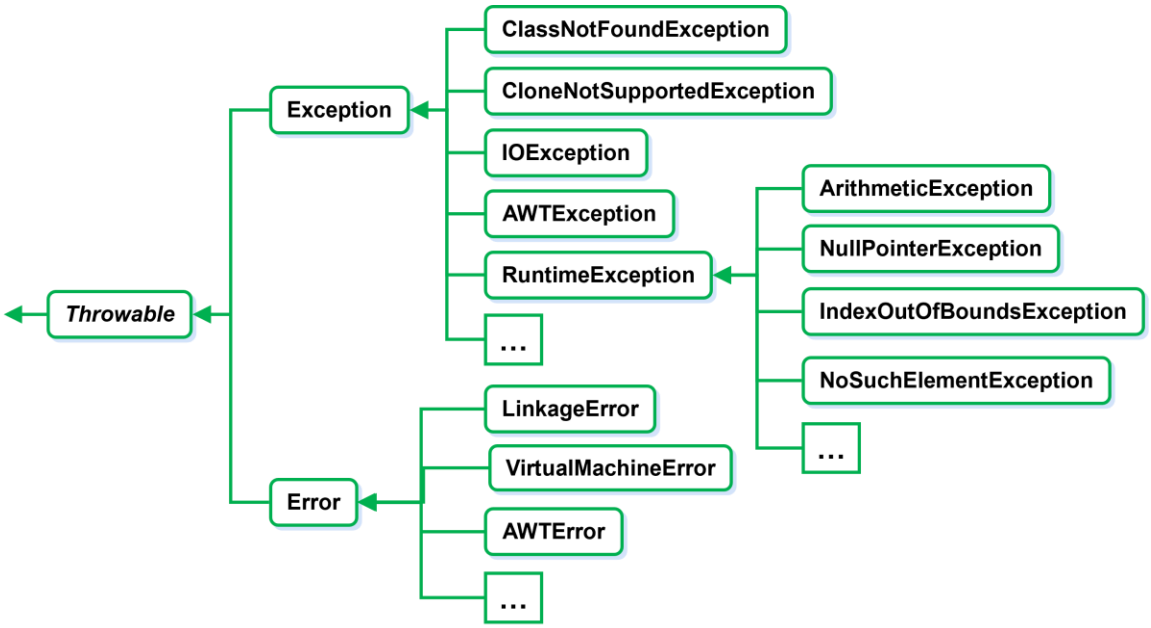

Figure 1: Java Exception class hierarchy

➢ Exception

An exception is an unexpected event that occurs during program execution. It affects the flow of the program instructions which can cause the program to terminate abnormally. Exceptions can be caught and handled bythe program. An exception can occur for many reasons, such as :

- Invalid user input
- Device failure
- Loss of network connection
- Physical limitations (out of disk memory)
- Code errors
- Opening a file that is unavailable

There are two common types of exception.

➢ Runtime Exception

A runtime exception happens due to a programming error. They are also known as unchecked exceptions. Theseexceptions are not checked at compile-time but at run-time. Some of the common runtime exceptions are:

- Improper use of an API - IllegalArgumentException
- Null pointer access (missing the initialization of a variable) - NullPointerException
- Out-of-bounds array access - ArrayIndexOutOfBoundsException
- Dividing a number by 0 – ArithmeticException

➢ IO Exception

An IOException is also known as a checked exception. They are checked by the compiler at the compile-timeand the programmer is prompted to handle these exceptions.

Some of the examples of checked exceptions are:

- Trying to open a file that doesn't exist results in FileNotFoundException
- Trying to read past the end of a file

When an exception occurs within a method, it creates an object. This object is called the exception object. It contains information about the exception such as the name and description of the exception and state of the program when the exception occurred. Exception handling refers to performing an action in response to an exception. For example,

- Exit program (abort)
- Deal with exception and continue
- Print error message
- Request new data
- Retry action

To understand exception handling properly, we need clear understanding of a few keywords.

- **Keywords**

Java exception handling is managed by via five keywords:

- **try**

Program statements to monitor are contained within a try block. If an exception occurs within the try block,it is thrown.

- **catch**

Code within catch block catch the exception and handle it. The scope of a catch clause is restricted to thosestatements specified by the immediately preceding try statement.

- **finally**

finally block is a block that is used to execute important code such as closing connection, stream etc. Javafinally block is always executed whether exception is handled or not.

- **throw**

It is possible for your program to to throw an exception explicitly using *throw ThrowableInstance*. Here, ThrowableInstance must be an object of type Throwable or a subclass Throwable

- **throws**

If a method is capable of causing an exception that it does not handle, it must specify this behavior so thatcallers of the method can guard themselves against that exception.

Lab Exercise (Submit as a report)

The task is divided into 3 parts.

1. 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!!!".
2. 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)".
3. 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".

Answer to the Q No: 01:

1. 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!!!".

Answer:-

```
public class Main {
   public static void main(String[] args) {
      int userAge = 40;  // Example user age
      int maxAge = 25;   // The requested age limit


      try {
         checkAge(userAge, maxAge);
      } catch (AgeOutOfRangeException e) {
         System.out.println(e.getMessage());
      }
   }


   public static void checkAge(int age, int maxAge) throws AgeOutOfRangeException {
      if (age > maxAge) {
         throw new AgeOutOfRangeException(age);
```

```
        } else {
            System.out.println("Age is within the acceptable range.");
        }
    }
}
```

Answer to the Q No: 02

2. 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)".

Answer:-

```
public class Main {
    public static void main(String[] args) {
        double userGpa = 2.0;
        double requiredGpa = 2.5;

        try {
            checkGpa(userGpa, requiredGpa);
        } catch (LowGpaException e) {
            System.out.println(e.getMessage());
        }
    }

    public static void checkGpa(double gpa, double requiredGpa) throws LowGpaException {
        if (gpa < requiredGpa) {
            throw new LowGpaException();
        } else {
            System.out.println("Your GPA meets the requirements.");
        }
    }
}
```

Answer to the Q No: 03

3.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 applicationis accepted and is under study".

Answer:-

```
import java.util.Scanner;

public class GPA {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter your age: ");
            int age = scanner.nextInt();
            checkAge(age);
        } catch (AgeOutOfRangeException e) {
            System.out.println(e.getMessage());
            return;
        }

        try {
            System.out.print("Enter your GPA: ");
            double gpa = scanner.nextDouble();
            checkGpa(gpa);
        } catch (LowGpaException e) {
            System.out.println(e.getMessage());
            return;
        }

        System.out.println("Your application is accepted and is under study.");
    }

    public static void checkAge(int age) throws AgeOutOfRangeException {
        int maxAge = 25;
        if (age > maxAge) {
            throw new AgeOutOfRangeException(age);
        }
    }

    public static void checkGpa(double gpa) throws LowGpaException {
        double requiredGpa = 2.5;
        if (gpa < requiredGpa) {
            throw new LowGpaException();
        }
```

```
    }
}
```

➢ **Discussion & Conclusion:**

Exception handling in Java involves managing runtime errors using try, catch, throw, and finally blocks. A standard Exception is a general error signal, while a custom exception like LowGpaException or AgeOutOfRangeException provides specific error messages tailored to particular issues. Custom exceptions enhance code clarity and error diagnosis by indicating precise problems and possible solutions.