Name: Gaurav Kishor Patil

Roll no:54 DIV: 2

Batch: C

Experiment No. 13
User Defined Exception.
Date of Performance: 13/10/23
Date of Submission: 17/10/23

Aim:- To implement User Defined Exception.

Objective:- To implement user defined exception through a simple java program to check whether the temperature is low or normal.

Theory :- An exception is an issue (run time error) that occurred during the execution of a program. When an exception occurred the program gets terminated abruptly and, the code past the line that generated the exception never gets executed.

Java provides us the facility to create our own exceptions which are basically derived classes of Exception. Creating our own Exception is known as a custom exception or user-defined exception. Basically, Java custom exceptions are used to customize the exception according to user needs. In simple words, we can say that a User-Defined Exception or custom exception is creating your own exception class and throwing that exception using the 'throw' keyword.
For example, MyException in the below code extends the Exception class.

Why use custom exceptions?
Java exceptions cover almost all the general types of exceptions that may occur in the programming. However, we sometimes need to create custom exceptions.

Following are a few of the reasons to use custom exceptions:

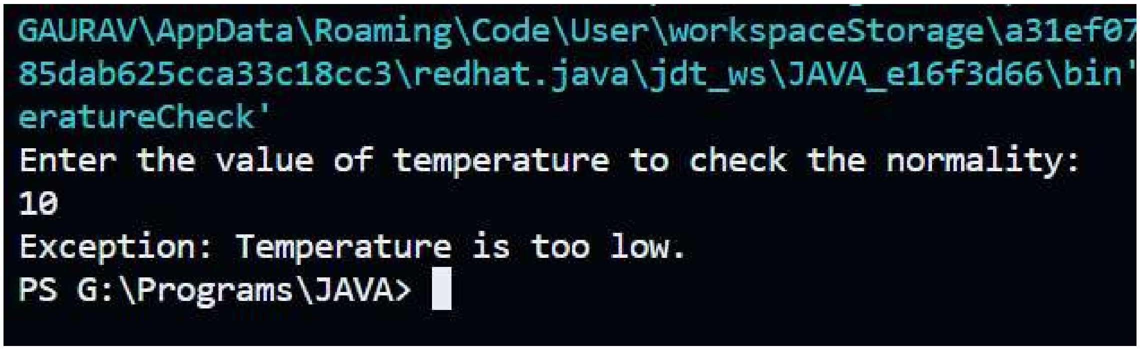
To catch and provide specific treatment to a subset of existing Java exceptions.
Business logic exceptions: These are the exceptions related to business logic and workflow. It is useful for the application users or the developers to understand the exact problem.
In order to create a custom exception, we need to extend the Exception class that belongs to java.lang package.

Code:-

```
import java.util.Scanner;

class TemperatureException extends Exception {

 public TemperatureException(String message) {

 super(message);

 }

}
```

```
public class TemperatureCheck {

 public static void main(String[] args) {

 Scanner obj = new Scanner(System.in);

 try {

 System.out.println("Enter the value of temperature to check the normality:");

 int a = obj.nextInt(); // Replace with the actual temperature value

 if (a < 20) {

 throw new TemperatureException("Temperature is too low.");

}

 System.out.println("Temperature is normal.");

 } catch (TemperatureException e) {

 System.err.println("Exception: " + e.getMessage());

 }

 }

}
```

Output:



```
GAURAV\AppData\Roaming\Code\User\workspaceStorage\a31ef07
85dab625cca33c18cc3\redhat.java\jdt_ws\JAVA_e16f3d66\bin'
eratureCheck'
Enter the value of temperature to check the normality:
10
Exception: Temperature is too low.
PS G:\Programs\JAVA>
```

Conclusion :-

In Java, user-defined exceptions are custom exceptions created by extending the built-in Exception class or one of its subclasses. These exceptions allow developers to handle specific error scenarios in their code more effectively by providing meaningful error messages and behavior. By creating custom exceptions, developers can improve code readability, maintainability, and error reporting, enabling more robust and structured error handling in Java applications.