Ameya Baropatre
Roll No-06 SE-DS

10/10

Experiment No - 13

Aim - WAP to implement user defined exception.

Resource Required → Notepad, JDK1.8, wordpad, paint, stationary.

Theory -

User defined exceptions in Java provide a way to create custom error condition that are specified to your application's logic. This allow for more granular control over error handling and enhance the Robustness and clarity of your code. Below is an exception of key concept illustrate in program.

Key Concept -

1) Exception handing → Exception handling in Java provide a structured way to manage runtime errors ensuring program can respond to ~~poros~~ prior to without crashing.

2) Creating a user-defined Exception → The user defined exception is created by extending the Exception class in this case the class in exception extends exception.

3) Finally - The finally block is used to heare exception. it must be programmed.

4) Throw - The throw keyword is use to declare execution It specify that the time may occur an exception in method it doesn't throw an exception. It is always used with method signature

Java by Block →

Java by block is used to enclose code that might throw an exception it must be used within method.

If an exception occurs at particular statement in try block rest of the block & code will not execute. It is recommended not to keep & code in try block.

~~Jawa Must~~

Java try block must be followed by either catch or finally black.

Syntax of Java try catch → try {
//code that may throw an exception {
catch exception - class - ref () }}

Syntax of try finally block
try {
//code that may throw an exception
}

```
class my Exception extende Exception
 {
    Public myException (string s)
        { super (s);
        } }
```

3) Throwing the exception →
In the main method, the program intentionally throw an instance of myException using throw keyword. This stimulates an error condition that program should handle.
The string "GeeksGeeks" is passed to condoctor of myException which sets exception message throw new
MyException("GeeksGeeks");

4) Catching the exception →
In catch block, the program prints a message indicating that the exception was caught and retrives the message from myException object using getmessage ().

```
Catch (MyException e)
   {
      System.out.println("caught");
      System.out.println(e getmessage ());
      my Exception object
   }
```

Conclusion
~~Conduction~~ → The program exemplifies the fundamental principal of exception handling in Java, focusing on user-defined exception. By creating a custom exception class, the program can effectively ~~con~~ signal and manage specific error condition.
We implement a program using user defined exception succesfully.

1/10/24