Experiment No. 08

Title:

Write a class having two integer data members. Provide facility to add, subtract, multiply and divide these numbers. If addition goes above 1000, it generates TooLongAddition exception. If subtraction is below 0, it generates Negative Answer exception. If multiplication is above 5000, it generates TooLongMultiplication exception.

Objectives:

1. To learn basics of Exception Handling

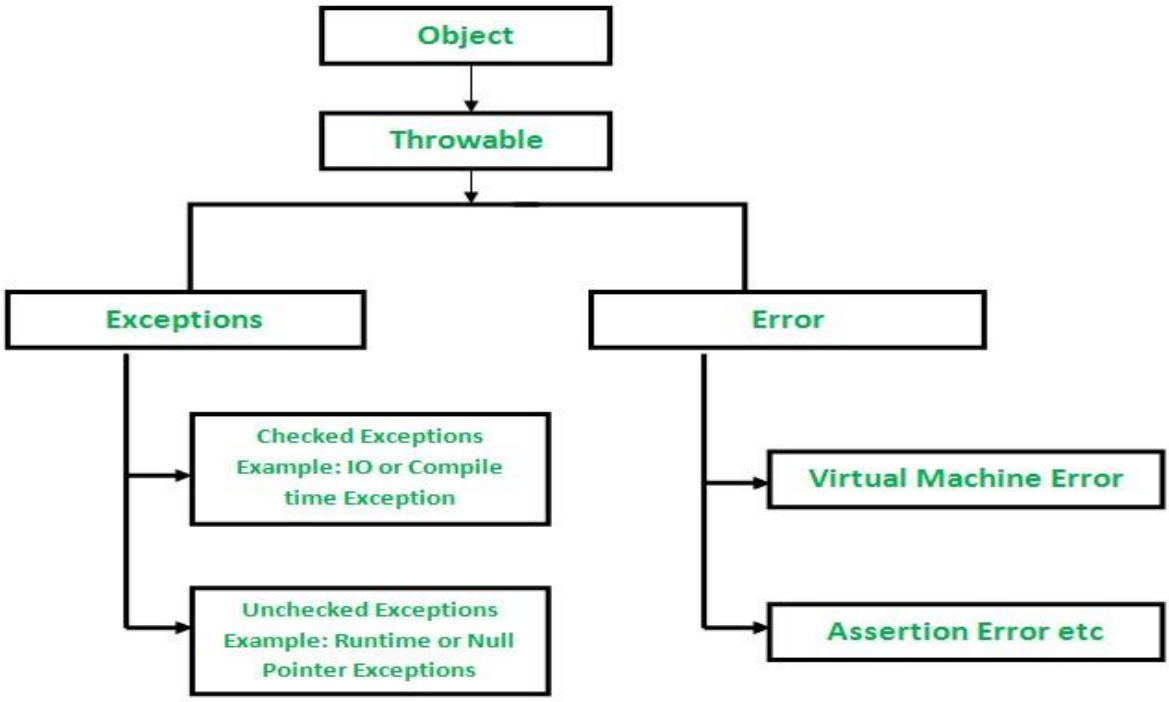
Theory:

An **exception** is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time that disrupts the normal flow of the program's instructions**.**

Error: An Error indicates serious problem that a reasonable application should not try to catch.

Exception: Exception indicates conditions that a reasonable application might try to catch.

Exception Hierarchy



.

Default Exception Handling: Whenever inside a method, if an exception has occurred, the method creates an Object known as Exception Object and hands it off to the run-time system(JVM). The exception object contains name and description of the exception, and current state of the program where exception has occurred. Creating the Exception Object and handling it to the run-time system is called throwing an Exception.There might be the list of the methods that had been called to get to the method where exception was occurred. This ordered list of the methods is called Call Stack.Now the following procedure will happen.

The run-time system searches the call stack to find the method that contains block of code that can handle the occurred exception. The block of the code is called Exception handler.
The run-time system starts searching from the method in which exception occurred proceeds through call stack in the reverse order in which methods were called.
If it finds appropriate handler then it passes the occurred exception to it. Appropriate handler means the type of the exception object thrown matches the type of the exception object it can handle.
If run-time system searches all the methods on call stack and couldn't have found the appropriate handler then run-time system handover the Exception Object to default exception handler , which is part of run-time system. This handler prints the exception information in the following format and terminates program abnormally.

Customized Exception Handling: Java exception handling is managed via five keywords: try, catch, throw, throws, and finally. Briefly, here is how they work. Program statements that you think can raise exceptions are contained within a try block. If an exception occurs within the try block, it is thrown. Your code can catch this exception (using catch block) and handle it in some rational manner. System-generated exceptions are automatically thrown by the Java run-time system. To manually throw an exception, use the keyword throw. Any exception that is thrown out of a method must be specified as such by a throws clause. Any code that absolutely must be executed after a try block completes is put in a finally block.
-The user (programmer) can also create his own exceptions which are called user-defined exceptionsThe user should create an **e**xception class as a subclass to Exception class. Because all exception are subclasses of Exception class.

class MyException extends Exception

-The user can create a parameterized constructor with a string as a parameter. He can use to store exception details. He can call super class (Exception) constructor from this and send the string there.

```
MyException(String str)
{
super (str);
}
```

-When the user wants to raise his own exception, he should create an object to his exception class and throw it using throw clause, as:

MyException me=new MyException("Exception details"); throw me;

Algorithm:

1] Create user-defined exception class MyException that extends Exception class 2] Define parameterized constructor with string as parameter.

3] Create another class operation.

4] Define methods for addition, Subtraction, multiply and divide in operation class and perform operations in it.

5] If addition goes above 1000, generate TooLongAddition exception.

6] If subtraction is below 0, generate NegativeAnswer exception.

7] If multiplication is above 5000, generate TooLongMultiplication exception.

8] In main method create object of class operation and call the methods. Provide exception handling code to handle exceptions

Key concepts: Exception, checked, unchecked exception, try, catch, throw, throws, finally.