Department of Software Engineering
Mehran University of Engineering and Technology, Jamshoro

Course: SWE121 – Object Oriented Programming			
Instructor	Mariam Memon	**Practical/Lab No.**	09
Date		**CLOs**	3
Signature		**Assessment Score**	

Topic	**Exception Handling**
Objectives	To resolve runtime errors using exception handling mechanism. To use try catch and finally blocks

Lab Discussion: Theoretical concepts and Procedural steps

What is an Exception?

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 vs Exception

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

All exception and errors types are sub classes of class **Throwable**, which is base class of hierarchy.One branch is headed by **Exception**. This class is used for exceptional conditions that user programs should catch. NullPointerException is an example of such an exception.Another branch, **Error** are used by the Java run-time system(JVM) to indicate errors having to do with the run-time environment itself(JRE). StackOverflowError is an example of such an error.

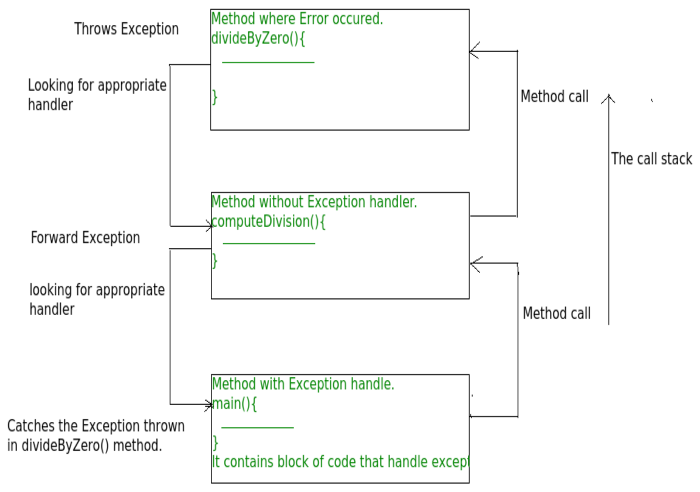
How JVM handle an Exception?

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**.

```
Exception in thread "xxx" Name of Exception : Description
... ...... ..  // Call Stack
```

See the below diagram to understand the flow of the call stack.


Throws Exception
Looking for appropriate handler
Forward Exception
looking for appropriate handler
Catches the Exception thrown in divideByZero() method.
Method where Error occured.
divideByZero(){
}
Method without Exception handler.
computeDivision(){
}
Method with Exception handle.
main(){
}
It contains block of code that handle except
Method call
Method call
The call stack

The call stack and searching the call stack for exception handler.

How to use try-catch clause

```
try {
// block of code to monitor for errors
// the code you think can raise an exception
}
catch (ExceptionType1 exOb) {
// exception handler for ExceptionType1
}
catch (ExceptionType2 exOb) {
// exception handler for ExceptionType2
}
// optional
finally {
// block of code to be executed after try block ends
}
```

throw

The throw keyword in Java is used to explicitly throw an exception from a method or any block of code. We can throw either checked or unchecked exception. The throw keyword is mainly used to throw custom exceptions.

Syntax:

```
throw Instance
Example:
throw new ArithmeticException("/ by zero");
```

throws

throws is a keyword in Java which is used in the signature of method to indicate that this method might throw one of the listed type exceptions. The caller to these methods has to handle the exception using a try-catch block.

Syntax:

```
type method_name(parameters) throws exception_list
exception_list is a comma separated list of all the
exceptions which a method might throw.
```

Lab Tasks

Task 1: Run the following code and explain the output

```
class Test
{
        String str = "a";
        void A()
        {
                try
                {
                        str +="b";
                        B();
                }
                catch (Exception e)
                {
                        str += "c";
                }
        }
        void B() throws Exception
        {
                try
                {
                        str += "d";
                        C();
                }
                catch(Exception e)
                {
                        throw new Exception();
                }
                finally
                {
                        str += "e";
                }
                str += "f";
        }
        void C() throws Exception
        {
                throw new Exception();
        }
        void display()
        {
                System.out.println(str);
        }
        public static void main(String[] args)
        {
                Test object = new Test();
                object.A();
                object.display();
        }
}
```

Task 2: Create a custom exception class InvalidAgeException.

Next create a simplified VotingMachine Class which has an array of candidate names and the following methods:

a) addCandidates(String names[]) to initialize the instance variable
b) castVote(int voterAge, String votesToCandidate)
   if voterAge is less than 18 this method throws an invalidAgeException otherwise increases the count of votes for the given candidate
c) printResults() to display the votes each candidate has received

In the main method catch the InvalidAgeException

Task 3: You are required to compute the power of a number by implementing a calculator. Create a class MyCalculator which consists of a single method long power(int, int). This method takes two integers, ***n*** and ***p***, as parameters and finds $\boldsymbol{n^p}$. If either ***n*** or ***p*** is negative, then the method must throw an exception which says "n or p should not be negative". Also, if both and are zero, then the method must throw an exception which says "n or p should not be zero"
Note: Use for loop to calculate the power