#### Experiment No: 09

**Aim:-** To write a java program which use try and catch for exception handling.

**Software Used:**- Eclipse

**Theory:**

## Exception

**Exception** is an unwanted or unexpected event, which occurs during the execution of a program. An abnormal event in a program is called Exception. Exception may occur at compile time or at runtime. Exceptions which occur at compile time are called Checked exceptions.

e.g.: ClassNotFoundException, NoSuchMethodException, NoSuchFieldException etc.

Exceptions which occur at run time are called Unchecked exceptions.

eg: ArrayIndexOutOfBoundsException, ArithmeticException, NumberFormatException etc.

#### Hierarchy of Exception classes

An error in a program is called bug. Removing errors from program is called debugging. There are basically three types of errors in the Java program:

#### Compile time errors:

* Errors which occur due to syntax or format is called compile time errors.
* These errors are detected by java compiler at compilation time. Desk checking is solution for compile-time errors.

#### Runtime errors:

* These are the errors that represent computer inefficiency.
* Insufficient memory to store data or inability of the microprocessor to execute
* some statement is examples to runtime errors.
* Runtime errors are detected by JVM at runtime.

#### Logical errors:

* These are the errors that occur due to bad logic in the program.
* These errors are rectified by comparing the outputs of the program manually.

#### Task:-

If the programmer suspects any exception in program statements, he should write them inside When there is an exception in try block JVM will not terminate the program abnormally. JVM stores exception details in an exception stack and then JVM jumps into catch block. The programmer should display exception details and any message to the user in catch block

#### catch ( ExceptionClass obj)

**{**

#### statements;

**}**

#### Programmer should close all the files and databases by writing them inside finally block. Finally block is executed whether there is an exception or not.

#### finally

**{**

#### statements;

**}**

#### Performing above tasks is called Exception Handling.

**Exception Handling keyword:**

#### Keywords: try, catch, finally, throw, throws.

**Try:**

The program statement that to be monitored for exception are contain within the try block.

If an exception are contain within try block, it is thrown.

#### try

**{**

#### statements;

**}**

#### Catch:

your code can catch the exception using keyword ‘catch ‘ & handle it in rational manner,

#### catch ( ExceptionClass obj)

**{**

#### statements;

**}**

The JVM firstly checks whether the exception is handled or not.

If exception is not handled, JVM provides a default exception handler that performs the following tasks: Prints out exception description. Prints the stack trace (Hierarchy of methods where the exception occurred). Causes the program to terminate. But if exception is handled by the application programmer, normal flow of the application is maintained i.e. rest of the code is executed**.**

**throw:**

It is the system generated exceptions are handle automatically thrown by the java runtime system. To manually throw an exception a throw keyword is used.

#### finally:

**Any code that must be executed before a method written is put in a finally block.**

#### finally

**{**

#### statements;

**}**

#### //block of code to executed before try block ends

The throws keyword is used in method declaration, in order to explicitly specify the exceptions that a particular method might throw. When a method declaration has one or more exceptions defined using throws clause then the method-call must handle all the defined exceptions.

When defining a method you must include a throws clause to declare those exceptions that might be thrown but doesn’t get caught in the method. If a method is using throws clause along with few exceptions then this implicitly tells other methods that – “ If you call me, you must handle these exceptions that I throw”.

#### Program :

#### Output:

**Conclusion:**

**Questions:**

1. How are exceptions handled in Java?
2. What is the difference between exception and error in Java?
3. Which block always executes whether or not an exception occurs?
4. Which statement is used to catch all types of exceptions?
5. Try block b) Catch block c) throw keyword d)finally block