***Dt : 29/10/2022***

***\*imp***

***define "Throwable"?***

***=>"Throwable" class is from java.lang package and which is root of***

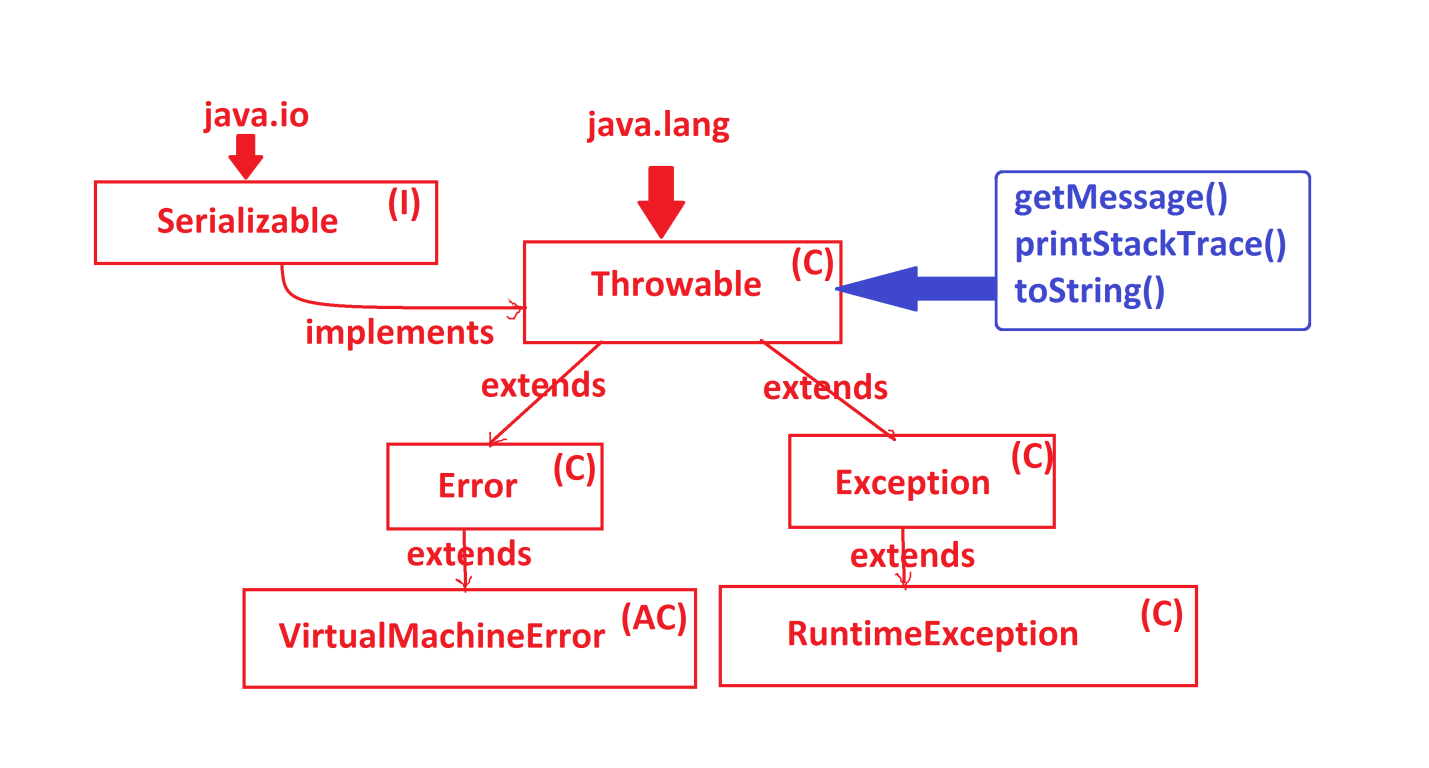
***Exception Handling process.***

***=>This "Throwable" class is extended into the following two SubClasses:***

***1.Error***

***2.Exception***

***Hierarchy of "Throwable":***

******

***-------------------------------------------------***

***1.Error:***

***=>The distrubance which is occured from the environment is known as "error"***

***=>"java.lang.Error" class is the PClass or SuperClass of all the errors***

***raised from the environment.***

***=>There is no separate process to handle errors.***

***2.Exception:***

***=>"java.lang.Exception" is the PClass or SuperClass of all the exceptions***

***raised from the application.***

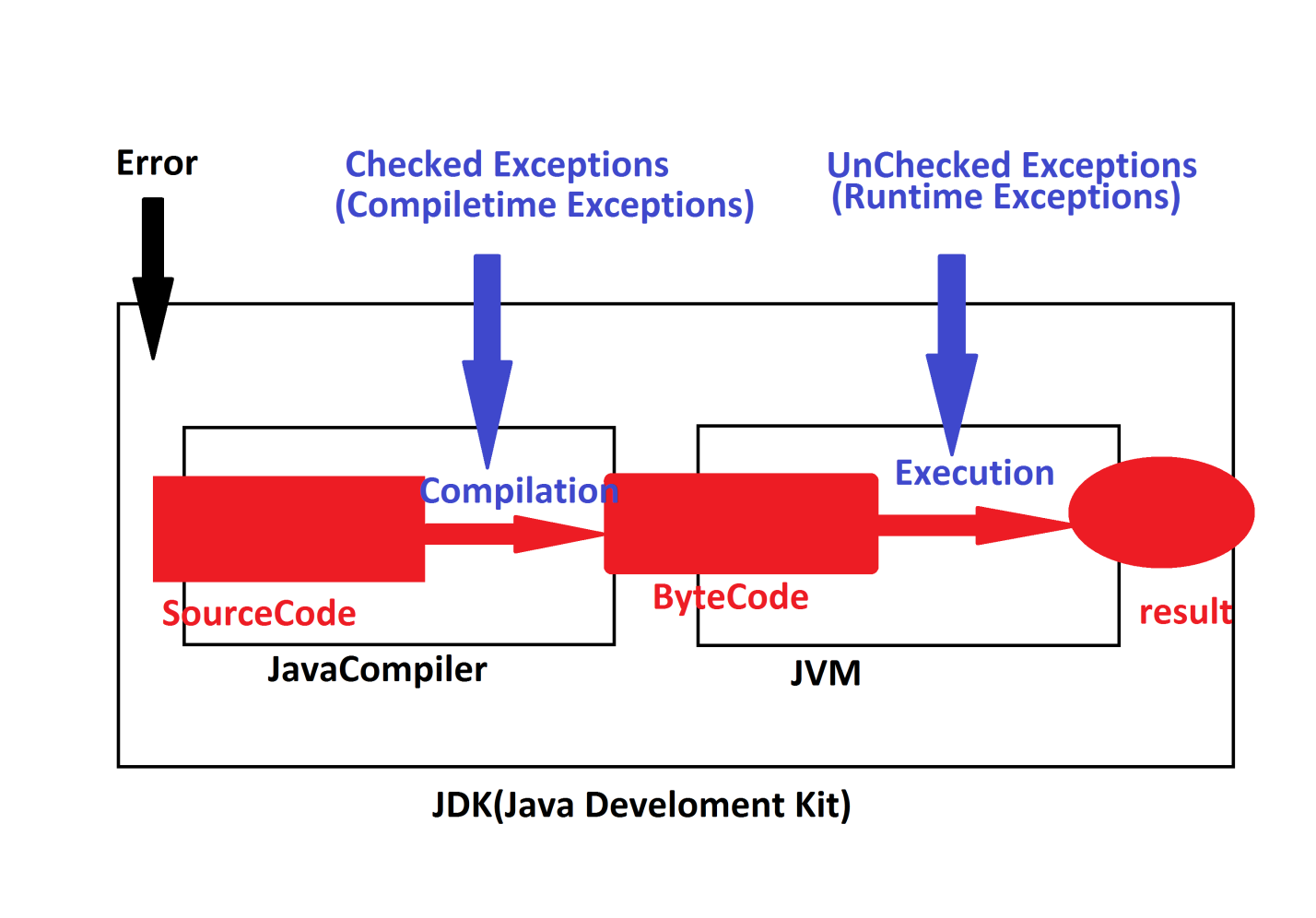
***--------------------------------------------------------***

***Types of Exceptions:***

***=>Exceptions are categorized into two types:***

***1.UnChecked Exceptions***

***2.Checked Exceptions***

******

***1.UnChecked Exceptions:***

***=>The exceptions which are not identified by the compiler at compilation***

***stage will be raised at execution stage are known as UnChecked Exceptions***

***or Runtime Exceptions.***

***=>These UnChecked Exceptions are categorized into two types:***

***(a)Pre-defined UnChecked Exceptions***

***(b)User defined UnChecked Exceptions***

***(a)Pre-defined UnChecked Exceptions:***

***=>The UnChecked Exceptions which are already defined and available from***

***JavaLib are known as Pre-defined UnChecked Exceptions or Built-in UnChecked***

***Exceptions.***

***Ex:***

***java.util.InputMismatchException***

***java.lang.NumberFormatException***

***java.lang.ArithmeticException***

***java.lang.ArrayIndexOutOfBoundException***

***(b)User defined UnChecked Exceptions:***

***=>The UnChecked Exceptions which are defined by the programmer are***

***known as User defined UnChecked Exceptions.***

***==================================================================***

***2.Checked Exceptions:***

***=>The exceptions which are identified by the compiler at compilation***

***stage are known as Checked Exceptions or Compiletime Exceptions.***

***=>These Checked Exceptions are categorized into two types:***

***(a)Pre-defined Checked Exceptions***

***(b)User defined Checked Exceptions***

***(a)Pre-defined Checked Exceptions:***

***=>The Checked Exceptions which are already defined and available from***

***JavaLib are knwon as Pre-defined Checked Exceptions or Built-in Checked***

***Exceptions.***

***Ex:***

***java.lang.ClassNotFoundException***

***java.lang.InterruptedException***

***java.io.IOException***

***faq:***

***define forName() method?***

***=>forName() method is used to load the class at runtime or execution***

***time.***

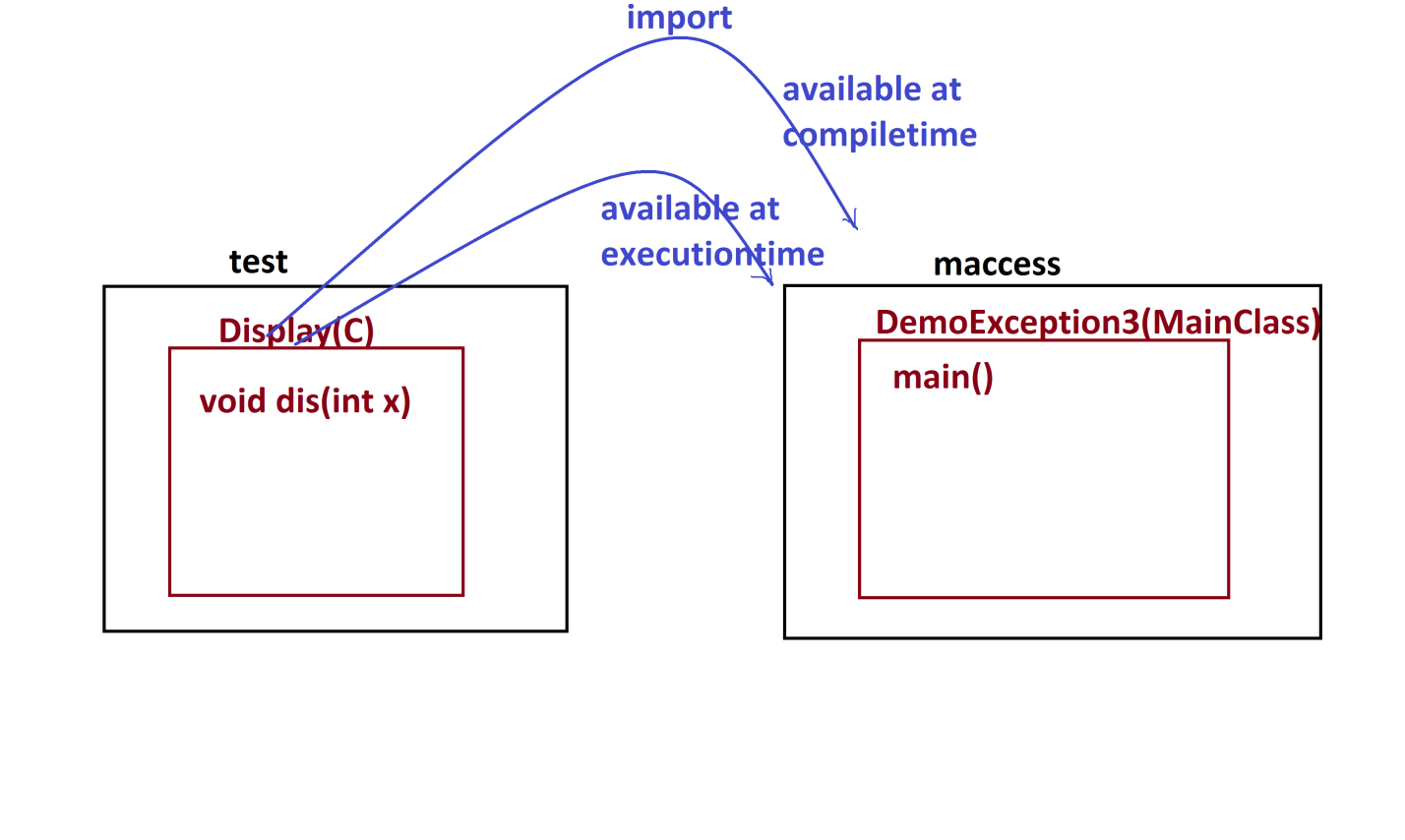
***=>This forName() method is available from "java.lang.Class"***

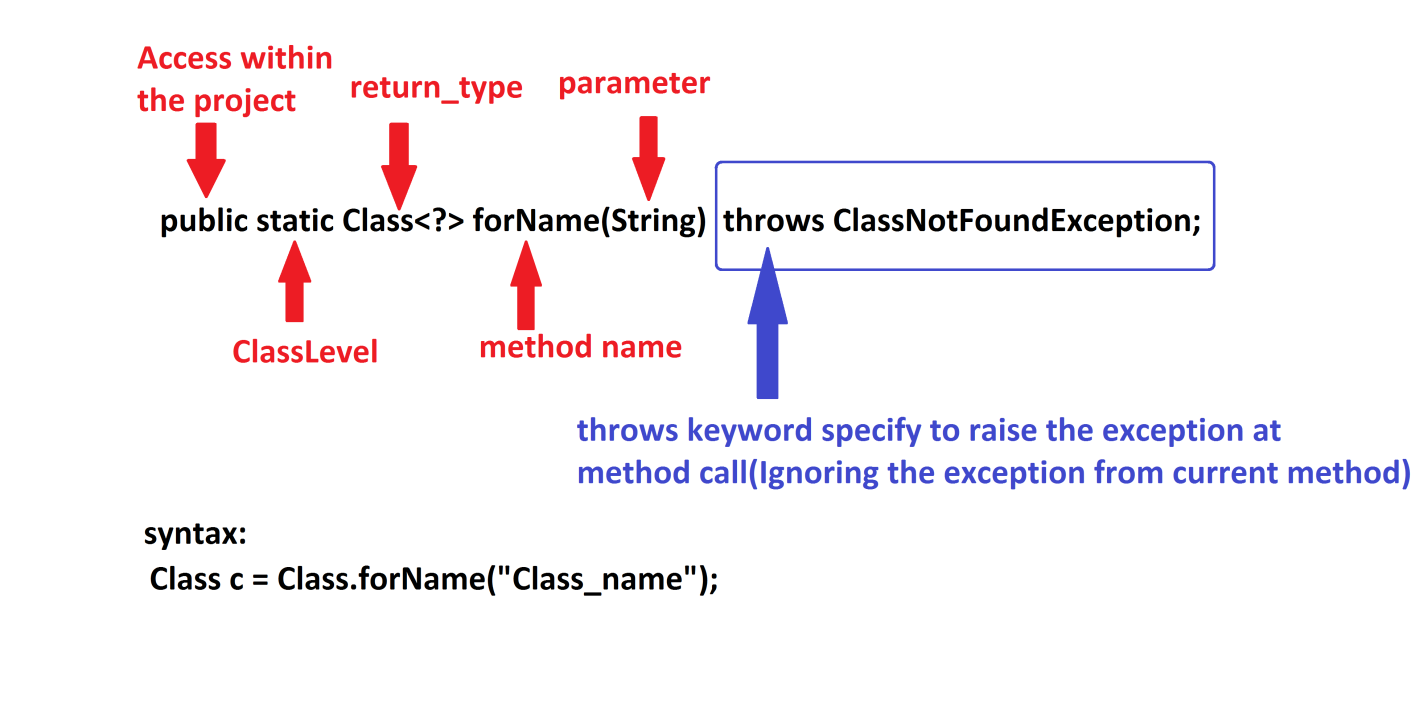
***Method Signature of forName():***

***public static java.lang.Class<?> forName(java.lang.String)***

***throws java.lang.ClassNotFoundException;***

***Diagram:***

******

******

***faq:***

***define newInstance() method?***

***=>newInstance() method is used to create object for the classes loaded at***

***runtime or execution time.***

***=>This newInstance() method is available from "java.lang.Class".***

***Method Signature of newInstance():***

***public T newInstance() throws java.lang.InstantiationException,***

***java.lang.IllegalAccessException;***

***syntax:***

***T ob = c.newInstance();***

***Ex:***

***Display.java***

***package test;***

***public class Display {***

***public void dis(int x) {***

***System.out.println("=====dis(x)====");***

***System.out.println("The value x:"+x);***

***}***

***}***

***DemoException3.java(MainClass)***

***package maccess;***

***public class DemoException3 {***

***@SuppressWarnings({ "rawtypes", "deprecation" })***

***public static void main(String[] args)throws ClassNotFoundException,***

***InstantiationException,IllegalAccessException***

***{***

***Class c = Class.forName("test.Display");//Loading class at runtime***

***test.Display ob = (test.Display)c.~~newInstance~~();***

***ob.dis(12);***

***}***

***}***

***o/p:***

***=====dis(x)====***

***The value x:12***

***===============================================================***