Dt : 2/11/2022

1.Error class:

=>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 concept to handle errors.

2.Exception class:

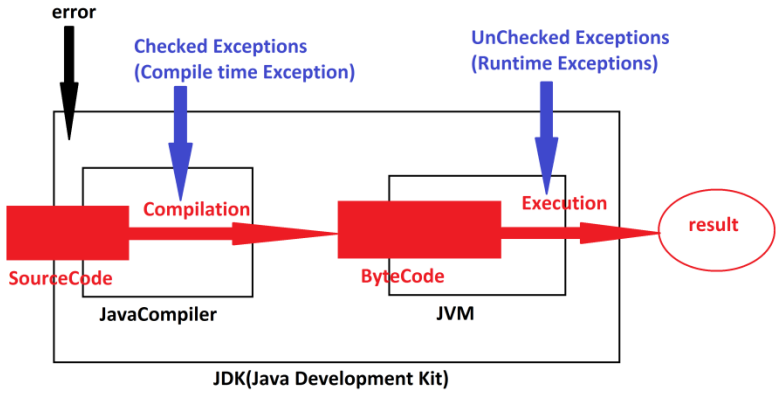
=>The disturbance which is occured from the application is known as exception.

=>"java.lang.Exception" is the PClass or SuperClass of all the exceptions raised from the application.

=>All the exceptions which are raised from the application are categorized into two types:

(a)UnChecked Exceptions

(b)Checked Exceptions



(a)UnChecked Exceptions:

=>The exceptions which are not identified by the compiler will be raised at execution stage are known as UnChecked Exceptions or Runtime Exceptions.

=>These UnChecked Exceptions are categorized into two types:

(i)Pre-defined UnChecked Exceptions

(ii)User defined UnChecked Exceptions

(i)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

(ii)User defined UnChecked Exceptions:

=>The UnChecked exceptions which are defined by the programmer are known as User defined UnChecked Exceptions

Ex:

DemoException1.java

DemoException2.java

==========

(b)Checked Exceptions:

=>The Exceptions which are identified by the compiler at compilation stage are known as Checked Exceptions or Compile time exceptions.

=>These Checked Exceptions are categorized into two types.

(i)Pre-defined checked exceptions

(ii)User defined checked exceptions

(i)Pre-defined checked exceptions:

=>The Checked exceptions which are already defined and available from JavaLib are known as Pre-defined Checked exceptions or Built-in Checked exceptions.

Ex:

java.lang.ClassNotFoundException

java.lang.InterruptedException

java.lang.CloneNotSupportedException

java.io.IOException

faq:

define forName() method?

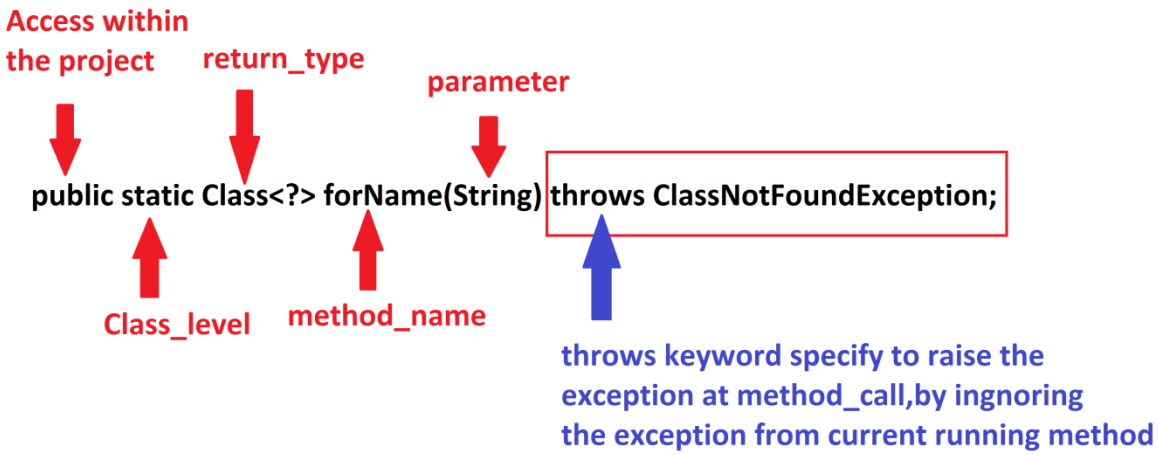
=>forName() method is used to load the class at runtime or execution time, in this process the class not available to Compilation time.

=>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;




```
syntax:
Class c = Class.forName("Class_name");
```

faq:

define newInstance() method?

=>newInstance() method is used to create object for the class 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;

suntax:

T ob = c.newInstance();

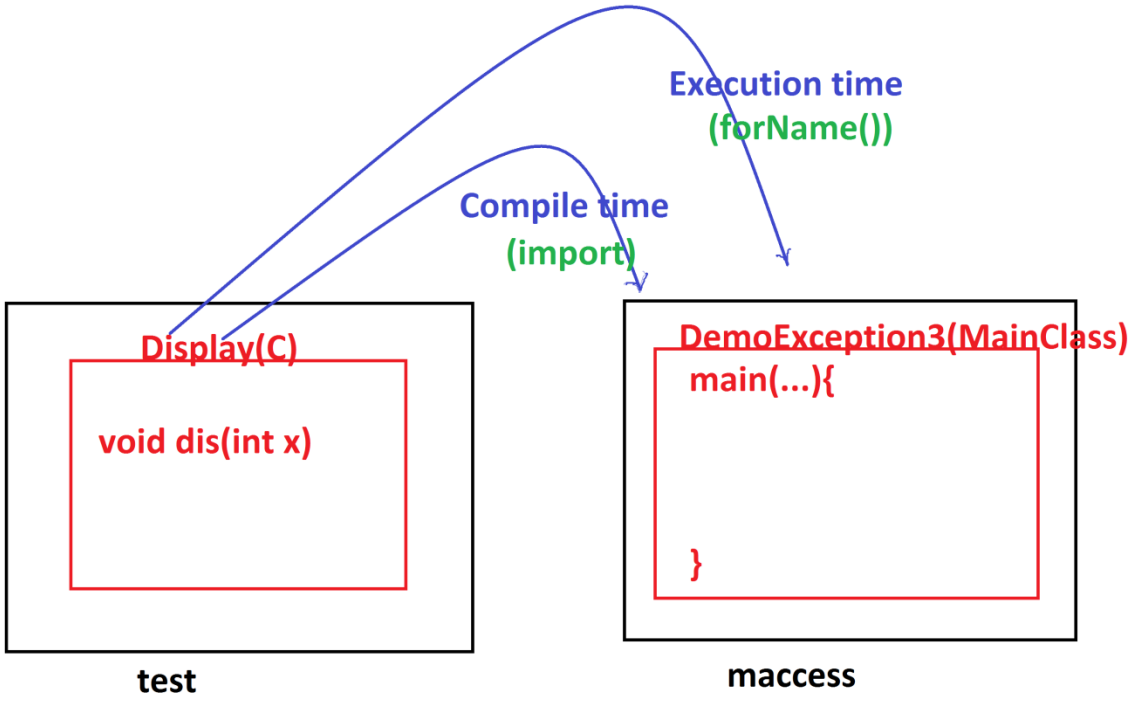
Ex:

Display.java

```
package test;
public class Display extends Object
{
	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();
                    //creating Object for Runtime loaded class
		ob.dis(123);
	}
}
```



==========

faq:

define "throws" keywork?

=>"throws" will specify to raise the exception at method call.

=>when exception raised at method call,compiler will identify the exception

=>when Compiler identifies the exception then it comes under Checked exception or Compiletime exception.

==========

DisAdvantage of "new" keyword:

=>"new" keyword cannot create object for the class which is loaded at runtime or execution time,which means "new" can create object for the

class which is available at compilation stage.

==========