

Checked Exception	Unchecked Exception
Exception checked by compiler for smooth execution of program	Exception not checked by compiler and handled by JVM
Mostly occurred so compiler checked	Rarely occurred so compiler not take care
Ex. file not found, InterruptedException	Ex. Arithmetic, null pointer, arrayIndexOutOfBoundsException

### **Final, finally, finalize**

#### **Program: multiple try catch**

```
package ExceptionProg;

public class Prog4 {

    public static void main(String[] args) {

        try
        {
            System.out.println("Divide = " + 10/0);
        }
        catch(ArithmeticException e)
        {
            System.out.println("Exception is handled + " + e);
        }

        try
        {
            String str = null;
            System.out.println(str.toUpperCase());
        }
        catch(Exception a)
        {
            System.out.println("Exception is handled + " + a);
        }
    }
}
```

#### **Program : multiple catch block**

```
package ExceptionProg;

public class Prog5 {

    public static void main(String[] args) {

        try
        {
            System.out.println(10/0);
        }
        catch (NullPointerException n) {
```

```

System.out.println("Exception handled - " + n);
}
catch(ArrayIndexOutOfBoundsException ai)
{
System.out.println("Exception handled - " + ai);
}
catch (ArithmeticException ae) {
System.out.println("Exception handled - " + ae);
}
catch(Exception e)
{
System.out.println("Exception handled - " + e);
}
finally {

}
}
}

```

## **Finalize**

Method available in object super class

Release the resources allocated by unused object, before removing unused object by garbage collection

Finalize is protected by default/ but we can use public

### **Program:**

```
package ExceptionProg;
```

```
public class Prog6 {
```

```
@Override
```

```
protected void finalize()
```

```
{
```

```
//Data Base connection and
```

```
//network connection
```

```
}
```

```
public static void main(String[] args) {
```

```
Prog6 a = new Prog6();
```

```
a.finalize();
```

```
}
```

```
}
```

## **What is garbage collection**

Delete the used entity

Data Base connection and network connection

Then finalize method is to be called by garbage collection before destroying the object for object to perform cleanup activity

<b>Finally</b>	<b>Finalize()</b>
It is block used in try catch	It is method
Used to close the resources open in try block	Used to take the resource back from unused object, garbage collector deallocate the resource before destroying the object

### Selenium configuration setup

1. Search selenium hq on google
  2. Select the link <https://www.selenium.dev>
  3. Go to downloads
  4. Look for previous releases and click on link here
  5. Look for Selenium 4.0.0
  6. Look for [selenium-java-4.0.0.zip](#)
  7. Download the zip
  8. Extract the zip file
- 
1. Open the installed chrome
  2. Click on 3 dots
  3. Select help option
  4. Select about chrome option
  5. Check for the version we are using
- 
1. Search chromedriver download in google
  2. Select the link <https://chromedriver.chromium.org/downloads>
  3. Look the current release
  4. Select and click the version which is available on machine
  5. Select and download proper link of chromedriver for OS we are using
  6. Extract the zip file
- 
1. Open the eclipse
  2. Right click on the project we are working
  3. Select the last option properties
  4. Select java build path
  5. Select libraries tab
  6. Click on add external JARs
  7. Select all the extracted jar of selenium 4.0.0 (from selenium folder and libs folder too)
  8. Click on Apply and Ok
- 
1. Copy the downloaded chrome driver and paste in project by right clicking on current project
  2. Look for chrome driver in project and right click
  3. Copy the location and paste in system.setProperties as second argument
- 
1. Create a new package

2. Create new class with main function
3. Use the below code to verify the selenium configuration

```
public class webdriverMethods {  
    public static void main(String [] args)    {  
        System.setProperty("webdriver.chrome.driver","Address of chrome driver");  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://www.facebook.com/");  
    }  
}
```

### **Program :**

```
package SimpleSeleniumProg;
```

```
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class Demo {  
  
    public static void main(String[] args) {  
        System.setProperty("webdriver.chrome.driver", "chromedriver");  
        WebDriver driver = new ChromeDriver();  
        driver.get("https://www.facebook.com/");  
    }  
}
```

### **Program : Return method**

```
package SimpleSeleniumProg;
```

```
public class ReturnMethod {  
  
    public void add()  
    {  
        System.out.println("Sum = " +(10+20));  
    }  
  
    public int multiply()  
    {  
        return 10*100; //1000  
    }  
  
    public String nameofEmp()  
    {  
        return "Harry";  
    }  
  
    public static void main(String[] args) {  
  
        ReturnMethod a = new ReturnMethod();  
        a.add();  
    }  
}
```

```
        System.out.println("Result of multiplication = " + a.multiply());

        int result = a.multiply(); //1000
        System.out.println("Multiplication = " + result);

        String res = a.nameofEmp();
        System.out.println("Name of Employee = " + res);
    }
}
```