

EXCEPTION HANDLING:-

An exception (or exceptional event) is a problem that arises during the execution program will terminate itself.

Types of Exception:-

- * Checked Exceptions (Compile time Exception)
- * Unchecked Exception

Unchecked Exception (Run time Exception) :-

- * Arithmetic Exception
- * Null pointer Exception
- * Input mismatch Exception
- * Array Index Out of Bound Exception
- * StringIndexOutOfBoundsException
- * IndexOutOfBoundsException

Example Program:-

Arithmetic Program:-

```
package org.exception;
public class Exceptioneg {
    public static void main(String [] args) {
        int a = 10;
        int b = 0;
        int c = a/b;
        System.out.println(c);
    }
}
```

Null pointer Exception :-

```
package org. exception;  
public class Exceptionex {  
    public static void main (String [] args) {  
        String s = null;  
        System.out.println(s.length());  
    }  
}
```

Input Mismatch Exception :-

```
package org. exception;  
public class Exceptionex {  
    public static void main (String [] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println ("Enter your input :");  
        int nextInt = sc.nextInt();  
        System.out.println ("Entered input is : " + nextInt)  
    }  
}
```

ArrayIndexOut of bound exception :-

```
package org. exception;  
public class Exceptionex {  
    public static void main (String [] args) {  
        int a [] = new int [3];  
        a[0] = 10;  
        a[1] = 20;  
        a[2] = 30;  
        System.out.println (a[5]);  
    }  
}
```

StringIndexOut of bound exception :-

```
package org. exception;
public class ExceptionEx {
    public static void main (String [] args) {
        String s = "java";
        System.out.println (s.charAt(6));
    }
}
```

IndexOut of bound Exception :-

```
package org. exception;
import java.util.ArrayList;
import java.util.List;
public class ExceptionEx {
    public static void main (String [] args) {
        List <Object> li = new ArrayList();
        li.add (10);
        li.add (20);
        li.add (30);
        System.out.println (li.get (5));
    }
}
```

Checked Exception :-

* File not found Exception

* IO not found Exception

Exception Handling :-

Try
Catch
Finally
Throw

Try and Catch:-

```
package org.exception;
public class Exceptionex {
    public static void main (String [] args) {
        try {
            String a = null;
            System.out.println(a.length());
        } catch (Exception e) {
            // System.out.println(e.getMessage());
            System.out.println("String declared as Null");
        }
    }
}
```

Multiple Catch:-

```
package org.exception;
public class Exceptionex {
    public static void main (String [] args) {
        try {
            String a = null;
            Sysout (a.length());
        } catch (ArithmaticException e) {
            Sysout (e.getMessage());
        } catch (NullPointerException n) {
            Sysout (n.getMessage());
        }
    }
}
```

Try, Catch and Finally:-

```
package org.exception;
public class Exceptionex {
    public static void main (String [] args) {
        try {
            String s = null;
```

```
Sysout (e.getMessage());  
} catch (ArithmeticException e) {  
    Sysout (e.getMessage());  
} finally {  
    Sysout ("Thanks You");  
}  
}
```

Possible Combinations of Try, Catch and Finally :-

- * Try and Catch → Possible
- * Try, Catch, Finally → Possible
- * Try, Finally → Possible
- * Try → Not possible
- * Finally - Not possible
- * Catch & Finally Not possible

Throw or Customised User defined exception :-

```
package org.exception;  
public class ExampleEx {  
    public static void main (String [] args) {  
        try {  
            int age = 12;  
            if (a >= 18) {  
                Sysout ("Allow to vote");  
            } else {  
                throw new Exception ("Age Not Met");  
            }  
        } catch (Exception e) {  
            Sysout (e.getMessage());  
        }  
    }  
}
```

Throws Example :-

```
Package org. collection;  
public class Exception ex {  
    public static void main (String args) {  
        int acc-bal = 2000;  
        int wit = 5000;  
        int rem = acc-bal - wit;  
        try {  
            if (acc-bal < wit)  
                throw new InsufficientFundException();  
            System.out.println (rem);  
        } catch (InsufficientFundException e) {  
            System.out.println ("Not enough Money");  
        }  
    }  
}
```

Throws Example:-

```
Package org. exception;  
public class other {  
    public static void divide (int a, int b) throws Arithmetic  
        int c = a/b;  
        Sysout (c);  
    }  
}
```

```
Package org. exception;  
public class Exception throws {  
    public static void main (String [] args) {  
        try {  
            other.divide (10, 0);  
        } catch (ArithmeticException e) {  
            Sysout ("Check the input");  
        }  
    }  
}
```

File Handling:-

In Google search type Maven Repository.

then Select <http://mvnrepository.com>

then In Search type Commons IO then click Search

then Select 1. Apache Commons IO

then click 2.8.0 (Second Most Usage)

then click jar (It'll get download)

Meanwhile Right click on Project (where you have your code)

New → Folder → Give Folder Name - Finish

Now copy that download file and paste into created folder.

Right click on the CommonsIO 2.8.0 in external folder

then select Buildpath click add buildpath.

Ex:-

```
package org.exception;
import java.io.File;
import java.io.IOException;
import org.apache.commons.io.FileUtils;
public class FileIO {
    public static void main (String args) throws IOException {
        File file = new File ("E://FileOperation");
        System.out.println (file.mkdir());
```

```
File file1 = new File("E://FileOperation//java.txt");
boolean CreateNewFile = file1.createNewFile();
System.out.println(CreateNewFile);
FileUtils f = new FileUtils();
f.write(file1, "October Java Batch SIX");
System.out.println(f.readFileToString(file1));
File file2 = new File("E://FileOperation//Selenium.txt");
boolean CreateNewFile2 = file2.createNewFile();
System.out.println(CreateNewFile2);
f.copyFile(file1, file2);
}
```