Errors and Exception in Java :-

⇒ Three types of Error in Java :-

1) Syntax error
2) Logical error
3) Runtime error.

Exceptions in Java :-

An Exception is an event that occurs when a program is executed dissented the normal flow of instruction.

Mainly two types of exceptions in Java :-

i) Checked Exceptions :- Compile time exception

ii) Unchecked Exceptions :- Runtime exception.

Commonly Occuring Exceptions :-

1. NullPointer Exception
2. Arithmetic Exception
3. ArrayIndexOutOfBound Exception
4. IllegalArgument Exception
5. NumberFormat Exception

try-catch block in Java :-

Syntax :-

```
try {
    // code try to run
}
catch (Exception e){
    // catch the error
}
```

eg-

```
int a=30, b=0;
try {
    sout("result "+ a/b);
}
catch(Exception e){
    sout(e);
}
```

Handling specific Exception –

eg-
```
catch (ArithmeticException e){
    sout(e);
}
catch (ArrayIndexOutOfBoundsException e){
    sout(e);
}
```

Nested try catch in Java :-

eg-
```
try{ sout("Hello");
    try{ sout("Array-" + arr[3]);
    }
    catch (ArrayIndex... e){
        sout("Error" + e);
    }
}
catch (Exception e){
    sout("Outer Error" + e);
}
```

Exception Class in Java :-

⇒ we can write our own custom exception using the Exception class in Java -

eg - class MyException extends Exception {
    // Overridden method
}

⇒ The Exception class has the following important methods -

① Strings toString() executed when sout(e) is ran.

② void printStackTrace() - print Stack trace.

③ String getMessage() - Prints the exception message.

eg- class MyException extends Exception {

```
    @Override
        public string toString(){
            return "return from to String()";
        }
        @Override
        public String getMessage(){
            return "I am getMessage()";
        }
    }
```

*In main method- int a=30, b=0;

```
        try {
            if(b==0) throw new MyException();
            Sout("Result is "+ a/b);
        }
        catch(Exception e){
            Sout(e);
            Sout(e.toString());
            Sout(e.getMessage());
            e.printStackTrace();
        }
```

✗ It's always a good practice to include all the exceptions in try and catch block.

Throw vs Throws Keywords in Java -

 # throw - used to throw an exception explicitly by the programmer.

eg- throw new ArithmeticException ("Div by 0");

 # throws - This gives information to the programmer that there might be an exception So it is better to be prepared with try catch block

eg -

```
pubblic static int divide (int a, int b) throws
        Arithmetic exception {
    int result = a/b;
    return result;
}
```

main -

```
try {
    int c = divide (6,0);
    sout (c);
}
catch (Exception e){
    sout(e);
}
```

Finally Block in Java —

⇒ Finally block contains the code which is always executed whether the exception is handled or not.

eg -

```
try {
    return 1;
}
catch (Exception e) {
    System.out.println(e);
}
finally {
    sout("Cleaning up resources ...");
}
```