

## EXCEPTION HANDLING

TRY BLOCKS

CATCHING AN  
EXCEPTION

ADVANTAGES

THROWING AN  
EXCEPTION

WHAT IS  
EXCEPTION  
HANDLING

FINALLY  
STATEMENT

# EXCEPTION HANDLING

- Exception handling is one of the most important feature of java programming that allows us to handle the runtime errors caused by exceptions.

What is an exception?

An Exception is an unwanted event that interrupts the normal flow of the program. When an exception occurs program execution gets terminated. In such cases we get a system generated error message. The good thing about exceptions is that they can be handled in Java.

Why an exception occurs?

There can be several reasons that can cause a program to throw exception. For example: Opening a non existing file in your program, Network connection problem



## Advantages of exception handling

- Exception handling ensures that the flow of the program doesn't break when an exception occurs.
- By handling we make sure that all the statements execute and the flow of program doesn't break.
- One of the important purposes of exception handling in Java is to continue program execution after an exception is caught and handled.



# Try Block

- The try block contains set of statements where an exception can occur.
- A try block is always followed by a catch block, which handles the exception that occurs in associated try block.
- A try block must be followed by catch blocks or finally block or both

## Syntax of try block:

```
try{  
    //statements that may cause an exception  
}
```

The try statement allows you to define a block of code to be tested for errors while it is being executed.



# Catching Exception Handling

- Java catch block is used to handle the Exception by declaring the type of exception within the parameter.
- The catch keyword catches exceptions generated by try statements. The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

**General form:**

```
catch (Exception Type1 e1) {  
    // exception handler for Exception Type1  
}  
catch (Exception Type2 e2) {  
    // exception handler for Exception Type2  
}
```



# Throwing Exception Handling

- Throw is a keyword which is used to throw an exception explicitly in the program inside a function or inside a block of code.
- Throws keyword is used in the method signature to declare an exception which might be thrown by the function while the execution of the code.
- Exceptions indicate a broken contract
  - Precondition (e.g. file is open for read)
  - Postcondition (e.g. read a character from file)



# Finally Statement

- Java supports another statement known as finally statement that can be used to handle an exception that is not caught by any of the previous catch statements.
- We can put finally block after the try block or after the last catch block.
- The finally block is executed in all circumstances. Even if a try block completes without problems, the finally block executes.

General form:

```
finally {  
    // block of code to be executed before try block  
    ends  
}
```

