



# Exception Handling

# Agenda

1

**Introduction to Exception Handling**

2

**Exception Handling Keywords**

# Introduction to Exception Handling



# What is an Exception?

- An exception is an event that occurs **during the execution of a program** that disrupts the normal flow of instructions
- The **ability** of a program to intercept run-time errors, take corrective measures and continue execution is referred to as exception handling
- There are various situations when an exception could occur:
  - Attempting to access a file that does not exist
  - Inserting an element into an array at a position that is not in its bounds
  - Performing some mathematical operation that is not permitted
  - Declaring an array using negative values

## What is an Exception? (Contd.).

- The following exceptions have to be addressed by the programmer.
- They are frequently encountered in java programs.
  - NPE                    -- NullPointerException
  - NFE                    -- NumberFormatException
  - AIOOBE                -- ArrayIndexOutOfBoundsException
  - SIOOBE                -- StringIndexOutOfBoundsException
  - AE                     -- ArithmeticException
- Can You list under which situation, these exceptions occur?
- Can You say, **How to avoid them** ?
- Can You list some exceptions ?

# Uncaught Exceptions

```
class Demo {  
    public static void main(String args[]) {  
        int x = 0;  
        int y = 50/x;  
        System.out.println("y = " + y);  
    }  
}
```

Will compile, but when you execute it, displays:

**java.lang.ArithmeticException:** / by zero

at Demo.main(Demo.java:4)

At What Line Exception occurred?  
Can you see?

# Exception Handling Techniques

- There are several built-in exception classes that are used to handle the very fundamental errors that may occur in your programs
- You can create your own exceptions also by extending the **Exception** class
- These are called **User-defined Exceptions**.
- **Can you say some example situations, where you will create User-defined Exceptions?**

# Handling Runtime Exceptions

- Whenever an exception occurs in a program, an object representing that exception is created and thrown in the method in which the exception occurred
- Either you can **handle** the exception, or ignore it
- In the latter case, the exception is handled by the Java run-time-system and the program terminates
- Handling the exceptions will avoid **abnormal program termination!**



# Exception Handling Keywords



# Exception Handling Keywords

Java's exception handling is managed using the following keywords: **try**, **catch**, **throw**, **throws** and **finally**.

```
try {  
    // code comes here  
}  
  
catch(TypeofException obj) {  
    //handle the exception  
}  
  
finally {  
    //code to be executed before the program ends  
}
```

# Summary

In this session, you were able to :

- Learn brief introduction on exception and techniques to handle exception
- Learn about exception handling keywords



# Thank You