Exception Handling

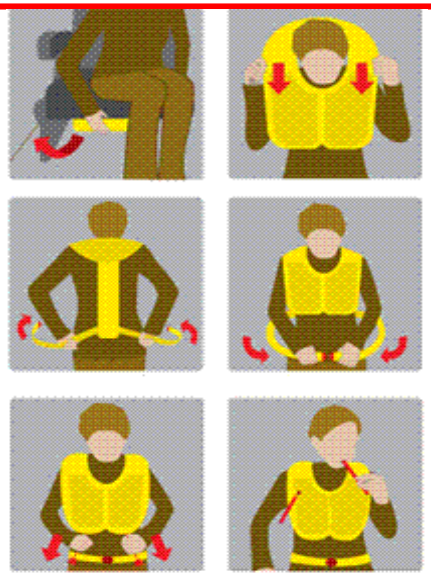
Agenda

1. Introduction to Exception Handling
2. Exception Handling Techniques
3. Exception Handling Keywords

Introduction to Exception Handling

Scenario

Meera is flying to NewYork



What are these?

Scenario (Contd.).

The previous slide depicts an air hostess giving a demonstration of steps that we have to take as passengers, in case of emergency.

Why this demonstration is important?

Why the air line staff insists on fastening our seat belts?

Scenario (Contd.).

You have to be aware of how to tackle a situation in case of an emergency while you are flying aboard an aircraft.

You have to fasten your seat belts to protect yourself from mishaps that can occur during take off and landing.

This example demonstrates how you have to think in advance the many possibilities of mishaps that can occur and what are the preventive measures that can be taken.

Exception Handling

Similarly, when we write programs as part of an application, we may have to visualize the challenges that can disrupt the normal flow of execution of the code.

Once we know what are the different situations that can disrupt the flow of execution, we can take preventive measures to overcome these disruptions.

In java, this mechanism comes in the form of Exception Handling.

What is an Exception?

- In procedural programming, it is the responsibility of the programmer to ensure that the programs are error-free in all aspects
- Errors have to be checked and handled manually by using some error codes
- But this kind of programming was very cumbersome and led to **spaghetti code**
- Java provides an excellent mechanism for handling runtime errors

What is an Exception? (Contd.).

- 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

What is an Exception? (Contd.).

- 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

Uncaught Exceptions

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

Although this program will compile, but when you execute it, the Java run-time-system will generate an exception and displays the following output on the console :

java.lang.ArithmeticException: / by zero

at Demo.main(Demo.java:4)

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, and will be used in situations that are unique to your applications

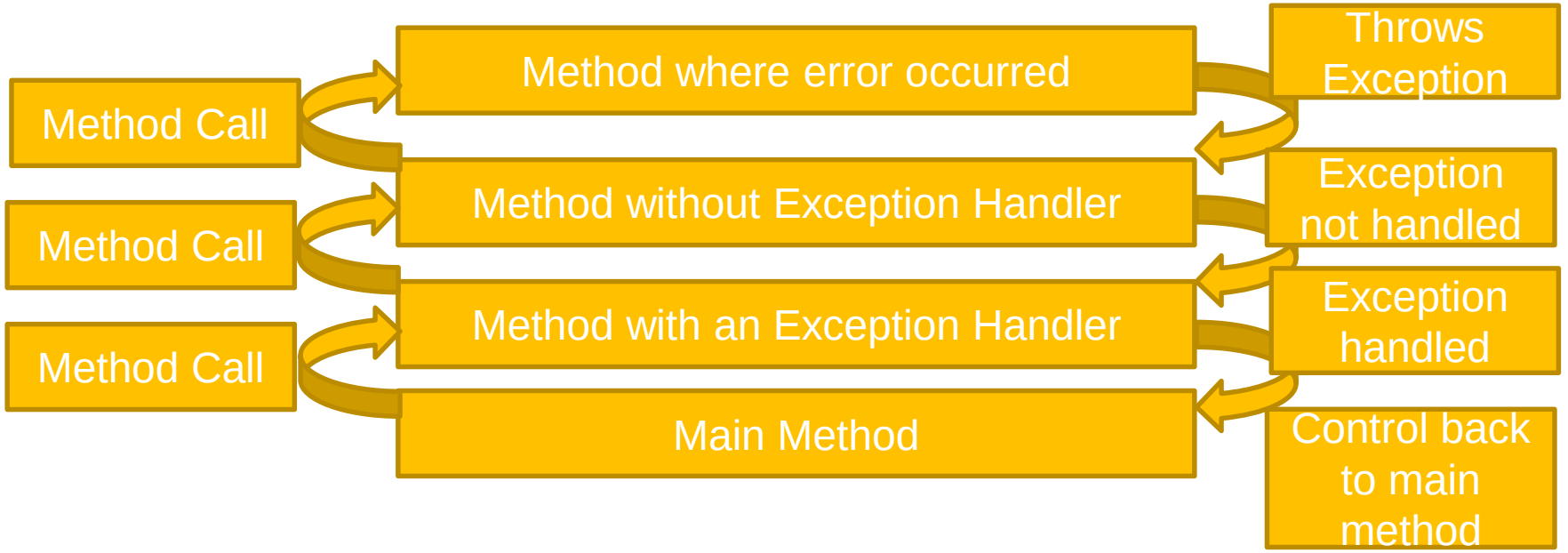
Handling Runtime Exceptions

- 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
- However, handling the exceptions will allow you to fix it, and prevent the program from terminating abnormally

Advantages - Exceptions

1. **Separating Error-Handling Code from "Regular" Code**

2. **Propagating Errors Up the Call Stack**



3. **Grouping and Differentiating Error Types**

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