

# Introduction to Java

## Objectives

- In this session, you will learn to:
  - Handle exceptions
  - Use the assert keyword

# Introduction to Java

## Handling Exceptions

### ■ Scenario:

Consider the scenario of the Classic Jumble Word game.

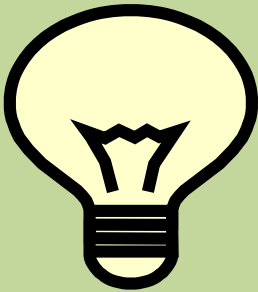


If the player enters an incorrect menu option, such as a string value instead of an integer value, the game terminates abnormally.

# Introduction to Java

## Handling Exceptions (Contd.)

- Scenario (Contd.):



In order to handle and prevent the abnormal termination, you need to implement exception handling.

# Introduction to Java

## Exploring Exceptions

- When a run-time error occurs, an exception is thrown by the JVM.
- The Java run-time system proceeds with the normal execution of the program after an exception is handled.
- If no appropriate exception handler is found by the JVM, the program is terminated.
- There are several built-in exceptions that have been identified in Java.
- To deal with the exceptions, Java has various built-in exception classes, which are organized in a hierarchical manner.
- The `Throwable` class is the base class of exceptions in Java.
- You can throw only those exception objects that are derived from the `Throwable` class.

# Introduction to Java

## Exploring Exceptions (Contd.)

- The following two classes are derived for the `Throwable` class:



- The `Exception` class represents the conditions that a program should handle.
- The various subclasses of `Exception` class are:
  - `ClassNotFoundException`
  - `IllegalAccessException`
  - `RuntimeException`

# Introduction to Java

## Exploring Exceptions (Contd.)

- The `ClassNotFoundException` exception is thrown when a class is being referred, but no definition for the same is found.
- The `IllegalAccessException` exception is thrown when a particular method is not found.
- The `Error` class defines the exceptions related to the Java run-time environment.

# Introduction to Java

## Identifying Checked and Unchecked Exceptions

- Java exceptions are categorized into the following types:
  - Checked exceptions
  - Unchecked exceptions
- Checked Exceptions:
  - Are the invalid conditions that occur in a Java program due to certain problems.
  - Are the objects of the `Exception` class or any of its subclasses, excluding the `RuntimeException` and `Error` classes.
- Some of the most commonly used checked exceptions are:
  - `ClassNotFoundException`
  - `IllegalAccessException`
  - `InstantiationException`
  - `NoSuchMethodException`

# Introduction to Java

## Identifying Checked and Unchecked Exceptions (Contd.)

- The unchecked exceptions occur because of programming errors.
- The compiler does not force a programmer to handle these exceptions.
- The various unchecked exceptions are:
  - `ArithmeticException`
  - `ArrayIndexOutOfBoundsException`
  - `ArrayStoreException`
  - `ClassCastException`
  - `IllegalArgumentException`
  - `NegativeArraySizeException`
  - `NullPointerException`
  - `NumberFormatException`