



**CHANDIGARH
UNIVERSITY**

Discover. Learn. Empower.

**INSTITUTE : UIE
DEPARTMENT : CSE**

Bachelor of Engineering (Computer Science & Engineering)

PROJECT BASED LEARNING IN JAVA

(20CST-319/20ITT-319)

TOPIC OF PRESENTATION:

Types of Exceptions, Exception handling in Java.

DISCOVER . LEARN . EMPOWER

Lecture Objectives

In this lecture, we will discuss:
Types of Exceptions, Exception
handling in Java.



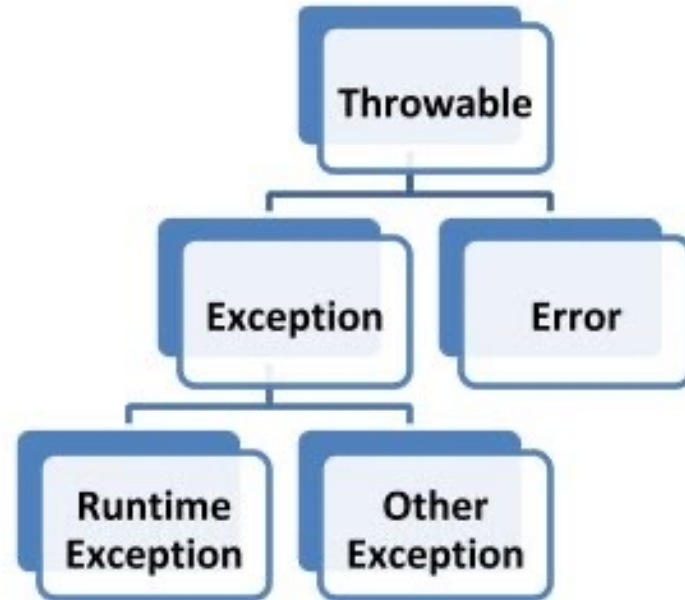
Support for various mechanisms such as exception handling → enables developers to manage the runtime errors caused by the exceptions.

Exceptions are the unwanted errors or bugs or events that restrict the normal execution of a program. Each time an exception occurs, program execution gets disrupted.

There are several reasons behind the occurrence of exceptions. These are some conditions where an exception occurs:

- Whenever a user provides invalid data.
- The file requested to be accessed does not exist in the system.
- When the **Java Virtual Machine (JVM)** runs out of memory.
- Network drops in the middle of communication.

The parent class of all the exception classes is the **java.lang.Exception** class.



If we talk about the **Exception** class, it is a subclass of the built-in **Throwable** class. There is another subclass which is derived from the Throwable class i.e. **Error** .

The error can be defined as an abnormal condition that indicates something has gone wrong with the execution of the program. These are not handled by Java programs.

Checked And Unchecked Exceptions in Java

Checked exceptions for all exceptional events that you can anticipate and that a well-written application should be able to handle. A checked exception extends the `Exception` class. A method that throws a checked exception or that calls a method that specifies a checked exception need to either specify or handle it.

Unchecked exceptions extend the `RuntimeException`. You should use them for internal errors that you can't anticipate and that, most often, the application can't recover from. Methods can but don't need to handle or specify an unchecked exception.

Typical examples that throw unchecked exceptions are:

the missing initialization of a variable which results in a *`NullPointerException`* or
the improper use of an API that causes an *`IllegalArgumentException`*

Checked Exceptions

1. `ClassNotFoundException`: Class not found.
2. `CloneNotSupportedException`: Attempt to clone an object that does not implement the `Cloneable` interface.
3. `IllegalAccessException`: Access to a class is denied. `IllegalAccessException` signals that a particular method could not be found.
4. `InstantiationException`: Attempt to create an object of an abstract class or interface.
5. `InterruptedException`: One thread has been interrupted by another thread.
6. `NoSuchFieldException`: A requested field does not exist.
7. `NoSuchMethodException`: A requested method does not exist.
8. `ReflectiveOperationException` - Superclass of reflection -related exceptions(Added by JDK 7.)

Runtime - Unchecked Exceptions

1. `ArithmeticException`: Arithmetic error, such as divide-by-zero.
2. `ArrayIndexOutOfBoundsException`: Array index is out-of-bounds.
3. `ArrayStoreException`: Assignment to an array element of an incompatible type.
4. `ClassCastException`: Invalid cast.
5. `EnumConstantNotPresentException`: An attempt is made to use an undefined enumeration value
6. `IllegalArgumentException`: Illegal argument used to invoke a method.
7. `IllegalMonitorStateException`: Illegal monitor operation, such as waiting on an unlocked thread.
8. `IllegalStateException`: Environment or application is in incorrect state.
9. `IllegalThreadStateException`: Requested operation not compatible with the current thread state.
10. `IndexOutOfBoundsException`: Some type of index is out-of-bounds.
11. `NegativeArraySizeException`: Array created with a negative size.
12. `NullPointerException`: Invalid use of a null reference.
13. `NumberFormatException`: Invalid conversion of a string to a numeric format.
14. `SecurityException`: Attempt to violate security.
15. `StringIndexOutOfBoundsException`: Attempt to index outside the bounds of a string.
16. `TypeNotPresentException`: Type not found. (Added by J2SE 5.)
17. `UnsupportedOperationException`: An unsupported operation was encountered.



Java Custom Exception

If you are creating your own Exception that is known as custom exception or user-defined exception. Java custom exceptions are used to customize the exception according to user need. By the help of custom exception, you can have your own exception and message.

Let's see a simple example of java custom exception.

```
class InvalidAgeException extends Exception{  
    InvalidAgeException(String s){  
        super(s);  
    }  
}
```



Java Custom Exception

```
class TestCustomException1{
    static void validate(int age)throws InvalidAgeException{
        if(age<18)
            throw new InvalidAgeException("not valid");
        else
            System.out.println("welcome to vote");
    }
    public static void main(String args[]){
        try{
            validate(13);
        }catch(Exception m){System.out.println("Exception occured: "+m);}

        System.out.println("rest of the code...");
    }
}
```

Output: Exception occured: InvalidAgeException:not valid
rest of the code...

QUIZ:

1. Checked exception caught at
 - a. compile time
 - b. run time
 - c. Both at compile and run time
 - d. None
2. Unchecked exception caught at
 - a. compile time
 - b. run time
 - c. Both at compile and run time
 - d. None



Summary:

In this session, you were able to :

- Learn about Types of Exceptions, Exception handling in Java.



References:

Books:

1. Balaguruswamy, *Java*.
2. A Primer, E.Balaguruswamy, *Programming with Java*, Tata McGraw Hill Companies
3. John P. Flynt Thomson, *Java Programming*.

ONLINE NOTES LINKS:

https://www.tutorialspoint.com/java/java_exceptions.htm
<https://www.javatpoint.com/exception-handling-in-java>
<https://www.geeksforgeeks.org/exceptions-in-java/>
<https://www.w3resource.com/java-tutorial/types-of-exception.php>
https://www.protechtraining.com/content/java_fundamentals_tutorial-exceptions

VIDEO LINKS:

<https://www.javatpoint.com/exception-handling-in-java>
<https://www.edureka.co/blog/java-exception-handling>
<https://www.youtube.com/watch?v=4my7mKFaNQs>
<https://www.youtube.com/watch?v=W-N2ltgU-X4>
<https://marcus-biel.com/advanced-exception-handling-in-java/>





THANK YOU

