



SNS COLLEGE OF ENGINEERING

Coimbatore-107
An Autonomous Institution

COURSE NAME : 23CSB201 & Object Oriented Programming

II YEAR/ III SEMESTER

UNIT – III EXCEPTION HANDLING AND MULTITHREADING

Topic: Java's Built-in Exceptions – User defined Exception

Dr.P.Poonkodi

Assistant Professor(SG)

Department of Computer Science and Technology



Introduction

- **exception** is an event that occurs during the execution of a program and disrupts the normal flow of the program execution
- Bugs or errors that we don't want and restrict our program's normal execution of code are referred to as **exceptions**
- we will focus on the **types of exceptions in Java**
- Exceptions can be categorized into two ways:
 - Built-in Exception
 - Checked Exception
 - Unchecked Exception
 - User Defined Exception



Unchecked Exception - Example

```
class UncheckedException1 {  
    public static void main(String args[])  
    {  
        int num[] = {10,20,30,40,50,60};  
        System.out.println(num[7]);  
    }  
}
```

```
C:\Windows\System32\cmd.exe  
  
C:\Users\ajeet\OneDrive\Desktop\programs>javac UncheckedException1.java  
  
C:\Users\ajeet\OneDrive\Desktop\programs>java UncheckedException1  
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index  
7 out of bounds for length 6  
        at UncheckedException1.main(UncheckedException1.java:5)  
  
C:\Users\ajeet\OneDrive\Desktop\programs>_
```



User Defined Exception

- we can write our own exception class by extends the Exception class
- We can throw our own exception on a particular condition using the throw keyword
- For creating a user-defined exception, we should have basic knowledge of the **try-catch** block and **throw** keyword



Example

```
import java.util.*;  
class UserDefinedException{  
    public static void main(String args[]){  
        try{  
            throw new NewException(5);  
        }  
        catch(NewException ex){  
            System.out.println(ex) ;  
        }  
    }  
}
```



Example

```
class NewException extends Exception
{
    int x;
    NewException(int y)
    {
        x=y;
    }
    public String toString()
    {
        return ("Exception value = "+x) ;
    }
}
```

```
C:\Windows\System32\cmd.exe

C:\Users\ajeet\OneDrive\Desktop\programs>javac CustomException.java
C:\Users\ajeet\OneDrive\Desktop\programs>java UserDefinedException
Exception value = 5

C:\Users\ajeet\OneDrive\Desktop\programs>
```



Example

S.No	Checked Exception	Unchecked Exception
1.	These exceptions are checked at compile time. These exceptions are handled at compile time too.	These exceptions are just opposite to the checked exceptions. These exceptions are not checked and handled at compile time.
2.	These exceptions are direct subclasses of exception but not extended from RuntimeException class.	They are the direct subclasses of the RuntimeException class.
3.	The code gives a compilation error in the case when a method throws a checked exception. The compiler is not able to handle the exception on its own.	The code compiles without any error because the exceptions escape the notice of the compiler. These exceptions are the results of user-created errors in programming logic.
4.	These exceptions mostly occur when the probability of failure is too high.	These exceptions occur mostly due to programming mistakes.



Example

S.No	Checked Exception	Unchecked Exception
5.	Common checked exceptions include IOException, DataAccessException, InterruptedException, etc.	Common unchecked exceptions include ArithmeticException, InvalidClassException, NullPointerException, etc.
6.	These exceptions are propagated using the throws keyword.	These are automatically propagated.
7.	It is required to provide the try-catch and try-finally block to handle the checked exception.	In the case of unchecked exception it is not mandatory.



References

- Java : the complete Reference (Eleventh Edition), Herbert Schildt, 2018.

