


JAVA Unchecked Exception aka Runtime Exceptions


1. NullPointerException

Happens when you try to use a `null` reference.

```
public class NullPointerExceptionDemo {  
    public static void main(String[] args) {  
        String str = null;  
        System.out.println(str.length()); //  NullPointerException  
    }  
}
```

2. ArithmeticException

Occurs when you perform an illegal arithmetic operation (like divide by zero).

```
public class ArithmeticExceptionDemo {  
    public static void main(String[] args) {  
        int a = 10, b = 0;  
        System.out.println(a / b); //  ArithmeticException: / by zero  
    }  
}
```

3. ArrayIndexOutOfBoundsException

Thrown when trying to access an invalid index in an array.

```
public class ArrayIndexOutOfBoundsExceptionDemo {  
    public static void main(String[] args) {  
        int[] arr = {1, 2, 3};  
        System.out.println(arr[5]); //  ArrayIndexOutOfBoundsException  
    }  
}
```

4. NumberFormatException

Occurs when converting a string into a number, but the string is not properly formatted.

```
public class NumberFormatDemo {  
    public static void main(String[] args) {  
        String str = "abc";  
        int num = Integer.parseInt(str); // ❌ NumberFormatException  
    }  
}
```

5. ClassCastException

Happens when you try to cast an object into an incompatible type.

```
public class ClassCastDemo {  
    public static void main(String[] args) {  
        Object obj = "Hello";  
        Integer num = (Integer) obj; // ❌ ClassCastException  
    }  
}
```

Checked Exceptions :

1. IOException

Thrown when an input/output operation fails.

```
import java.io.*;

public class IOExceptionDemo {
    public static void main(String[] args) {
        try {
            FileReader file = new FileReader("test.txt"); // File may not
exist
            BufferedReader reader = new BufferedReader(file);
            System.out.println(reader.readLine());
            reader.close();
        } catch (IOException e) {
            System.out.println("IOException handled: " + e);
        }
    }
}
```

2. FileNotFoundException

A subclass of `IOException`, thrown when a file is not found.

```
import java.io.*;

public class FileNotFoundExceptionDemo {
    public static void main(String[] args) {
        try {
            FileInputStream fis = new FileInputStream("missing.txt"); //
File missing
        } catch (FileNotFoundException e) {
            System.out.println("FileNotFoundException handled: " + e);
        }
    }
}
```

3. ClassNotFoundException

Occurs when a class is not found while loading dynamically.

```
public class ClassNotFoundExceptionDemo {  
    public static void main(String[] args) {  
        try {  
            Class.forName("com.unknown.ClassName"); // Class doesn't exist  
        } catch (ClassNotFoundException e) {  
            System.out.println("ClassNotFoundException handled: " + e);  
        }  
    }  
}
```