

Phanthira Kositjaroenkul - 6630003  
Kaung Khant Lin - 6540131  
Thit Lwin Win Thant - 6540122

## Task1: Concept Check

1. What happens if an exception is thrown but not caught in Java?  
Provide a sample Java program for such a case.

If an exception is thrown and no surrounding code catches it, it bubbles up the call stack; when it reaches the top (e.g., main) without a match, the program terminates and the JVM prints an "Exception in thread 'main' ..." stack trace.

```
public class UncaughtDemo {  
    public static void main(String[] args) {  
        System.out.println("Before");  
        int x = 1 / 0;          // throws ArithmeticException  
        System.out.println("After"); // never reached  
    }  
}
```

2. What is the difference between RuntimeException and IOException?

	RuntimeException	IOException
Hierarchy and Type	It is a subclass of <code>Exception</code> , but more specifically a subclass of <code>RuntimeException</code> itself.  It represents unchecked exceptions.	It is a subclass of <code>Exception</code> .  Specifically, it is a checked exception.
Unchecked vs Checked	Does not need to be declared in the method signature with <code>throws</code> .  The compiler does not require you to handle or declare them.  Typically signals programming errors (for example, bad logic or API misuse).	Must be declared in the method signature, or the method must handle it with a try-catch block.  The compiler requires handling or declaration of these exceptions.  Typically signals external conditions that a program should anticipate and handle (like file not found, network errors, etc.).

Typical Causes	<p>Indicates errors in the program's logic which could have been avoided by the programmer.</p> <p>Examples: Null pointer access, array index out of bounds.</p>	<p>Signals problems during input/output operations, often due to external factors.</p> <p>Examples: Failed to read or write files, network connection interruptions.</p>
----------------	--	--

## Code Examples

### Unchecked (RuntimeException):

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

### Checked (IOException):

```
import java.io.*;

public class IOExceptionDemo {
    public static void main(String[] args) throws IOException {
        FileReader fr = new FileReader("nonexistent.txt"); // Throws IOException (checked)
    }
}
```

## Task 2: Exception Handling

- Line 5: `numbers[5]` → `ArrayIndexOutOfBoundsException` (actual index range 0–2). Because that exception is thrown, control jumps to the first catch.
- Lines 6–7 (division) never execute, so the `ArithmeticException` (divide by zero) does NOT occur during that run.
- Line 17 (divide) would throw `ArithmeticException` (division by zero) only if execution reached it.