**17th Day Internship Report**

Date: 24 June 2025

Topic: Exception Handling in Java

Today’s Learning Summary:

On the 24th day of the internship, we learned about Exception Handling in Java, an essential concept used to manage runtime errors and maintain normal program flow.

Key Concepts Covered:

1. What is an Exception?

An exception is an event that disrupts the normal flow of a program.

It is an object that represents an error.

2. Types of Exceptions:

Checked Exceptions: Caught at compile-time (e.g., IOException, SQLException)

Unchecked Exceptions: Occur at runtime (e.g., ArithmeticException, NullPointerException)

3. Try-Catch Block:

Used to handle exceptions gracefully.

try { int result = 10 / 0;

} catch (ArithmeticException e) {

System.out.println("Cannot divide by zero.");

}

4. Finally Block:

Executes always, whether exception occurs or not.

finally {

System.out.println("This block always executes.");

}

5. Throw and Throws:

throw: Used to explicitly throw an exception.

throws: Declares exceptions that a method might throw.

throw new ArithmeticException("Demo");

void method() throws IOException {

// code that may throw IOException

}

Conclusion:

We concluded that:

Exception handling is crucial for writing robust and error-resistant programs.

It helps in identifying, handling, and recovering from errors.

Proper use of try-catch-finally blocks improves program stability and user experience.

This session improved our ability to handle runtime errors systematically and write cleaner, more reliable code.