**13th Day Internship Report**

Date: 18 June 2025

Topic Covered: Java Packages – Default Package, Static Import, and User-Defined Package

🔍 What We Learned

Today, we explored different types of packages in Java and how they help organize and reuse code effectively:

**1. Default Package**

Definition: When no package is explicitly defined, the class belongs to the default package.

Usage: Suitable for small programs or quick testing.

Limitation: Classes in the default package cannot be imported into classes belonging to a named package.

**2. Static Import**

Definition: Allows members (fields and methods) defined in a class to be used without class qualification.

Syntax:

import static java.lang.Math.\*;

Example:

import static java.lang.Math.\*;

class Demo {

public static void main(String[] args) {

System.out.println(sqrt(16)); // No need to write Math.sqrt()

}

}

**3. User-Defined Package**

Definition: A package created by the user to organize related classes.

Steps to Create:

1. Declare the package at the top using package keyword.

2. Compile using the -d flag to set directory.

3. Import the package where needed.

Example:

File: mypack/Message.java

package mypack;

public class Message {

public void display() {

System.out.println("Hello from user-defined package!");

}

}

File: Test.java

import mypack.Message;

class Test {

public static void main(String[] args) {

Message m = new Message();

m.display();

}

}

✅ Conclusion

Today's session provided a clear understanding of how Java packages help manage large codebases.

Default packages are quick but limited.

Static imports improve code readability by eliminating repetitive class names.

User-defined packages promote modularity and reusability, essential for real-world software projects.