**15th Day Internship Report**

Date: 20 June 2025

Topic Covered: Threads in Java (Multithreading)

🔍 What We Learned

Today’s session focused on Threads in Java, an essential concept for achieving multitasking and improving the performance of Java applications.

✅ What is a Thread?

A thread is a lightweight sub-process; it is the smallest unit of execution.

Java provides built-in support for multithreading, allowing multiple threads to run concurrently.

🔧 Creating Threads in Java

There are two main ways to create a thread in Java:

1. By Extending the Thread Class

class MyThread extends Thread {

public void run() {

System.out.println("Thread is running...");

}

}

public class TestThread {

public static void main(String[] args) {

MyThread t = new MyThread();

t.start(); // Starts the new thread

}

}

2. By Implementing the Runnable Interface

class MyRunnable implements Runnable {

public void run() {

System.out.println("Runnable thread is running...");

}

}

public class TestRunnable {

public static void main(String[] args) {

Thread t = new Thread(new MyRunnable());

t.start();

}

}

🔄 Important Thread Methods

start() – Starts a new thread.

run() – Entry point for thread execution.

sleep(ms) – Pauses execution for specified time.

join() – Waits for a thread to die.

isAlive() – Checks if thread is alive.

setPriority() – Sets thread priority (MIN\_PRIORITY, NORM\_PRIORITY, MAX\_PRIORITY).

✅ Conclusion

Today, we gained practical knowledge of how multithreading works in Java.

Threads enable parallel execution, improving the performance of applications, especially when tasks can be performed simultaneously. Understanding threading is critical for building efficient, responsive, and modern Java programs.