

**PROJECT REPORT**

**JAVA PROGRAMMING FUNDAMENTALS**

**(EBDS22ET2)**

2024-2025(EVEN SEMESTER)

**DEPARTMENT OF B.Tech DS & AI (E&T)**

**COURSE : B-TECH CSE-(DS &AI)**

**YEAR/SEM/SEC : Ist year / IInd sem / AB1**

**PROJECT TITLE : Lifecycle of threads**



**BONAFIDE CERTIFICATE**

**JAVA PROGRAMMING FUNDAMENTALS**

**DEPARTMENT OF B.Tech DS & AI (E&T)**

Certified that this project report **“ Lifecycle of threads ”** is confirmed work of **V.BALAMURUGAN AB1-18,ABISHEK SUNIL SARASWATHY AB1-15,AJESH RK AB1-22** I-year B-Tech CSE- (DS&AI) in **JAVA PROGRAMMING FUNDAMENTALS (EBDS22ET2)** who carried out the project work under the supervision

Signature of Lab-in-Charge Signature of Head of Dept

Submitted for the Practical Examination held on

Internal Examiner External Examiner

# ABSTRACT

* In Java, multithreading allows multiple threads to execute concurrently.
* A thread passes through various stages during its execution known as the thread life cycle.
* Understanding these stages helps developers manage thread behavior and optimize performance.
* This presentation explores each life cycle phase along with practical implementation.

📘 INTRODUCTION TO JAVA:

Java is a **high-level, object-oriented programming language** developed by Sun Microsystems (now owned by Oracle Corporation). It was designed with the principle of **"Write Once, Run Anywhere" (WORA)**, which means Java programs can run on any device equipped with the Java Virtual Machine (JVM), regardless of the underlying hardware or operating system.

Java is widely used for building a range of applications — from web and desktop software to mobile apps and embedded systems. Its **strong memory management, multithreading support, platform independence,** and a vast collection of libraries make it a preferred choice for developers.

In this project, Java is used to create a **console-based chatbot**, demonstrating key features of the language such as:

* **Input/output handling** using the Scanner class
* **Conditional and control statements** for decision-making
* **Loops** for continuous user interaction
* **String manipulation** and **array operations**
* **Use of standard libraries** like java.util, java.time, and java.lang

This project illustrates how Java can be employed to build **interactive and user-friendly command-line applications**, reinforcing fundamental programming concepts in a practical and engaging manner.

# PROJECT TITLE

**AIM:**

* **To understand and demonstrate the different states in the Java thread life cycle.**
* **To implement Java code that shows transitions between thread states.**

**ALGORITHMS:**

## Create a class extending Thread or implementing Runnable.

## Override the run() method.

## Create a thread instance using the class.

## Use start() to move thread from New to Runnable.

## Demonstrate sleep (Timed Waiting), join (Waiting), etc.

## Let the thread complete execution to reach Terminated state.

**SOURCE CODE:**

**class MyThread extends Thread {**

**public void run() {**

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

**try {**

**Thread.sleep(1000); // Timed Waiting**

**} catch (InterruptedException e) {**

**System.out.println("Thread interrupted.");**

**}**

**System.out.println("Thread finished.");**

**}**

**public static void main(String[] args) {**

**MyThread t1 = new MyThread();**

**System.out.println("Thread state after creation: " + t1.getState()); // NEW**

**t1.start();**

**System.out.println("Thread state after start(): " + t1.getState()); // RUNNABLE**

**try {**

**t1.join(); // Main thread waits**

**} catch (InterruptedException e) {**

**e.printStackTrace();**

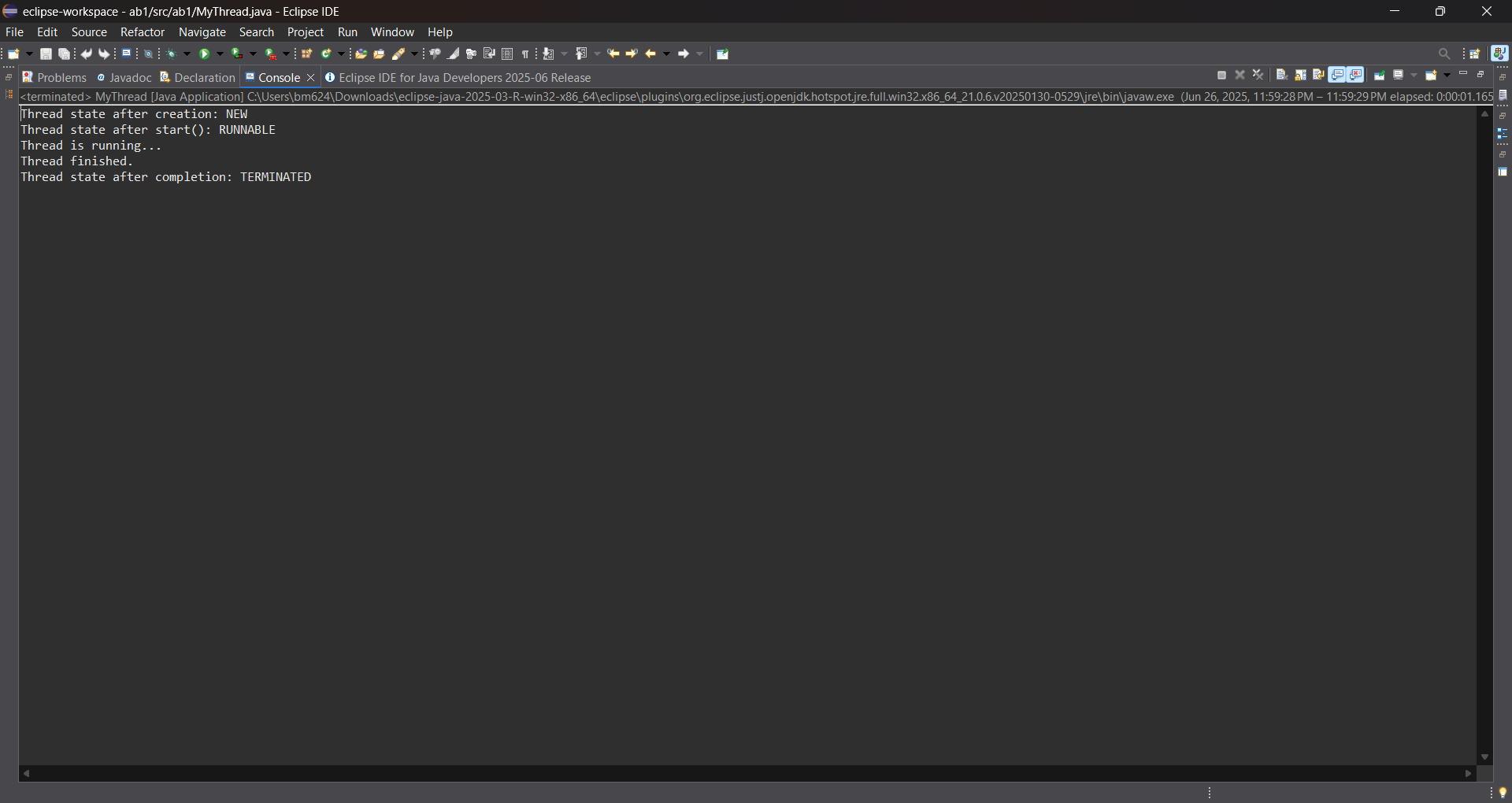
**}**

**System.out.println("Thread state after completion: " + t1.getState()); // TERMINATED**

**}**

**}**

**OUTPUT:**

****

**RESULT:**

## 

* The code successfully demonstrates thread life cycle states:
  + NEW before start()
  + RUNNABLE during execution
  + TIMED\_WAITING during sleep()
  + TERMINATED after run() completes
* It helps visualize thread behavior for better multithreaded programming.

**Git huh link :** [**https://github.com/balamurugangethub/life-cycle-of-thread.git**](https://github.com/balamurugangethub/life-cycle-of-thread.git)

**abishek github link:** [**https://github.com/abisheksunilgit/life-cycle-of-a-thread-in-java.git**](https://github.com/abisheksunilgit/life-cycle-of-a-thread-in-java.git)