Day 10 - 20th June- 113195064 - Bondili Bhavya Charitha

My VM is not working properly so I have worked with online compiler.

**TASK01: What is a Process?**

A process is like a little worker in my computer. Every time when I open an app, run a program, or even launch a browser tab, my computer spins up one of these workers — a process — to handle that specific job.

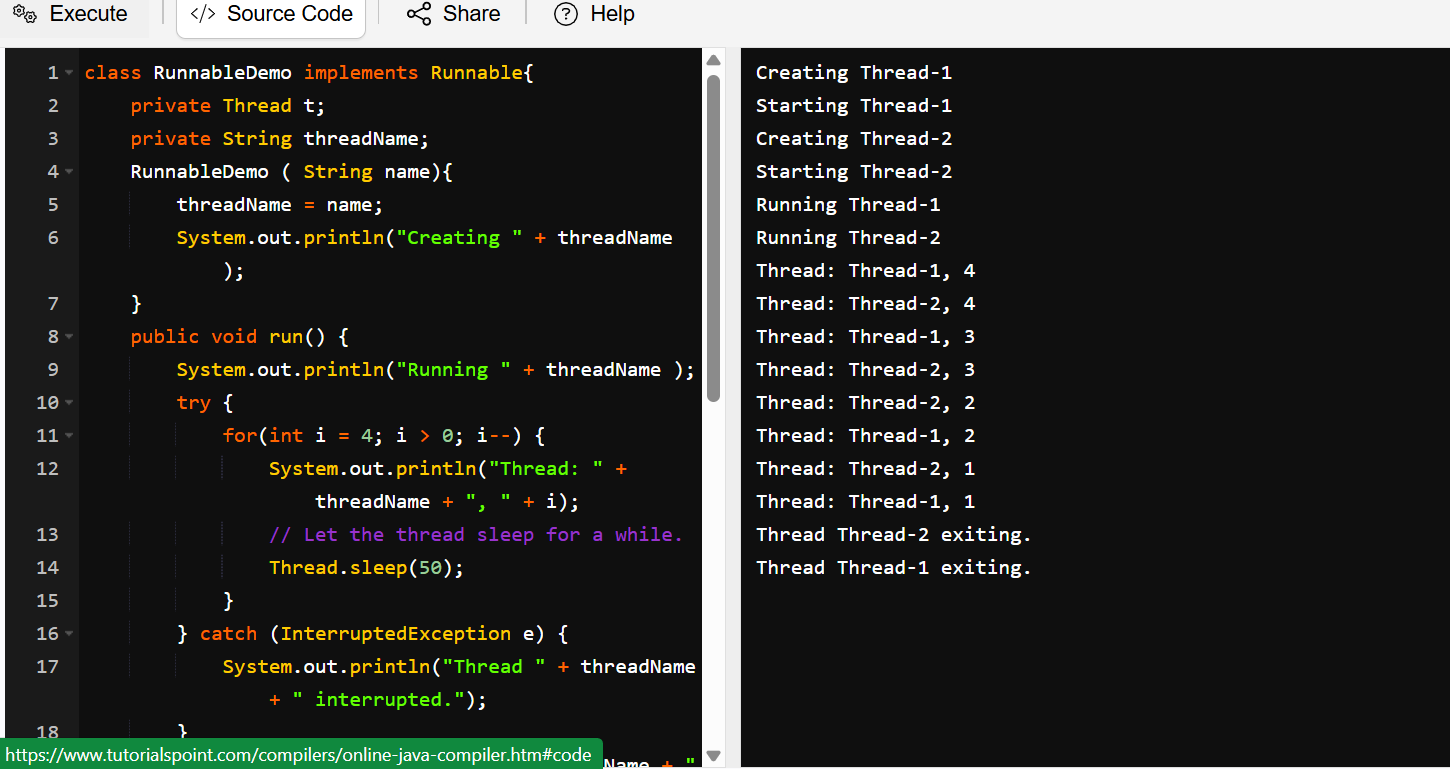
**Ex:** I am running a kitchen. Every dish i prepare is like a program, and each chef i assign to cook that dish is a process. They're focused, have their own workspace, tools (memory), and know exactly what they’re cooking (instructions). Once the dish is done, the chef (process) clocks out.

**TASK02: What is a Thread?**

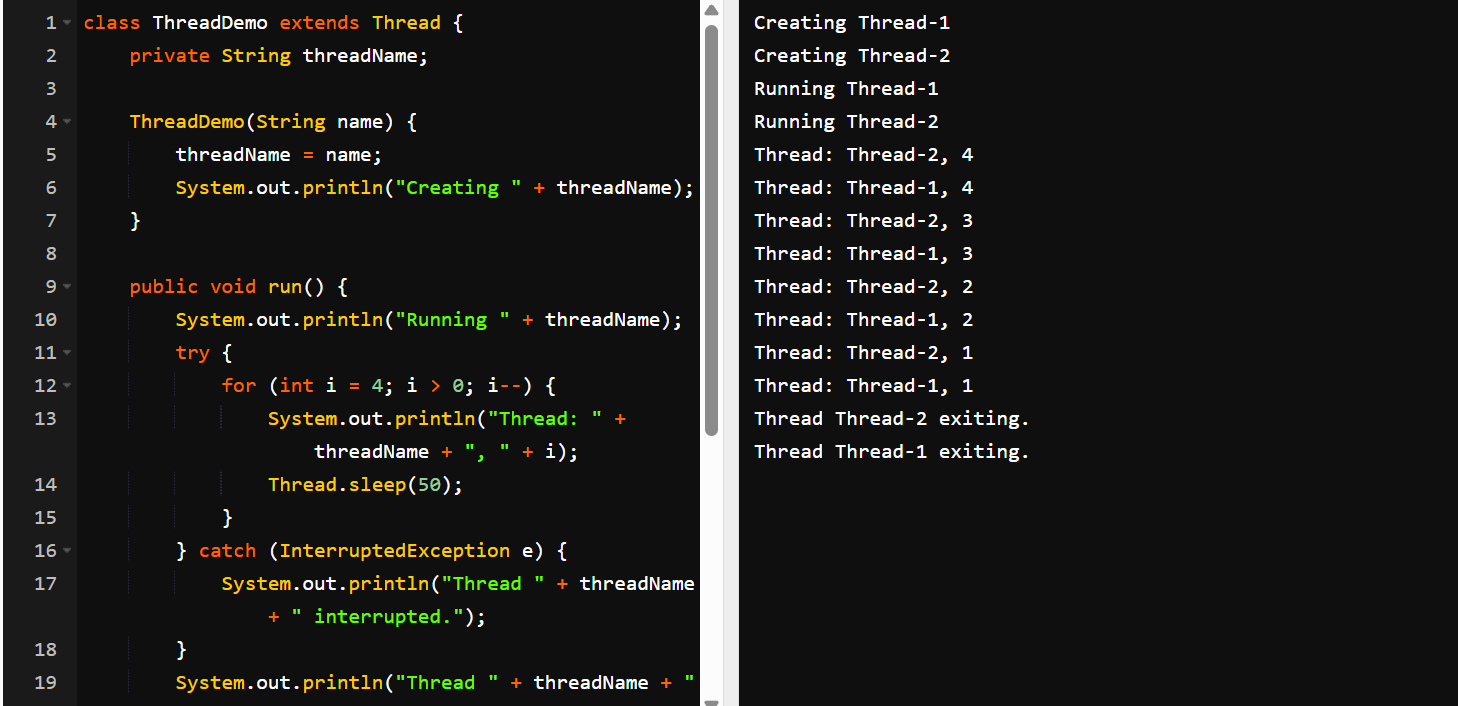
A thread is like a little helper inside a running program (a process). If a process is my whole Netflix app, then threads are the different things it's doing at once—like streaming video, loading subtitles, and buffering—all happening simultaneously without crashing my party.

In simple terms, a process is the full program and a thread is a mini-task running inside the program.

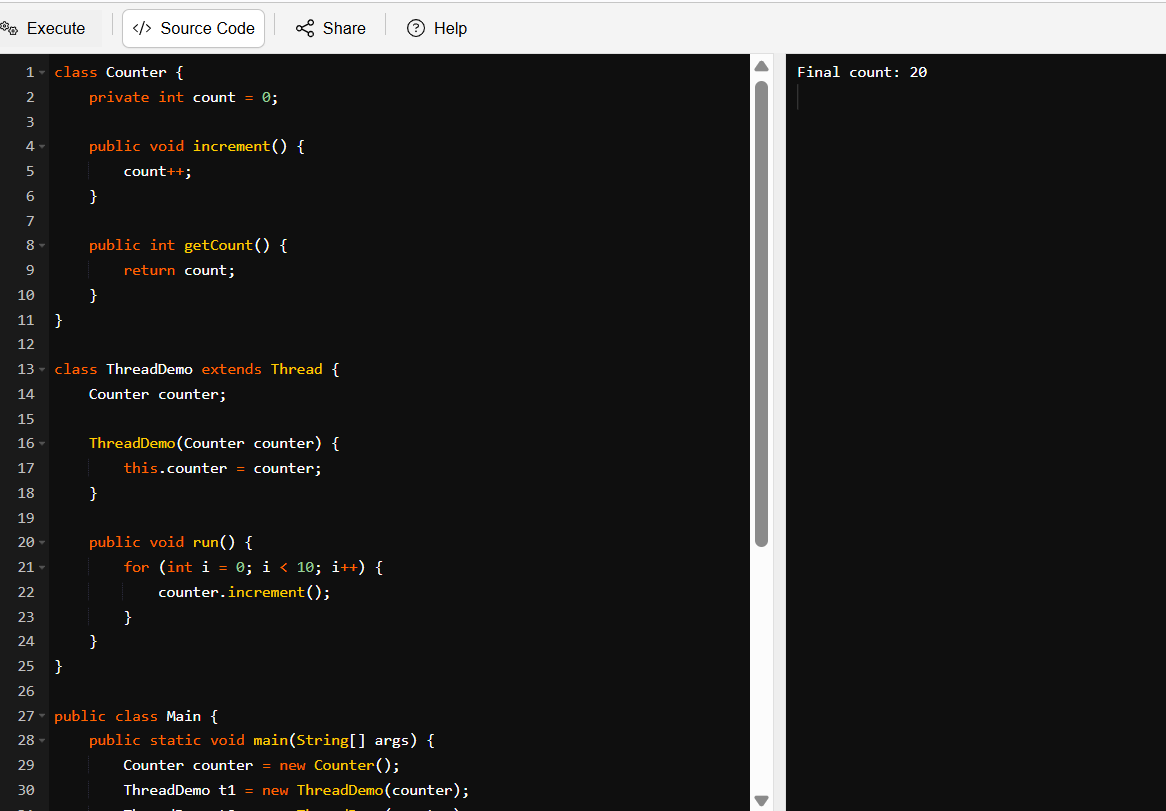
**TASK03:**

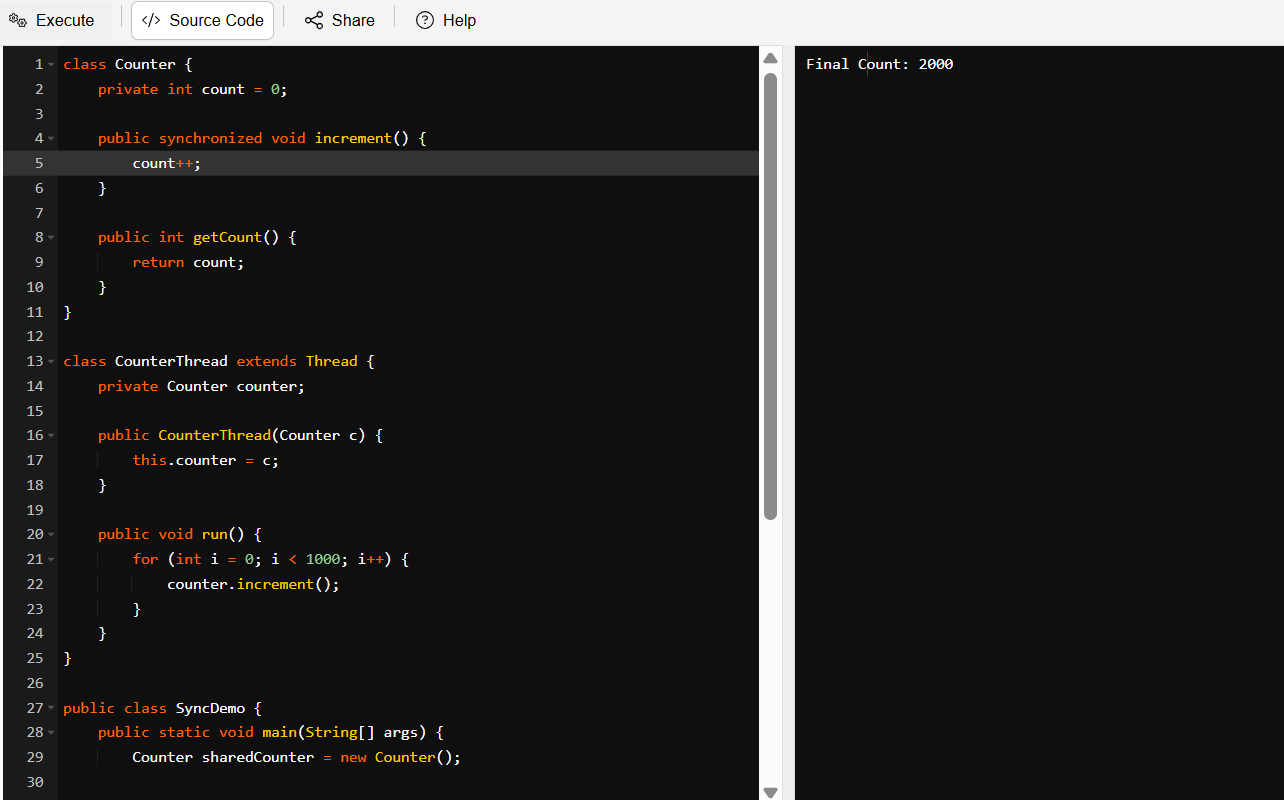


**TASK04:**

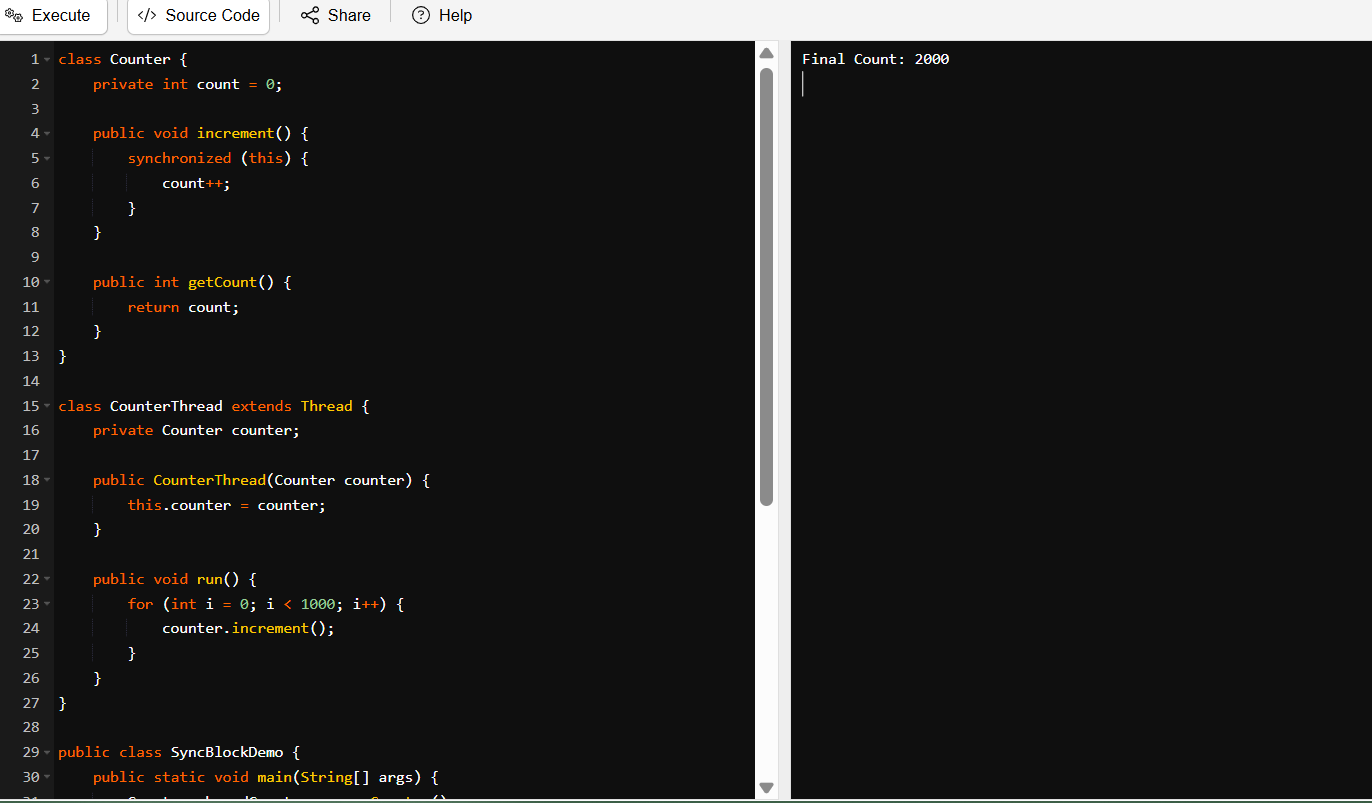
****

**TASK05:**

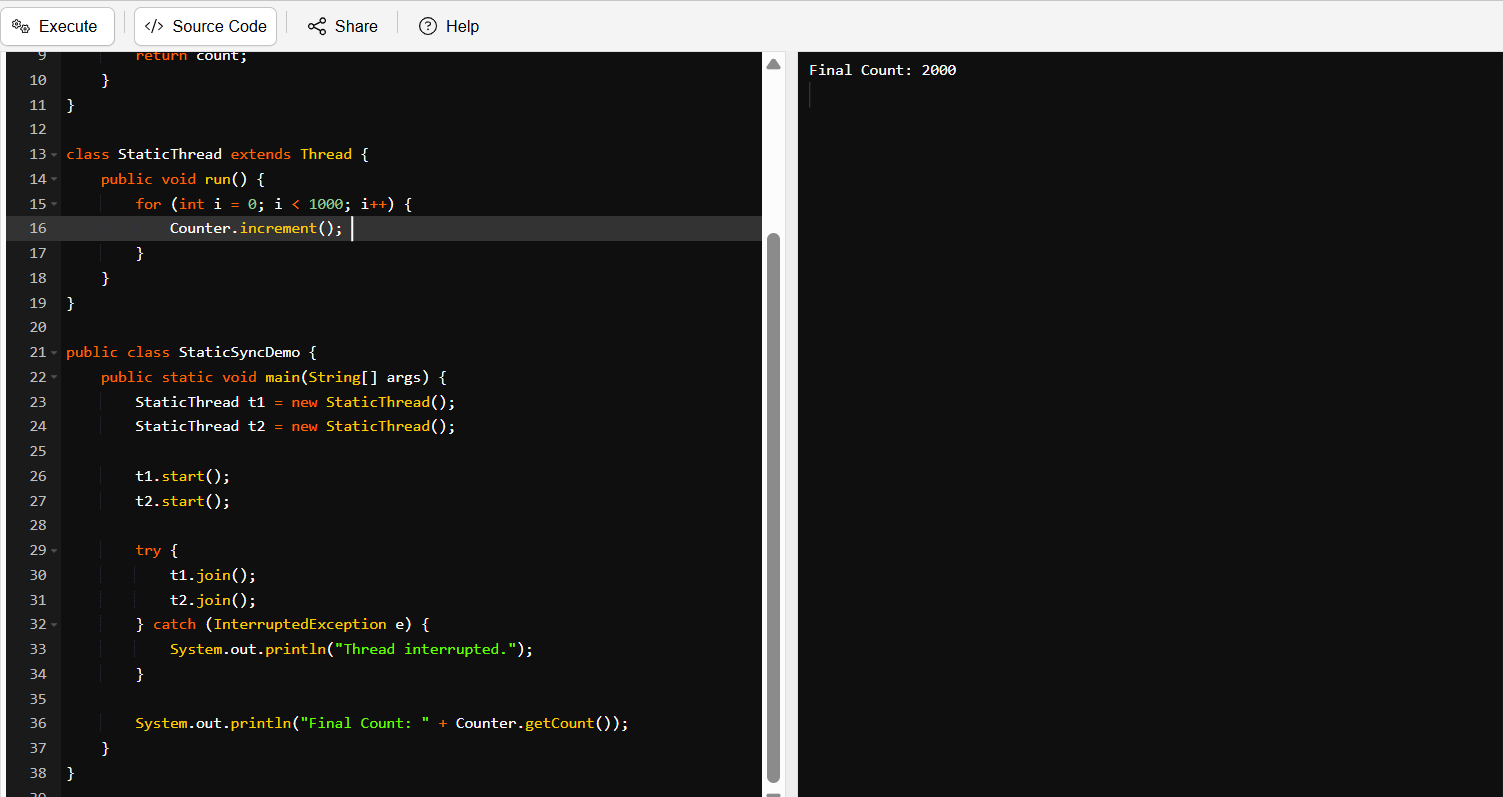
****

**TASK06:** ****

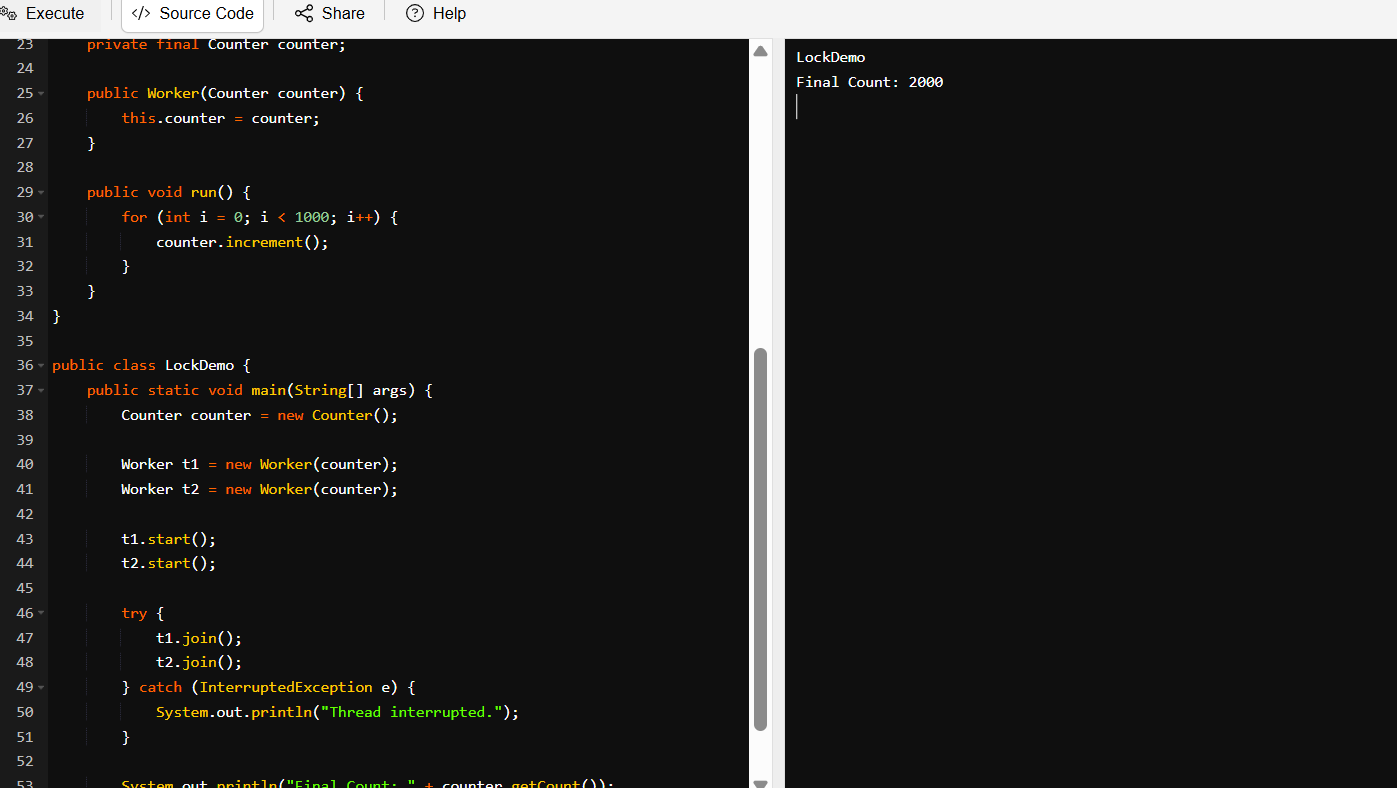
**TASK07:**

****

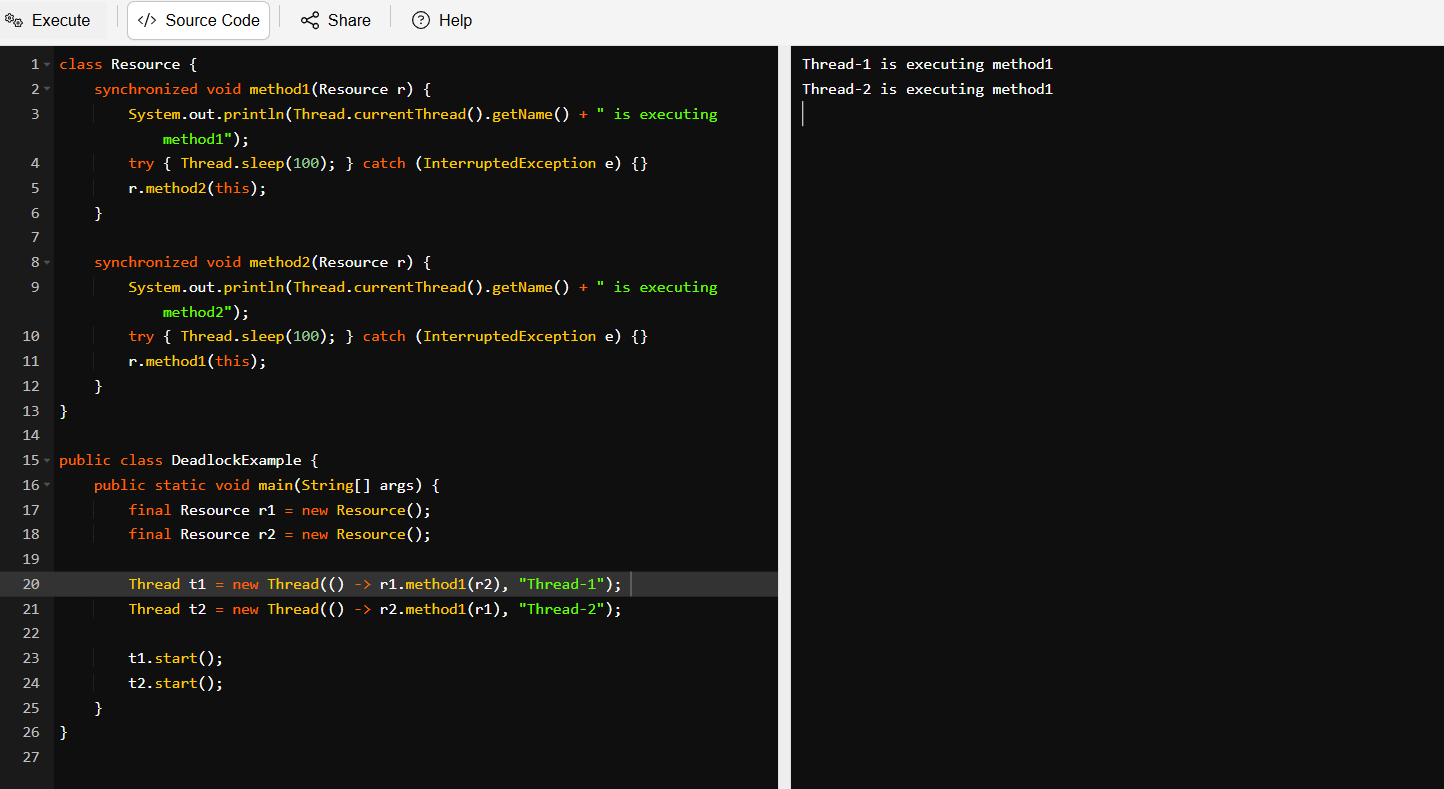
**TASK08:**

****

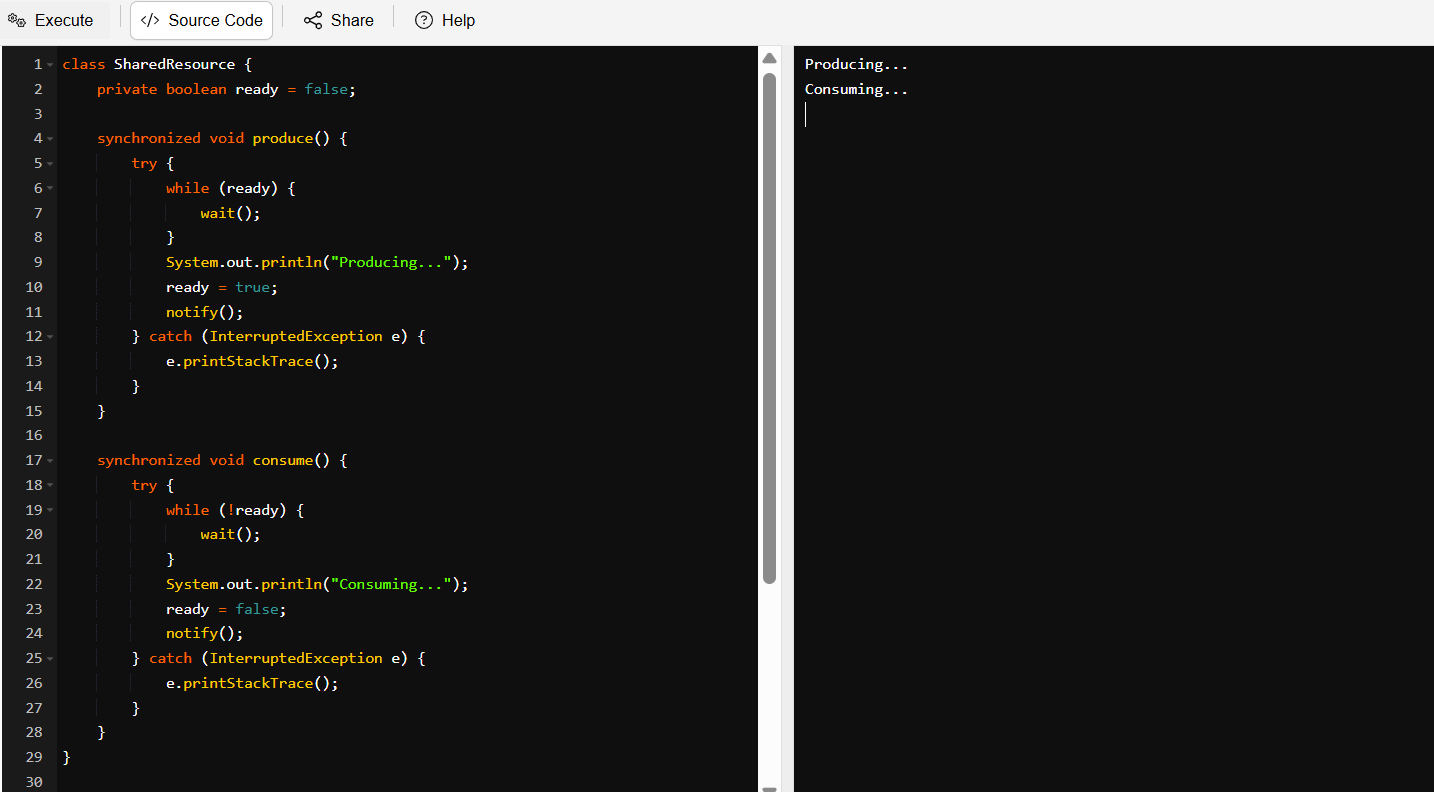
**TASK09:**

****

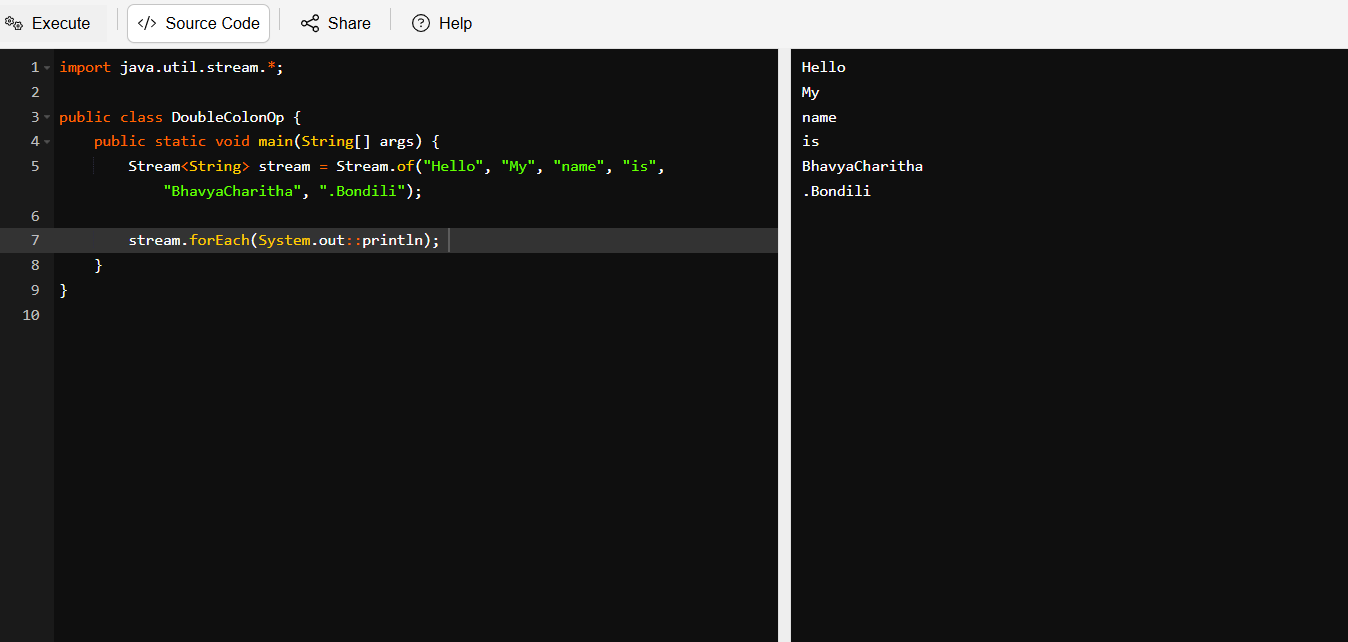
**TASK10:**

****

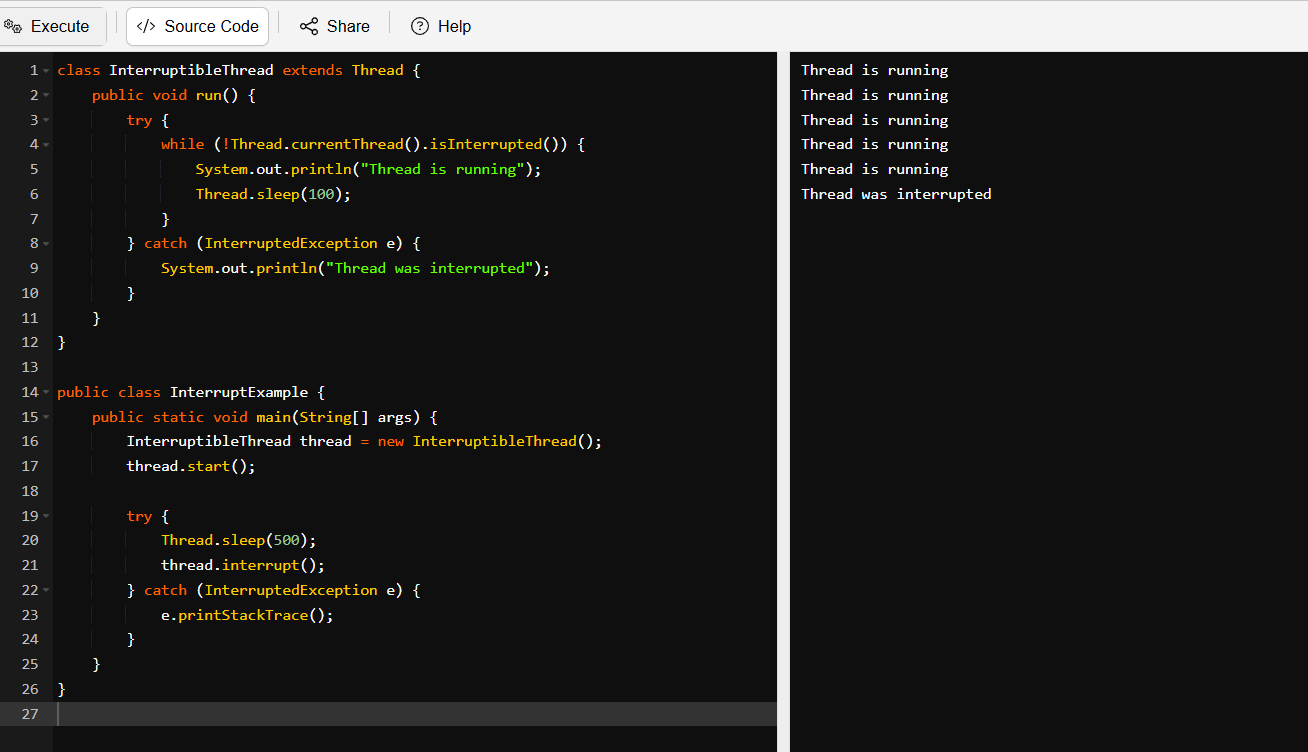
**TASK11:**

****

**TASK12:**

****

**TASK13:**

****

**TASK14: What are Daemon Threads?**

Daemon Thread is a background helper. I am cooking food; I am main thread here. I asked my husband to clean utensils, while I am cooking food. He is a daemon 😈. Once I finish my cooking, I will leave everything like the main work is done, program ends automatically even daemon is still working.

**TASK15:**

Debugging Tools in Java – List Down Few:

* Eclipse Debugger – Built into Eclipse IDE; lets you pause your code, inspect variables, and step through it like a detective solving a code mystery.
* IntelliJ IDEA Debugger – Super sleek debugger inside IntelliJ. You can set breakpoints, watch expressions, and even jump back in time (kind of).
* JDB (Java Debugger) – The command-line boss of debugging. A bit old-school but great when you're working without an IDE.
* Your Brain + `System.out.println()` – The classic "print and pray" method. Not elegant, but hey, it works in a pinch.

**TASK16:**

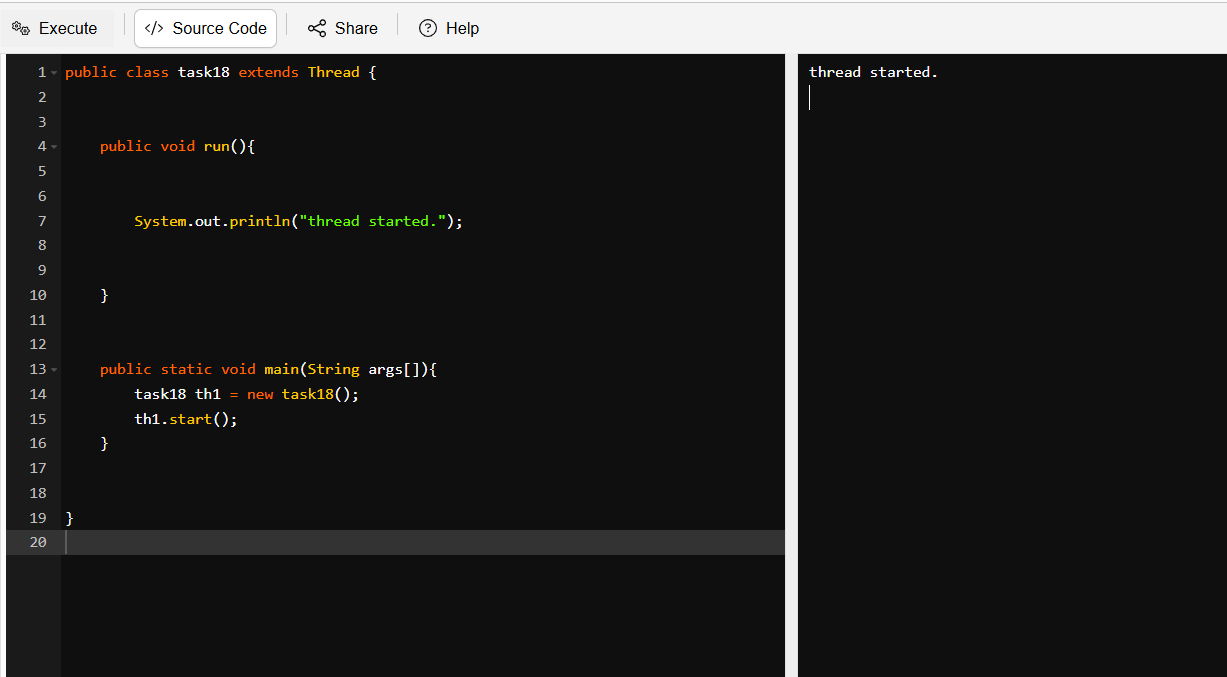
Understanding Error Messages – Done

**TASK17:**

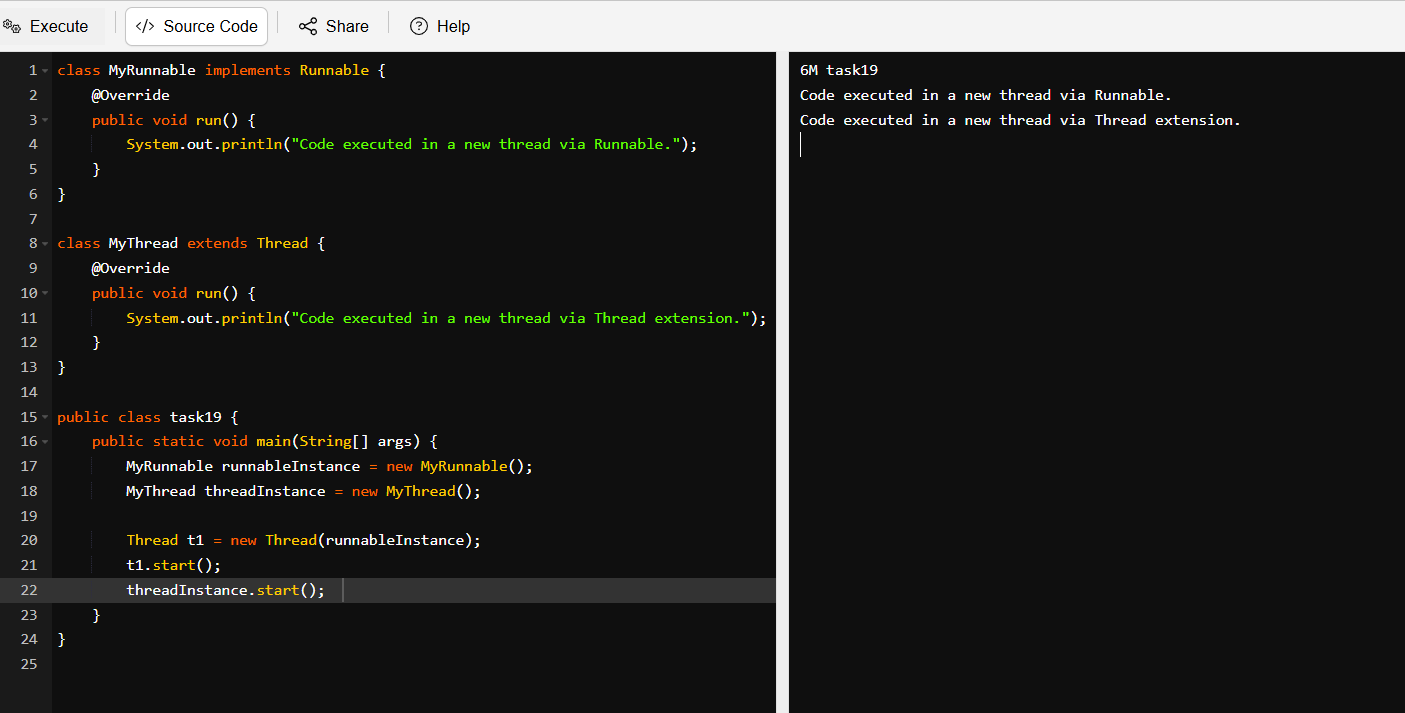
What is Stack Trace? What will it do?

* A stack trace is like a breadcrumb trail left behind when something crashes in your code. It’s a list of function or method calls that were in progress when the program hit an error. Think of it as the “black box” recording of a program’s last moments before an accident.

**TASK18:**



**TASK19:**



**TASK20:**

