

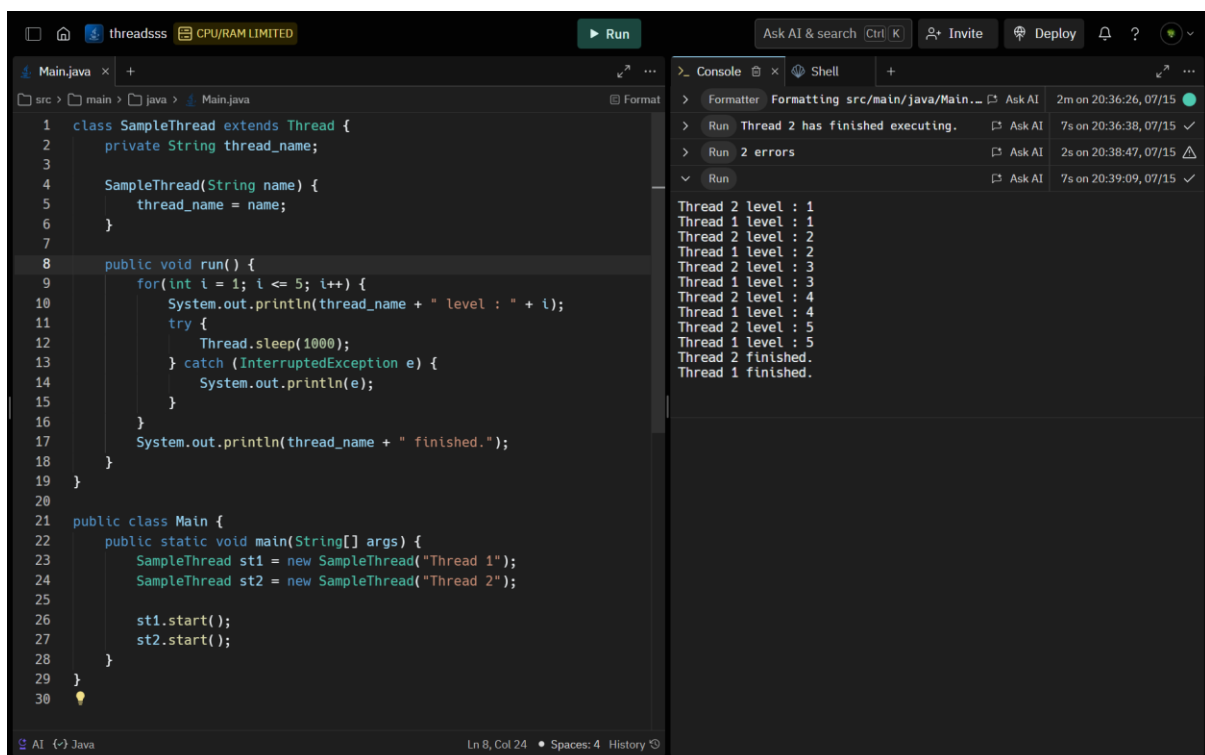
# OS LAB WORK

## 2 Marks Add-on program: Implement the threads using Java Code.

BY ADITYA MADWAL

2200290120011

Use the concepts of Object oriented programming to implement the concepts of Threads. Show the multi-threading, as discussed in Java Lab.



The screenshot displays a Java IDE with a file named `Main.java` open. The code defines a `SampleThread` class that extends `Thread` and implements the `run()` method. The `run()` method contains a loop that prints the thread's name and level (from 1 to 5), sleeps for 1000ms, and then prints "finished.". The `Main` class creates two instances of `SampleThread`, `st1` and `st2`, and starts them. The IDE's console shows the output of the program, which is interleaved, demonstrating concurrent execution. The output shows `Thread 2` finishing its execution before `Thread 1` does, indicating that the threads are running in parallel.

```
1 class SampleThread extends Thread {
2     private String thread_name;
3
4     SampleThread(String name) {
5         thread_name = name;
6     }
7
8     public void run() {
9         for(int i = 1; i <= 5; i++) {
10             System.out.println(thread_name + " level : " + i);
11             try {
12                 Thread.sleep(1000);
13             } catch (InterruptedException e) {
14                 System.out.println(e);
15             }
16         }
17         System.out.println(thread_name + " finished.");
18     }
19 }
20
21 public class Main {
22     public static void main(String[] args) {
23         SampleThread st1 = new SampleThread("Thread 1");
24         SampleThread st2 = new SampleThread("Thread 2");
25
26         st1.start();
27         st2.start();
28     }
29 }
30
```

Console Output:

```
Thread 2 level : 1
Thread 1 level : 1
Thread 2 level : 2
Thread 1 level : 2
Thread 2 level : 3
Thread 1 level : 3
Thread 2 level : 4
Thread 1 level : 4
Thread 2 level : 5
Thread 1 level : 5
Thread 2 finished.
Thread 1 finished.
```