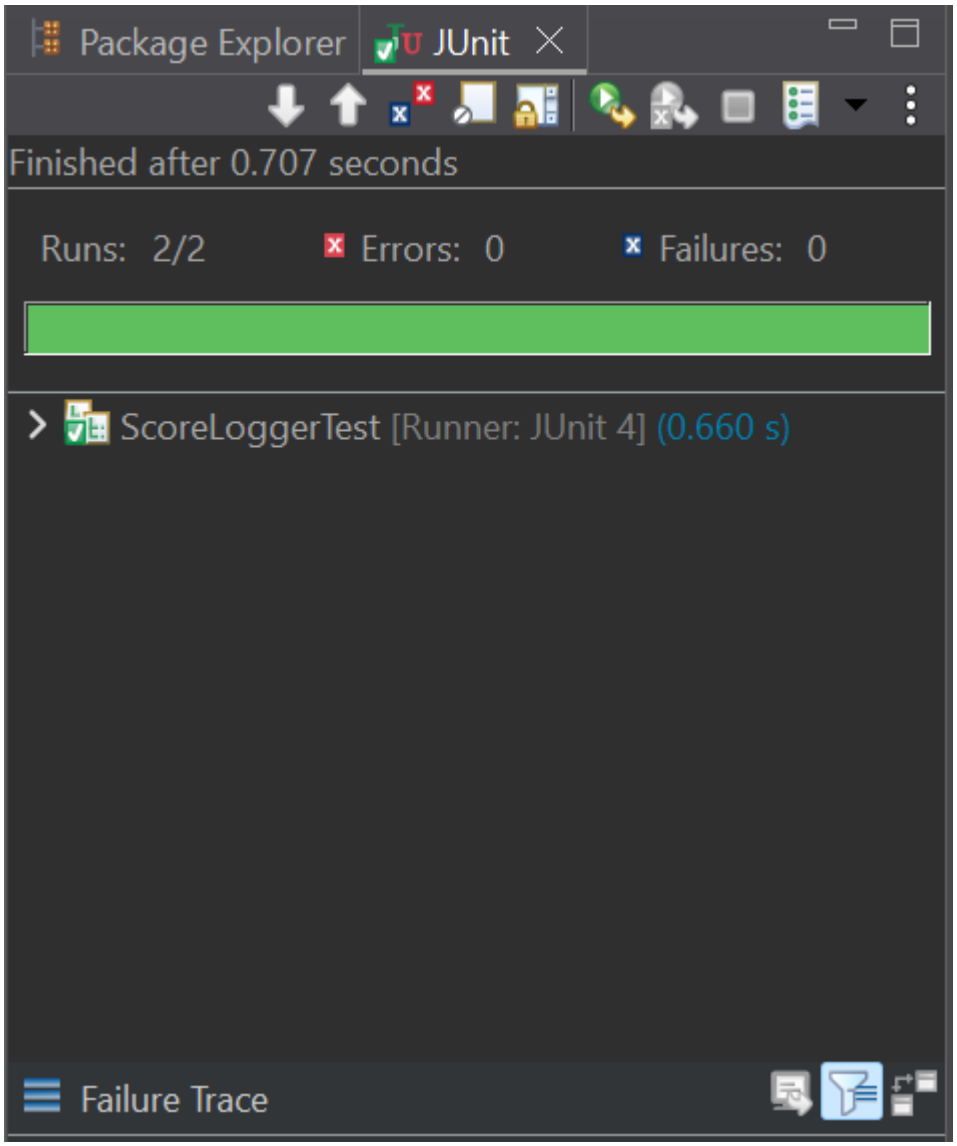
```
public class Student {
    private String name;
    private int rollNo;
    private int marks;

    public Student(String name, int rollNo, int marks) {
        this.name = name;
        this.rollNo = rollNo;
        this.marks = marks;
    }

    public String toCSV() {
        return name + "," + rollNo + "," + marks + "\n";
    }

    @Override
    public String toString() {
        return "Student{name='" + name + "', rollNo=" + rollNo + ", marks=" + marks + "}";
    }
}
import java.io.FileWriter;
import java.io.IOException;

public class ScoreLogger {
    private final String filePath;

    public ScoreLogger(String filePath) {
        this.filePath = filePath;
    }

    // synchronized to avoid multiple threads writing at same time
```

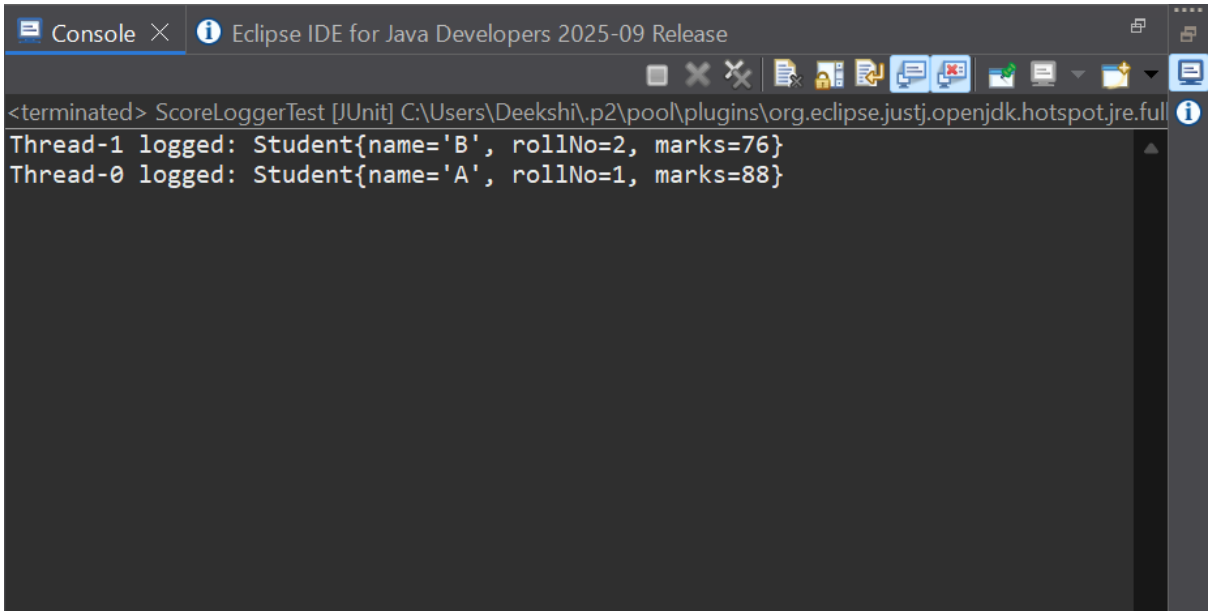
```
    public synchronized void writeScore(Student s) throws IOException {
        try (FileWriter fw = new FileWriter(filePath, true)) {
            fw.write(s.toCSV());
        }
    }
}

public class StudentWorker extends Thread {
    private final Student student;
    private final ScoreLogger logger;

    public StudentWorker(Student student, ScoreLogger logger) {
        this.student = student;
        this.logger = logger;
    }

    @Override
    public void run() {
        try {
            // Simulate time delay for realism
            Thread.sleep((int)(Math.random() * 1000));
            logger.writeScore(student);
            System.out.println(Thread.currentThread().getName() + " logged: " + student);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
public class Main {
    public static void main(String[] args) {
        String file = "student_scores.csv";
```

```
        ScoreLogger logger = new ScoreLogger(file);

        // Create few student threads
        StudentWorker t1 = new StudentWorker(new Student("Deekshi", 64, 85), logger);
        StudentWorker t2 = new StudentWorker(new Student("Arjun", 21, 92), logger);
        StudentWorker t3 = new StudentWorker(new Student("Megha", 45, 78), logger);

        t1.start();
        t2.start();
        t3.start();

        try {
            t1.join();
            t2.join();
            t3.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("✅ All threads finished logging scores!");
    }
}

import static org.junit.Assert.*;
import org.junit.Test;
import java.io.*;
import java.nio.file.*;
import java.util.List;

public class ScoreLoggerTest {
```

```
public void testFileWrite() throws Exception {
    String path = "test_scores.csv";
    Files.deleteIfExists(Paths.get(path));

    ScoreLogger logger = new ScoreLogger(path);
    Student s = new Student("Riya", 10, 95);
    logger.writeScore(s);

    List<String> lines = Files.readAllLines(Paths.get(path));
    assertTrue(lines.get(0).contains("Riya"));
}


public void testThreadCompletion() throws Exception {
    String path = "multi_thread.csv";
    Files.deleteIfExists(Paths.get(path));

    ScoreLogger logger = new ScoreLogger(path);
    StudentWorker t1 = new StudentWorker(new Student("A", 1, 88), logger);
    StudentWorker t2 = new StudentWorker(new Student("B", 2, 76), logger);

    t1.start();
    t2.start();

    t1.join();
    t2.join();

    List<String> lines = Files.readAllLines(Paths.get(path));
    assertEquals(2, lines.size());
}
```

}


Package Explorer
JUnit
Finished after 0.707 seconds
Runs: 2/2
Errors: 0
Failures: 0
ScoreLoggerTest [Runner: JUnit 4] (0.660 s)
Failure Trace


Console
Eclipse IDE for Java Developers 2025-09 Release
<terminated> ScoreLoggerTest [JUnit] C:\Users\Deekshi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.ful
Thread-1 logged: Student{name='B', rollNo=2, marks=76}
Thread-0 logged: Student{name='A', rollNo=1, marks=88}