Homework on The Threading API and Thread Priorities

16 February 2026 19:51

1. Smart Classroom Simulation

Design a Java program simulating a smart classroom system using threads:

Threads:

- Attendance system
- Projector display
- Lecture recording

Requirements:

- Each thread runs independently
- Use sleep() to simulate real-time activity
- Display thread state changes

1. Online Exam System

Create a multi-threaded program where:

- One thread counts exam time (countdown)
- One thread accepts student answers
- One thread auto-submits when time ends

Concepts to use:

  - Runnable interface
  - join()
  - Thread lifecycle

3. CPU Scheduler Demonstration

Simulate CPU scheduling using thread priorities.

Create three threads:

- System process (MAX priority)
- User process (NORM priority)
- Background process (MIN priority)

Each prints execution steps.

Observe and explain execution behavior.

4. Parallel Mathematical Processor

Create a Java program that computes simultaneously:

Thread 1 → Factorial

Thread 2 → Fibonacci

Thread 3 → Prime numbers

All results must be displayed only after all threads finish (use join()).

5. Bank Transaction Monitor

Simulate bank account monitoring using threads:

Thread A → Deposits

Thread B → Withdrawals
Thread C → Balance display
Requirements:
- Display thread name
- Use sleep delays
- Demonstrate concurrent execution

6. Thread Lifecycle Visualizer
Write a Java program that clearly demonstrates transitions:
NEW → RUNNABLE → RUNNING → TIMED_WAITING → TERMINATED
Print state at each step.

7. Parent–Child Thread Control Systemem
Design a program where:
- Parent thread creates two child threads
- Parent waits until children complete
- Children perform different tasks

Demonstrate:
  - Priority inheritance
  - join()
  - State tracking

8. Real-Time Logger System
Simulate a logging system:
Thread 1 → Error logs
Thread 2 → Warning logs
Thread 3 → Info logs
Each thread prints messages at different time intervals.

9. Multi-Stage Task Execution
Simulate a production pipeline:
Stage 1 → Data generation
Stage 2 → Data processing
Stage 3 → Result output
Each stage runs in separate thread.
Ensure order using join().

10. Thread-Based Countdown Timer
Create a countdown timer thread that:
- Counts from N to 0
- Displays message when finished
- Main thread performs another task simultaneously

11. Thread Priority Analyzer

Write a Java program to:

- Create 5 threads with different priorities
- Each thread prints its execution time
- Compare scheduling behavior

12. Multi-User Chat Simulation

Simulate users sending messages:
Each user = separate thread
Messages appear with delay
Display:

- Thread name
- Message number

13. Background Auto-Save System

Design a program where:

- Main thread simulates typing work
- Background thread auto-saves every 3 seconds

14. Thread Performance Comparator

Create two implementations:
1. Using Thread class
2. Using Runnable interface
Both perform same task.
Compare design flexibility.

15. Complete Thread Management System

Build a menu-driven program:
Menu:
1. Create thread
2. Start thread
3. Set priority
4. Display state
5. Wait for completion
6. Exit
Use real thread objects.