JAVA MULTITHREADING ASSIGNMENT SHEET (All Levels)

LEVEL 1 — THREAD FUNDAMENTALS

- Q1 – Thread Creation (extends Thread & implements Runnable)
- Q2 – Thread Names and Priorities
- Q3 – Sleep and Yield
- Q4 – Join Example
- Q5 – Daemon Thread
- Q6 – Thread States

LEVEL 2 — SYNCHRONIZATION & DATA SAFETY

- Q7 – Race Condition Demo
- Q8 – Synchronized Method vs Block
- Q9 – Static Synchronization
- Q10 – Deadlock
- Q11 – Using ReentrantLock
- Q12 – Thread-Safe Collections

LEVEL 3 — INTER-THREAD COMMUNICATION

- Q13 – Producer–Consumer (wait/notify)
- Q14 – Producer–Consumer (BlockingQueue)
- Q15 – Spurious Wakeups

LEVEL 4 — EXECUTOR FRAMEWORK

- Q16 – Fixed Thread Pool
- Q17 – Callable and Future
- Q18 – invokeAll() and invokeAny()
- Q19 – Executor Shutdown
- Q20 – ScheduledExecutorService
- Q21 – Custom ThreadPoolExecutor

LEVEL 5 — ADVANCED CONCURRENCY UTILITIES

- Q22 – AtomicInteger Counter
- Q23 – CountDownLatch
- Q24 – CyclicBarrier
- Q25 – Semaphore
- Q26 – Phaser
- Q27 – ConcurrentHashMap
- Q28 – CompletableFuture Chain
- Q29 – Parallel Stream

LEVEL 6 — REAL-WORLD MINI PROJECTS

- Q30 – File Downloader (Simulation)
- Q31 – Web Crawler Simulator

- Q32 – Async Image Processor
- Q33 – Database Connection Pool
- Q34 – Restaurant Order System
- Q35 – Race Simulation

■ HOW TO USE THIS ASSIGNMENT SHEET

1■■ Create a folder `/Multithreading-Practice/` 2■■ Each question → one `.java` file (e.g., `Q07_RaceConditionDemo.java`) 3■■ Maintain a README.md with learnings per level 4■■ Run each code → note console behavior 5■■ Refactor using modern APIs (`Executor`, `CompletableFuture`, `Streams`) once confident