10. QB - Multithreading assignment

🧠 Java Multithreading Assignment Questions

1. Create a Thread by Extending Thread Class

Question:

Write a program that creates a thread by extending the `Thread` class. The thread should print numbers from 1 to 10 with a delay of 500ms between each number.

---

2. Create a Thread by Implementing Runnable Interface

Question:

Write a program that creates a thread by implementing the `Runnable` interface. The thread should print the squares of numbers from 1 to 5.

---

3. Multithreaded Program with Two Threads

Question:

Create two threads:

- One prints even numbers from 0 to 10
- Another prints odd numbers from 1 to 9
  Make sure both threads run concurrently.

---

4. Thread Sleep and Join

Question:

Write a program that demonstrates the use of `sleep()` and `join()` methods. Start two threads and use `join()` to ensure one thread waits for another to complete.

---

5. Synchronization Example

Question:

Create a class with a synchronized method that prints a multiplication table (e.g., 5 * 1 = 5 to 5 * 10 = 50). Start two threads that call this method simultaneously for different numbers (e.g., 5 and 7). Use synchronization to avoid interleaved output.

---

6. Bank Account Simulation with Synchronization

Question:

Create a class `BankAccount` with a method `withdraw(int amount)`. Start two threads trying to withdraw money from the same account simultaneously. Use synchronization to avoid inconsistent balances.

---

7. Inter-Thread Communication (Producer-Consumer Problem)

Question:

Write a Java program that implements inter-thread communication between a producer and consumer using `wait()` and `notify()`. The producer should produce data and the consumer should consume it.

---

8. Daemon Thread Example

Question:

Create a daemon thread that prints a message every second. Also create a user thread that runs for 5 seconds. Observe the behavior of the daemon thread when the user thread finishes.

---

9. Thread Priorities

Question:

Create three threads with different priorities: MIN_PRIORITY, NORM_PRIORITY, and MAX_PRIORITY. Observe and explain the order in which they execute.

---

10. Deadlock Situation

Question:

Write a Java program that demonstrates a deadlock between two threads using two synchronized blocks. Then, explain how deadlock can be avoided.

Document by ***Suyash*** 😇