Week 10 - LAQ's

Instructions

Explain thread priorities.

In Java, thread priorities are a way to indicate the relative importance of threads to the thread scheduler. The thread scheduler uses these priorities to determine the order in which threads are executed. However, it's important to note that thread priority behavior can be platform-dependent, meaning that the actual scheduling of threads may vary between different operating systems.

Key Concepts of Thread Priorities

1. Priority Levels:

- Java defines a set of priority constants in the Thread class:
- Thread.MIN_PRIORITY (value 1)
- Thread.NORM_PRIORITY (value 5, which is the default)
- Thread.MAX_PRIORITY (value 10)
- These constants represent the minimum, normal, and maximum priority levels that a thread can have.

2. Setting Thread Priority:

- You can set the priority of a thread using the setPriority(int priority) method. The priority value must be between Thread.MIN_PRIORITY and Thread.MAX_PRIORITY.

```
Thread thread = new Thread();

thread.setPriority(Thread.MAX_PRIORITY); // Set to maximum priority
```

3. Getting Thread Priority:

• You can retrieve the current priority of a thread using the getPriority() method.

```
int priority = thread.getPriority();
```

4. Thread Scheduling:

• The Java Virtual Machine (JVM) does not guarantee that higher-priority threads will always run before lower-priority threads. The actual scheduling is determined by the underlying operating system's thread scheduler.

• In practice, higher-priority threads may receive more CPU time than lower-priority threads, but this is not guaranteed.

5. Default Priority:

• When a thread is created, it inherits the priority of its parent thread. If the parent thread is the main thread, the default priority is NORM_PRIORITY (5).

6. Impact of Thread Priority:

• While thread priorities can influence the order of execution, they should not be relied upon for critical application logic. Instead, they are more useful for optimizing performance in scenarios where certain tasks are more important than others.

Example of Setting Thread Priorities

Here’s a simple example demonstrating how to set and get thread priorities:

```
class MyThread extends Thread {
    public void run() {
        System.out.println(Thread.currentThread().getName() + " - Priority: 
" + Thread.currentThread().getPriority());
    }
}

public class ThreadPriorityExample {
    public static void main(String[] args) {
        MyThread thread1 = new MyThread();
        MyThread thread2 = new MyThread();
        MyThread thread3 = new MyThread();

        thread1.setPriority(Thread.MIN_PRIORITY); // Set to minimum 
priority
        thread2.setPriority(Thread.NORM_PRIORITY); // Set to normal 
priority
        thread3.setPriority(Thread.MAX_PRIORITY); // Set to maximum 
priority

        thread1.start();
        thread2.start();
        thread3.start();
    }
}
```