Possible Multi-Threading Interview Questions

Difference between a Thread and a Process?

What is the need for multi-threading?

Difference between Runnable and Thread?

How to implement a Thread in Java Program?

Difference between start () and run ()?

Where do you find call () method?

Difference between wait () and sleep ()?

What is a volatile variable?

How thread creation different from Java and C?

Use of Join () in multi-threading?

Purpose of volatile variable in Multi-threading?

How do you find whether a method or class is thread safe?

How do you achieve thread safety in multi-threading?

In point of Multithreading, which is better to use – String builder or String buffer?

Race condition in multi-threading?

How to stop a thread and continue its execution?

Difference in the working of suspend (), stop (), wait () and notify ()?

Difference between notify () and notify All ()?

How to create a thread in java?

Difference between run () and call ()?

Difference between run () and start ()?

What happens when an Exception occurs in a Thread?

Name some Popular Exception in Threading and how to handle those?

What is the use of an Un-Caught Exception Handler?

Which data structure is used to communicate between threads?

What happens if a function is not thread safe?

Difference between interrupted () and isInterrupted ()?

Can we use wait () and notify () from an A-synchronized block?

Difference between multi-threading and Concurrency?

What is the use of Garbage Collector?

What is a Thread pool?

What is deadlock in multi-threading?

How to check whether a thread is already locked or not?

In which package thread class Is present?

Use of yield () in multi-threading?

Difference in usage of Volatile and Atomic variable in Multi-threading?

What happens if a thread throws and Exception inside a Synchronized block?

Which function is used to force start a thread?

When does Producer and Consumer problem occur in java multi-threading?

What is meant by context switching in Multi-threading?

- **Prepared by:**
GowthamRaj K