1. Define the following thread synchronization approaches:

Thread synchronization is defined as a mechanism which ensures that two or more concurrent processes or threads do not simultaneously execute some particular program segment known as critical section.

Locks

A lock is designed to enforce mutual exclusion concurrency control policy. It is a more flexible and sophisticated thread synchronization mechanism than the standard synchronized block

Mutex

It is a program object that allows multiple program threads to share the same resources, such as file access, but not simultaneously. When a program is started, a mutex is created with a unique name. (rubber chicken example)

Semaphores

It is a thread synchronization construct that can be used either to send signals between threads to avoid missed signals, or to guard a critical section like you would with a lock

Synchronized

When we start two or more threads within a program, there may be a situation when multiple threads try to access the same resource and finally they can produce unforeseen result due to concurrency issues. For example, if multiple threads try to write within a same file then they may corrupt the data because one of the threads can override data or while one thread is opening the same file at the same time another thread might be closing the same file. So there is a need to synchronize the action of multiple threads and make sure that only one thread can access the resource at a given point in time.

Volatile

tells the compiler that the value of a variable must never be cached as its value may change outside of the scope of the program itself. only ensures, that at the moment of access of such a variable, the new value will be immediately visible to all other threads and the order of execution ensures that the code is at the state you would expect it to be.

Atomic

There is a branch of research focused on creating non-blocking algorithms for concurrent environments. These algorithms exploit low-level atomic machine instructions such as compare-and-swap (CAS), to ensure data integrity.

So basically: read; try to store incremented value; if not successful (the value is no longer equal to current), read and try again. The compareAndSet() is implemented in native code (assembly).

2. Define Deadlock conditions.

A deadlock happens in operating system when two or more processes need some resource to complete their execution that is held by the other process.

Four Necessary and Sufficient Conditions for Deadlock

- mutual exclusion: at least one process must be held in a non-sharable mode.
- hold and wait: there must be a process holding one resource and waiting for another.
- No preemption: resources cannot be preempted.
- circular wait: there must exist a set of processes
- [p1, p2, p3, pn] such that p1 is waiting for p2, p2 for p3, and so on p3.

3. Define Race conditions.

Race conditions arise in software when an application depends on the sequence or timing of processes or threads for it to operate properly. As with electronics, there are critical race conditions that result in invalid execution and bugs. Critical race conditions often happen when the processes or threads depend on some shared state. Operations upon shared states are critical sections that must be mutually exclusive. Failure to obey this rule opens up the possibility of corrupting the shared state.

Example:

As a simple example, let us assume that two threads want to increment the value of a global integer variable by one. Ideally, the following sequence of operations would take place:

Thread 1	Thread 2		Integer value
			0
read value		←	0
increase value			0
write back		→	1
	read value	←	1
	increase value		1
	write back	→	2

In the case shown above, the final value is 2, as expected. However, if the two threads run simultaneously without locking or synchronization, the outcome of the operation could be wrong. The alternative sequence of operations below demonstrates this scenario:

Thread 1	Thread 2		Integer value
			0
read value		←	0
	read value	←	0
increase value			0
	increase value		0
write back		→	1
	write back	→	1

In this case, the final value is 1 instead of the expected result of 2. This occurs because here the increment operations are not mutually exclusive. Mutually exclusive operations are those that cannot be interrupted while accessing some resource such as a memory location.

4. What is an ORM?

Object-relational mapping (ORM) is a programming technique in which a metadata descriptor is used to connect object code to a relational database. Object code is written in object-oriented programming (OOP) languages such as Java or C#. ORM converts data between type systems that are unable to coexist within relational databases and OOP languages.

5. What is a memory leak?

In computer science, a memory leak is a type of resource leak that occurs when a computer program incorrectly manages memory allocations in such a way that memory which is no longer needed is not released. A memory leak may also happen when an object is stored in memory but cannot be accessed by the running code. A memory leak has symptoms similar to a number of other problems and generally can only be diagnosed by a programmer with access to the programs' source code.

6. What is an ANR (Application Not Responding) and what are some common causes?

It is an abbreviation that describes an unresponsive Android app. When an app is running on an Android device and stops responding, an "ANR" event is triggered. Two conditions may cause an ANR error on an Android device:

- An active app does not respond to an input event within 5 seconds.
- The BroadcastReceiver class does not finish executing after a long period of time.

reasons

Some are developer-related, such as a poorly written function that loops more times than necessary. Others are device-related, meaning the hardware cannot keep up with the demands of the app. For example, if an app is rendering a large document, it may take several seconds to load the data and render the image on the screen. This could produce an ANR message, though the process might complete a few seconds later.