**Importance of Multithreading concept**

1. **How to create a Thread.**

What are different ways in which thread can be created. There are two different ways in which thread can be created:

1. Our class will extend a **Thread class** and override the run method. within create a Thread class and invoke

a start () method on it. Another way to create a thread.

1. The second way, **implement the** **Runnable interface**, instead of extends the Thread class, because it uses the Capability of extending the any other java class if required, in case of extending a Thread class we cannot extend any other class in java. If required in our other application, because java does not support multi-class level inheritance.
2. **What is the key difference b/w Process v/s Thread**

* The program is execution is called as a Process.
* A Process are Independent.
* Process has different address space in memory.
* A Thread is subset of the process**.**
* However, a Thread contain a shared address space of the process.
* The Interthread communication is faster and easier in the intersperses communication.
* The context switching is also b/w threads when compared to the process

1. **Can we invoke the start method twice?**

No, it is not possible for restart the thread multiple time, means once the thread is started is executed by the thread scheduler it completes the run () method after that thread enters the dead state. If once thread executes then we got a lligalThreadStateException.

1. **Lifecycle of a thread?**

* Each Thread must go through the following states during its lifetime.
* Starting with New Thread, Here the new instance of thread will create, However the Thread is not yet alive.
* The Runnable Thread the Thread becomes Runnable once the start () method is called on the thread, it makes the Thread ready to run, the Thread is not yet running as its not yet selected by the Thread scheduler.
* Running Thread, when the thread scheduler selects the Thread from the Thread Pool the run () method is invoked, this method denotes the thread is running.
* Waiting or blocked, in this state the Thread alive but it is not running, or it might be waiting for the otherThreads to finish.
* Dead or terminated, this state is achieved when the execution of the run () method is complete the Thread is set to be dead or terminated after this.

1. **What is Thread Synchronization?**

* When we have multiple Threads accessing the same object or data, there is a chance that these threads might corrupt each other’s data or objects, this reason Synchronization comes in.
* Once we mark or method using the synchronized keyword the first Thread will enters the method will attain the lock on that method as well as all the synchronized methods in that Thread class, so that no other Thread will be able to access those methods until the first Thread finishes its job.
* We can also have synchronized keyword on static methods and at a block level as well

1. **What is a class level lock?**

Every class in java as a unique lock associated with it, when a Thread executes a synchronized static methodwhich will attain the class level lock all the other Threads will have to wait access this static synchronized method and any other synchronized static methods on that class.

1. **What are synchronized blocks?**

* when we used synchronized keyword on a method entire method will be locked and can used only by one Thread at atime.
* If we know exactly which lines of code within those methods needs locking, instead of locking the entire method we can use a synchronized block instead.
* When we used synchronized block only those few lines of code within the block will be locked, when we do that,we can pass in object or a class which we want to attain the lock as well.

1. **How to threads communicate each other?**

* Threads communicate with using wait () notify () notifyAll () methods
* when the Thread invokes a wait method it gives up the lock on a particular object and another Thread can take over that lock and do its works, once the other Threads can finish the work, it will use the notify () to notify the other Thread which can gave up the lock that it finishes the work, and it can continue doing its job.
* Also use a notifyAll () method there are multiple threads waiting for that lock and gave up the lock for this Thread.
* To use this wait () notify () notifyAll () methods the Threads should be in a synchronized context, weather in asynchronized block or a synchronized method otherwise IIligalMonitorStateException will be thrown.

1. **what does the join () method do?**

* The join () method causes the current running thread to pause its execution until the Thread joins completes its task.There are three different variations in join () method.
* Doesn’t take any parameter. This method will put the current running Thread in wait mode until the Thread on which the method is called dead. In case if the Thread is interrupted then it will throw an interruptedException.
* join (long milliseconds) it takes join the milliseconds this method will wait for the given number of milliseconds instead of waiting for the other Thread to finish wait for certain amount of time the start working again or it will come out of the wait mode.
* join (long milliseconds, int nanoseconds) we can tell the amount of time to wait both in milliseconds andnanoseconds.

1. **What is the use of wait () method?**

The object class as a wait () method which is used for interthread communication, The method is used to pausethe current Thread, and it will wait until, and other Thread calls the notify () or notifyAll () method.

1. **What is a Shutdown Hook?**

The Shutdown Hook is a Thread that is invoked implicitly before the JVM shuts down, one can use it to perform cleaning up of the resources to save the state of the JVM when its shuts down either normally or abruptly.

1. **What is a Deamon Thread?**

A Deamon Thread is low priority Thread, it provides the backgrounds of support and services to the user threadsweather we check using isDeamon () method.