

1. Session 10: Multithreading in JAVA programming.

2. Intended Learning Outcome:

- a. Will learn about thread and Multithreading.
- b. How to use multithreading in Java application.

3. Expected skills:

- Can explain what is multithreading.
- Can differentiate between multitasking and multithreading.
- Can draw the life cycle of thread.
- Know two ways of creating threads in Java.
- Knows the use of common methods like Thread.sleep(), Thread.currentThread().
- Knows the use of .join() and synchronized.

4. Tools Required:

- a. JDK
- b. ECLIPSE / NETBEANS

5. Session Detail:

1. Teacher will give lecture on:
 - Thread, Multithreading, Thread and Multitasking, Life cycle of thread, Application thread (30 minutes)
2. Teacher will show how to create threads in Java: (60 minutes)
 - Using Thread class: Example no. 10.1 or 10.3.
 - Using Runnable interface: Example no. 10.2 or 10.4.
3. Teacher will show the use of:
 - .join() method: Exercise no. 10.6.
 - synchronized. Exercise no. 10.7.

6. Post Lab Exercise:

- a. Max value: Write a program called MaxValue.java that finds the maximum value in an array of ints using 4 threads. Your main should be similar as the one in the above-linked [SumThread](#) example, though you should construct your array of random numbers instead of increasing numbers. You may assume in your threaded code that the array has at least 4 elements.Exercise no.
- b. Shared counter: Write a program called SharedCounter.java in which 10 threads each increment a shared int counter 10 times. When all the threads have finished, print the final value of the counter. If the initial value is zero, do you always get 100? Arrange for your code to sometimes print the wrong answer. (Hint: try using some well-placed calls to Thread.yield() or Thread.sleep().)

7. Further Readings:

- a. <http://beginnersbook.com/2013/03/java-threads/>
- b. https://www.tutorialspoint.com/java/java_multithreading.htm
- c. [http://www.javatpoint.com/sleep\(\)-method](http://www.javatpoint.com/sleep()-method)
- d. <http://www.javatpoint.com/synchronization-in-java>
- e. https://www.tutorialspoint.com/java/lang/thread_join.htm