# **Multi-threading, synchronization in programming using Java: Part 1**

In the part 1 we have discussed about the, mutli threading programming, creation pf thread, execution, implementation as well as prevention of its action. In this part we are going to discuss more about muti threading concepts, synchronization, thread-safety, occurrence and prevention of deadlock and some other features related to thread of execution in Java programming language.

We already are aware that threads allow multiple actions to occur simultaneously. In real world most of the cases requires multiple resources to be used by different entities in same or different time. And all these can or cannot make differences to the resources. But it is utmost important that every time one action is performed and it is over the other ones must get the actual, real value of the resources. Two important features can be described here.

Such as if money is withdrawn from the same Bank account by two person one from the ATM other being from the bank counter, the remaining amount should show the proper balance left. Or if one tires to overdraw the amount an error message should be displayed. The operations should be synchronized.

People can buy tickets for a flight from one destination to the other from various places. A flight has limited number of seats and the number of tickets sold should not be more than the available seats. But before the flight gets filled each person should be allowed to put all the required details and book the flight. If this is done sequentially, and a person takes much longer time, the others after that person will have to wait till the previous operation gets over. This is not at all acceptable. Alsoif a person while booking gets in an emergency and stops there no other person can start booking leading to a deadlock situation.

To overcome the *deadlock* and to let the values be *synchronized* we use multi-threading program.

**What is deadlock:**

**What is race-condition:**

**What is synchronization:**