

1. Create two threads in java. One using thread class and other using Runnable interface.
2. Create two threads, one will print "hello world" every second and other will print "Bye" every 2 seconds.
3. Write a Java program to create 5 threads to print 1-10 in a loop in every thread. The output must look like as displayed below:
<Thread Name> – 1
<Thread Name> - 2 and so on
Each thread will print in the same manner. Check the output sequence. Is it in sequence or random order. Record your findings with the reason in form of comments.
4. In the above question, set the priority of Thread 1 to be highest. See the impact on the output.
5. Create two threads, one will print "hello" and other will print Bye for 10 times. The order of printing the message should be one after another i.e "hello" "bye" "hello" "bye"
6. Write a java program that creates a number of threads and each thread must start after the completion of previous thread except the first one.
7. Write a java program taking array as a shared resource and which is access by multiple threads with and without synchronization it.
8. Write a java program that creates a number of threads which access the static field and prints the incremented value of static field.
9. Write a java program to present a deadlock condition.
10. Write a java program with two threads in which each thread reads the data from a list of Strings and display the data of each list on Console alternatively such that one object from first list is printed and then one object from another list is printed.
11. 3 customers (A, B, C) visit car showroom. The visitors take a test drive for random time and the other visitors have to wait till a car becomes free. i.e if "A" is taking the drive B and C have to wait. Implement this functionality using threads in java
12. Create a collection containing 1000 LoanAgreement objects. Use the LoanAgreement Class created in Java Day– 2 assignment. The objects will have all the information other than the emiPerMonth field value. We need to calculate EMI for all the 1000 objects. Use threading to implement the same. Do the parallel processing to achieve the result faster.