

# Lab Manual 7

## Multi-threaded Programming

1. Write Java Program that implements a multithread application that has three threads. First thread generates random integer for every second and if the value is even, second thread computes the square of number and prints. If the value is odd, the third thread will print the value of cube of number.
2. 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.
3. Write a program called `ReverseHello.java` that creates a thread (let's call it Thread 1). Thread 1 creates another thread (Thread 2); Thread 2 creates Thread 3; and so on, up to Thread 50. Each thread should print "Hello from Thread <num>!", but you should structure your program such that the threads print their greetings in *reverse order*.
4. 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()`.)