# CS6235
# Assignment 1

The assignment expects that students will write their own code. All the Java codes snippets - should be valid Java code that is compilable using a Java compiler. Compile your code, test it and only then use it in the assignment.

1. [10+10] Write Java code snippets to show (i) data-races, (ii) how atomicity violations can exist despite race freedom (using `synchronized` blocks or methods).

2. [10+5] Write a Java program and use it to illustrate Amdahl's law. Show the execution time numbers to empirically establish the law. You may use any multi-core system (with at least 8 cores) for the experiment.

3. [5] Write a Java program to prove that Java threads share the heap.

4. [10+10] For the code written in Q1(i) and Q1(ii): show the static Happens Before (HB) relation between the different Java statements. You would need to add a line number to each line in the Java code to illustrate the HB relation.

   Note: (1) If two statements `S1` and `S2` may run in parallel with each other - they have no HB relation. Else, either `S1` HB `S2` and/or `S2` HB `S1`. (2) While analyzing a program statically, unlike the actual execution, there will be cases where we may say that two statements `S1` and `S2`, may have HB relation with each other in both directions (that is, `S1` HB `S2` and `S2` HB `S1`).

5. [10+10] Write a Java program that leads to a deadlock due to parallelism related constructs: (i) uses threads, and synchronized methods. (ii) uses threads and cyclic-barriers.

6. [20] Write a Java program (using CyclicBarriers) that implements a single consumer, single producer scenario. Assume that the producer produces an item, waits for the consumer to consume. Similarly, the consumer thread waits for an item to be available, and then consumes it. The consumer and producer threads repeat the same tasks in an infinite loop.