**Title: Conquering the Nodes (50%)**

The purpose of this assignment is to assess the students’ ability to apply adequate programming skills and use appropriate constructs in the Java language to solve a concurrent problem.

1. **Assignment Description**

You are required to build a simple game using multithreading. The game (your program) accepts three arguments, *n*, *t* and *m*, where *n* >> *t*. After receiving the arguments, the program will sequentially create *n* random points. The points are floating point (floats or doubles) coordinates in a 1000 x 1000 region. NO two points should overlap. Each point is represented as an object containing the coordinates.

After that, your program will launch *t* threads. Each thread will randomly pick a pair of points to add an edge between them. Use an appropriate data structure to store the pair of connected points. However, NO point should be connected with more than one other point, i.e. each point could be used to form at most one edge (locking needed when forming an edge).

The program will run for *m* seconds, or when any one thread has failed to form a single edge after 20 attempts. Display the number of edges each thread was successfully created.

1. **Tasks and Deliverables**

Form a group of 4-5 members (not confined to your tutorial group) to develop the game. Use an online collaboration tool (Github, Jira, Beanstalk etc.) for development. Create a wiki to document your project and report your progress. The progress will be assessed in Weeks 9 and 12. Each member’s responsibility and contribution must be reported on the wiki. In Week 14, upload your source code, executable jar file, and video (link) to Spectrum.

1. **Assessments**

Interim assessments through wiki (Weeks 9 and 12) – 8 % each (Total = 16%)

* Group/team formation, responsibility and contribution of each member
* Project planning
* Selection of development tool
* Progress of project
* Program design
* Issues encountered and solutions
* Short demo video of the partially done program (< 1 minutes)

Final assessment (Week 14 – due Wednesday 23 June 2020 at 11.59p.m.) (Total = 34%)

Marking scheme

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Criteria** | **Excellent** | **Good** | **Average** | **Poor** | **Very Poor** |
| **Source code 1 (10%)** | | | | | | |
| 1 | Programming style, including code organization, readability, descriptive variables, clear comments, etc. | 4 | 3 | 2 | 1 | 0 |
| 2 | Data structures for efficient data representation and processing | 2 | 1.5 | 1 | 0.5 | 0 |
| 3 | Implement multithreading correctly | 4 | 3 | 2 | 1 | 0 |
| **Executable jar file (8%)** | | | | | | |
| 4 | Error free/handling | 4 | 3 | 2 | 1 | 0 |
| 5 | Visualisation of end results | 2 | 1.5 | 1 | 0.5 | 0 |
| 6 | Visualisation of the process (drawing edges) | 2 | 1.5 | 1 | 0.5 | 0 |
| **Wiki (6%)** | | | | | | |
| 7 | Up-to-date progress report | 2 | 1.5 | 1.0 | 0.5 | 0 |
| 8 | Program design and algorithms | 4 | 3 | 2 | 1 | 0 |
| **Video (10%)** | | | | | | |
| 9 | Clear and true reflection of the program | 4 | 3 | 2 | 1 | 0 |
| 10 | Individual member’s testimony on responsibility, work done, and lessons learnt. | 6 | 4.5 | 3 | 1.5 | 0 |

## Late submission

If for some good reason it is impossible to get the assignment in by the deadline, email the lecturer to seek for a late submission. This should be done prior to the deadline. Late submisisons without prior approval will have marks deducted (2% per day overdue).

## Penalties

If it is found that plagiarism has occurred, one or more of the following penalties may be imposed:

• loss of all or partial marks for the assessment item;

• downgrading the final grade in the course;

• imposing a grade of fail in the course.