

Gebze Technical University
Computer Engineering Department
CSE443 - Object Oriented Analysis and Design
Fall 2019-2020

Homework 2 - v1

Rule 1: no plagiarism (from colleagues or other sources). Detected cases of plagiarism will lead to a significant penalty of your course grade at the end of the semester.

Rule 2: no late submissions! Even if it is late by **one minute**, it will be ignored. Learning to plan your schedule according to deadlines is part of your education and an invaluable professional asset.

What to submit: a) the source code of your project fully documented (with javadoc), b) a nicely formatted pdf report of your design decision explanations and class diagrams and c) an executable demo that fully illustrates your program's capabilities whenever code is requested.

Question 1 (50 points): You should have never lied at your resumé. Why on earth did you add pattern recognition to your lists of skills, when you have never followed such a course? It's done now. Your boss considers you the designated specialist on pattern recognition matters, and comes to you for every single optimization task he encounters. His latest and brightest idea is to solve constrained optimization problems with **genetic algorithms**:

```
Generate the initial population
Compute fitness
REPEAT
    Selection
    Crossover
    Mutation
    Compute fitness
UNTIL population has converged
```

The function to optimize is:

Maximize: $f(x_1, x_2) = 20x_1x_2 + 16x_2 - 2x_1^2 - x_2^2 - (x_1 + x_2)^2$
Subject to: $x_1 + x_2 \leq 5; 0 \leq x_1 \leq 5; 0 \leq x_2 \leq 5$

Now, the tricky part is that your boss wants you to implement 3 versions of this algorithm, and more may be needed in the future:

Version 1 will use "roulette wheel selection" and 1 point crossover.

Version 2 will use "rank selection" and 2 point crossover.

Version 3 will use "tournament selection" and 1 point crossover.

[It's](#) a good thing you took that design pattern course. This seems like an excellent opportunity for using the template method design pattern. Develop the application that solves the above constrained optimized problem using a genetic algorithm and implement all 3 versions using the template method design pattern.

Question 2 (50 points): Prepare a graphical user interface for the algorithms you coded in question 1. Your GUI should have a start/pause/stop button to initiate the genetic algorithm loop. The tricky part is that if the user pushes the start button, all 3 versions of the algorithm must run/pause/stop at the same time. Make sure your interface shows in real time (using a graphical 2d plot: iterations vs fitness) how fast each version of the algorithm is converging to an optimal solution.

Good luck.

