EECS 293
Software Craftsmanship
2016 Fall Semester

Programming Assignment 6

Due at the beginning of your discussion session on

October 10-14, 2016

Reading

Read Sections 9.2, 9.3, 19.6 in Code Complete and the Quick Reference on Routine Names on blackboard.

Grading Guidelines

An automatic C (or less) is triggered by:

- Any routine with complexity greater than 4, or by
- Any piece of code that is essentially repeated.

Starting with Programming Assignment 7, an automatic C (or less) will be triggered by improperly named routines.

Assignment

The two opposing clans of Nobunaga and Hideyoshi are locked in a war for supremacy. They have recently started to hire ninja warriors to launch covert operations against each other, and the shogun is worried. He has asked you to find out more about the ninjas, not an easy task given their methods and secrecy. However, your informants occasionally report that two ninjas belong to different factions. At that point, although you are still in the dark as to whether a ninja fights for Nobunaga or for Hideyoshi, at least you know that they are on different sides. Furthermore, the shogun often asks you if you have any information on whether two warriors are on the same side.

To get a handle on the ongoing war, you will design a data structure that supports the following operations:

CREATE(x): add object x to the set of known objects.

OPPOSE(x, y): add the fact that objects x and y belong to different sides, or report an error if x and y are already known to be on the same side.

OPPONENTS(x, y): return whether x and y belong to different sides, or if this property cannot be established.

Here are some general considerations about this data structure:

- Although the shogun is mostly interested in the case when x and y are of type `Ninja`, the data structure should be generic.
- You should come up with the fastest design for these methods.

Write pseudo-code for the data structure methods. In the class discussion, you could be asked to justify the correctness of your data structure or to analyze its running time. It is to your advantage to write down a justification for the algorithm's correctness, an analysis of its running time, and to sketch a few examples of the algorithm operation.

No implementation is required: you will implement your pseudo-code in the next programming assignments. After this assignment, you will not be allowed to make major changes to the pseudo-code.

Blackboard Resources

The Course Documents page contains a folder (Pseudocode Examples) with a cheat sheet and five examples of pseudo-code from various textbook.

Discussion Guidelines

The class discussion will focus on the pseudo-code for the data structure methods. The pseudo-code must be of sufficiently good quality that you can easily generate code from it in the next programming assignment. You may also be required to walk through your pseudo-code on some examples. It is better to present

correct pseudo-code for a clearly specified subset of requirements rather than incorrect pseudo-code, or to address an unclearly stated subset of requirements.

Submission

Submit an electronic copy of your pseudo-code and any other ancillary documents to Blackboard.