

January 2014

Proposals:

1. To add a unit testing topic to the course objectives and outline.
2. To add sets, lists, and maps to the course objectives and outline.

By: Kip Irvine

Justifications:

Proposal 1: Many of our students apply for internships at companies that include software testing as a required activity. Although undergraduate IT students are not required to take a software testing course, COP 4814 may be a good place for them to at least be exposed to unit testing. We see evidence of unit testing, for example, in Websites such as Stanford University's codingbat.com, designed for first-semester programming students. Unit Testing frameworks are already included in the tools our students currently use, including Visual Studio, Eclipse, and NetBeans. Unit testing has several advantages for beginning programmers: (1) they learn to create functions that have a specific purpose; (2) they learn to independently validate each function in their application; and (3) they learn how continuous testing and test-driven development can enhance software project development.

Proposal 2: After having taught this course for one semester, the most prominent weakness shown by the students is a poor understanding of object-oriented design. The students have had little practice in creating programs that use inheritance, abstract classes, and interfaces. They have almost no knowledge of the Java Collections API, including sets, maps, interfaces, stacks, and queues. In order to teach design patterns such as decorator, facade, factory classes, and composition, I have found the need to review basic OO concepts.