



THE OHIO STATE UNIVERSITY
COLLEGE OF ENGINEERING

Software I: Software Components

CSE 5022

Credit Hours:

3.00

Course Levels:

Graduate (5000-8000 level)

Course Components:

Lecture

Lab

Course Description:

Intellectual foundations of software engineering; design-by-contract principles; mathematical modeling of software functionality; component-based software from client perspective; layered data representation. Previous programming experience in any language required.

Prerequisites and Co-requisites:

Prereq: At least one term of Calculus.

Course Goals / Objectives:

- Reasons it is important that software be "correct", i.e., why "good enough" is not good enough when it comes to software quality.
- Reasons for designing software to minimize the impact of change, and why it is difficult to achieve this.
- Be competent with using design-by-contract principles to write software that uses existing software components based on their interface contracts.
- Be competent with using interface contracts that are described using simple predicate calculus assertions with mathematical integer, string, finite set, and tuple models.
- Be competent with extending existing software components by layering new operations on top of existing operations.
- Be competent with layering new software components' data representations on top of existing software components.
- Be competent with using simple recursion.
- Be competent with using simple techniques to test application software, layered implementations of extensions, and layered data representations, including developing and carrying out simple specification-based test plans.
- Be competent with using simple techniques to debug application software, layered implementations of extensions, and layered data representations.
- Be exposed to using basic algorithm analysis techniques and notations to analyze and express execution times of operations whose implementations involve straight-line code and simple loops.
- Be competent with writing Java programs in a procedural style using the basic control structures, primitive value types, character strings, and input/output.
- Be familiar with writing Java programs using core language features including interfaces, classes, inheritance, and assertions.
- Be familiar with writing Java programs that use software components similar to (but simplified from) those in the Java collections framework.
- Be familiar with using an understanding of the difference between value types and reference types to trace the execution of simple Java code in situations involving both flavors of types, including their use as parameters to method calls.
- Be familiar with testing using JUnit.
- Be familiar with illustrating key dependencies between software components using UML class diagrams (or similar).
- Be familiar with using the most important features of a modern IDE, e.g., Eclipse.

Course Topics:

Designation:

Elective