Software I: Software Components

CSE 2221

Credit Hours:
4.00

Course Coordinator:

Course Length:
14 weeks (autumn or spring)
12 weeks (summer only)

Representative Textbooks and Other Course Materials:

Title	Author	Year
On-line reference materials		

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.

Prerequisites and Co-requisites:
Prereq: 1212, 1221, 1222, 1223, 1224, Engr 1221, 1281.01H, 1281.02H, or CSE Placement Level A. Prereq or concur: Math 1151, 1161.01, or 1161.02.

Designation:
Required
Elective

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

---

ABET-CAC Criterion 3 Outcomes:

Significant contribution (7+ hours)	1	Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
Significant contribution (7+ hours)	2	Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
Some contribution (1-2 hours)	3	Communicate effectively in a variety of professional contexts.
Some contribution (1-2 hours)	4	Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles
Substantial contribution (3-6 hours)	6	Apply computer science theory and software development fundamentals to produce computing-based solutions.

ABET-EAC Criterion 3 Outcomes:

Significant contribution (7+ hours)	1	an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
Significant contribution (7+ hours)	2	an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
Some contribution (1-2 hours)	3	an ability to communicate effectively with a range of audiences - pre-2019 EAC SLO (g)
Some contribution (1-2 hours)	4	an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
Some contribution (1-2 hours)	6	an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions
Some contribution (1-2 hours)	7	an ability to acquire and apply new knowledge as needed, using appropriate learning strategies

Course Topics:

- Introduction to Java; value types; control structures; basic input/output; introduction to Eclipse
- Software components; packages; interfaces; design-by-contract; classes; reference types; methods, calls, and parameter passing; equals and toString methods; Text component; Natural component; introduction to UML class diagrams (or similar)
- Layered implementations of new Text and Natural methods; introduction to recursion; introduction to specification-based testing and JUnit
- Generics; Sequence component; Queue component; Stack component; List component; layered implementations of new Sequence, Queue, Stack, and List methods; more recursion
- Set component; Map component; iterators
- Layered data representation concepts; representation invariants and abstraction functions; Natural representation using a Stack; Sequence/Queue/Stack representation using a List