THE OHIO STATE UNIVERSITY
COLLEGE OF ENGINEERING

Software II: Software Development and Design

CSE 2231

Credit Hours:
4.00

Course Levels:
Undergraduate (1000-5000 level)

Course Components:
Lecture
Lab

Course Description:
Data representation using hashing, search trees, and linked data structures; algorithms for sorting; using trees for language processing; component interface design; best practices in Java.

Prerequisites and Co-requisites:
Prereq: 2221. Concur: 2321.

Course Goals / Objectives:

- Be competent with using design-by-contract principles and related best practices, including separation of abstract state from concrete representation
- Be competent with using interface contracts, representation invariants, and abstraction functions 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 familiar with simple linked data representations, including why and when it is (and is not) appropriate to use them rather than layered data representations
- Be competent with using simple recursion
- Be competent with using simple techniques to test application software, layered implementations of extensions, and layered or linked 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 typical data representations
- Be familiar with using basic algorithm analysis techniques and notations to analyze and express execution times of operations whose implementations involve straight-line code, simple loops, and simple recursion (e.g., in manipulating binary trees)
- Be competent with writing Java programs using core language features including interfaces, classes, inheritance, and assertions
- Be competent with writing Java programs that use software components similar to (but simplified from) those in the Java collections framework
- Be familiar with using many industry-standard "best practices" for Java design and development
- Be familiar with working as part of a team on a software project with multiple milestones
- Be exposed to using a version control system, e.g., CVS or SVN
- Be exposed to the client-server model of distributed computing.

Course Topics:

- Set and Map representations using an array of Queues with hashing
- BinaryTree component; Set representation using a BinaryTree with binary search tree algorithms
- Linked representations of Stack/Queue/List components and variations; singly-linked and doubly-linked lists
- Tree component; language processing using trees; elaboration of small programming language compiler team project (with related programming lab assignments continuing beyond this module); introduction to version control
- Component interface design principles and practices
- Advanced Java language constructs and uses; best practices in Java

---

Designation:

Required
Elective