CSE 2231 Syllabus

CSE 2231: Software II: Software Development and Design

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.

Level and Credits

- U 4 (two one-hour lectures, two one-hour labs)

Prerequisites

- CSE 2221; co-req: CSE 2321

General Information, Exclusions, etc.

- Java is used

Course Goals (Intended Learning Outcomes)

- 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

Texts

- All course materials are provided on-line for free.

Course Topics

- Layered data representation concepts; representation invariants and abstraction functions; NaturalNumber representation using a Stack; Sequence/Queue/Stack representation using a List
- Set and Map representations using an array of Queues with hashing
- BinaryTree components; Set and Map representations using a BinaryTree with binary search tree algorithms
- SortingMachine components; sorting algorithms and their embeddings into SortingMachine implementations
- Tree components; 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

Grading Plan

- **Note: A passing score on the final exam is *required* in order to receive a passing grade for the course.**

Homework Assignments (many)	6%
Project Assignments (several)	30%
Midterm Exams (2 @ 15% each)	30%
Final Exam	30%
Participation	4%

BS CSE Program Outcomes

Course Contribution		Program Outcome
***	a	an ability to apply knowledge of computing, mathematics including discrete mathematics as well as probability and statistics, science, and engineering;
*	b	an ability to design and conduct experiments, as well as to analyze and interpret data;
***	c	an ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints such as memory, runtime efficiency, as well as appropriate constraints related to economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability considerations;
**	d	an ability to function on multi-disciplinary teams;

**	e	an ability to identify, formulate, and solve engineering problems;
	f	an understanding of professional, ethical, legal, security and social issues and responsibilities;
*	g	an ability to communicate effectively with a range of audiences;
	h	an ability to analyze the local and global impact of computing on individuals, organizations, and society;
*	i	a recognition of the need for, and an ability to engage in life-long learning and continuing professional development;
	j	a knowledge of contemporary issues;
***	k	an ability to use the techniques, skills, and modern engineering tools necessary for practice as a CSE professional;
**	l	an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
*	m	an ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
***	n	an ability to apply design and development principles in the construction of software systems of varying complexity.