Software II: Software Development and Design

CSE 2231

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.

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
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

Grades Breakdown:

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework and Class Participation</td>
<td>8%</td>
</tr>
<tr>
<td>Closed Labs</td>
<td>12%</td>
</tr>
<tr>
<td>Programming Lab Assignments</td>
<td>30%</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>20%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>30%</td>
</tr>
</tbody>
</table>

Designation:
Required
Elective

Instruction Modes:
In Person (75-100% campus; 0-24% online)
### Representative Textbooks and Other Course Materials:

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-line reference materials</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>