Software II: Software Development and Design

CSE 2231

Description / Conditions

Transcript Abbreviation:
SW 2: Dev & Dsgn

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 Levels:
Undergraduate (1000-5000 level)

Designation:
Required
Elective

General Education Course:
(N/A)

Cross-Listings:
(N/A)

Course Detail

Credit Hours (Minimum if “Range” selected):
4.00

Max Credit Hours:
(N/A)
Select if Repeatable:
Off

Maximum Repeatable Credits:
(N/A)

Total Completions Allowed:
(N/A)

Allow Multiple Enrollments in Term:
No

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

Off Campus:
Never

Campus Location:
Columbus
Lima
Marion

Instruction Modes:
In Person (75-100% campus; 0-24% online)

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

Electronically Enforced:
No

Exclusions:
Not open to students with credit for 2231.01.

Course Goals and Learning Objectives
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

Check if concurrence sought:
No

Contact Hours
**Contact Hours:**

<table>
<thead>
<tr>
<th>Topic</th>
<th>LEC</th>
<th>REC out-of-class</th>
<th>REC in-class</th>
<th>Weekly LAB out-of-class</th>
<th>Weekly LAB in-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set and Map representations using an array of Queues with hashing</td>
<td>3.0</td>
<td>0.0</td>
<td>0</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>BinaryTree component; Set representation using a BinaryTree with binary search tree algorithms</td>
<td>2.0</td>
<td>0.0</td>
<td>0</td>
<td>2.0</td>
<td>0</td>
</tr>
<tr>
<td>Linked representations of Stack/Queue/List components and variations; singly-linked and doubly-linked lists</td>
<td>4.0</td>
<td>0.0</td>
<td>0</td>
<td>4.0</td>
<td>0</td>
</tr>
<tr>
<td>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</td>
<td>6.0</td>
<td>0.0</td>
<td>0</td>
<td>6.0</td>
<td>0</td>
</tr>
<tr>
<td>Component interface design principles and practices</td>
<td>4.0</td>
<td>0.0</td>
<td>0</td>
<td>4.0</td>
<td>0</td>
</tr>
<tr>
<td>Advanced Java language constructs and uses; best practices in Java</td>
<td>8.0</td>
<td>0.0</td>
<td>0</td>
<td>8.0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>27</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>0</td>
</tr>
</tbody>
</table>

**Grading and Texts**

**Grading Plan:**
Letter Grade

**Course Components:**
Lecture
Lab

**Grade Roster Component:**
Lecture
Credit by Exam (EM):
No

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>

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>

ABET Student Learning Outcomes
## ABET-CAC Criterion 3 Outcomes:

<table>
<thead>
<tr>
<th>Contribution Level</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significant contribution (7+ hours)</td>
<td>1</td>
<td>Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.</td>
</tr>
<tr>
<td>Significant contribution (7+ hours)</td>
<td>2</td>
<td>Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.</td>
</tr>
<tr>
<td>Some contribution (1-2 hours)</td>
<td>3</td>
<td>Communicate effectively in a variety of professional contexts.</td>
</tr>
<tr>
<td>Some contribution (1-2 hours)</td>
<td>4</td>
<td>Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles</td>
</tr>
<tr>
<td>Substantial contribution (3-6 hours)</td>
<td>5</td>
<td>Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline</td>
</tr>
<tr>
<td>Substantial contribution (3-6 hours)</td>
<td>6</td>
<td>Apply computer science theory and software development fundamentals to produce computing-based solutions.</td>
</tr>
</tbody>
</table>

## ABET-ETAC Criterion 3 Outcomes:

(N/A)
## ABET-EAC Criterion 3 Outcomes:

<table>
<thead>
<tr>
<th>Contribution</th>
<th>Hours</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significant contribution</td>
<td>7+</td>
<td>an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics</td>
</tr>
<tr>
<td>Significant contribution</td>
<td>7+</td>
<td>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</td>
</tr>
<tr>
<td>Some contribution</td>
<td>1-2</td>
<td>an ability to communicate effectively with a range of audiences - pre-2019 EAC SLO (g)</td>
</tr>
<tr>
<td>Some contribution</td>
<td>1-2</td>
<td>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</td>
</tr>
<tr>
<td>Substantial contribution</td>
<td>3-6</td>
<td>an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives</td>
</tr>
<tr>
<td>Some contribution</td>
<td>1-2</td>
<td>an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions</td>
</tr>
<tr>
<td>Some contribution</td>
<td>1-2</td>
<td>an ability to acquire and apply new knowledge as needed, using appropriate learning strategies</td>
</tr>
</tbody>
</table>

## Embedded Literacies (UG courses only)

## Embedded Literacies Info:

## Attachments / Additional Notes or Comments

### Attachments:

(N/A)

### Additional Notes or Comments:

(N/A)