COMPSCI 225: Introduction to Computer Science II
4-credit course with 4 lecture hours per week
Course Coordinator: Tim Andersen

Textbook(s) and Supplemental Material

Catalog Description
Recursion. Introduction to program correctness, testing and analysis of time and space requirements. Object-oriented programming and design, including hierarchies and inheritance. Abstract data types – both basic (lists, trees and sets) and derived (queues, stacks, priority queues, and dictionaries) – and their implementation and applications. Concrete data structures (linked lists, binary search trees, heaps, hash tables, etc.) Sorting and searching algorithms.

PREREQ: COMPSCI 125.

Required

Goals for the Course
Successful students will be expected to:

• determine basic space/run-time requirements of algorithms and code fragments
• evaluate tradeoffs in algorithm selection for a variety of problems
• intelligently discuss the abstractions used for stacks, lists, queues, sets trees, dictionaries, and graphs, and the different implementations for each, and be able to encode and use these data structures in Java
• analyze and decompose complex problems in terms of algorithmic and data structure design
• use a modern Integrated Development Environment effectively

Outcomes Addressed

a. an ability to apply knowledge of computing and mathematics appropriate to the discipline
b. an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
c. an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
i. an ability to use current techniques, skills, and tools necessary for computing practice
j. 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 trade-offs involved in design choices
k. an ability to apply design and development principles in the construction of software systems of varying complexity

Outcomes Assessed: i

Topics Covered
Object Oriented Programming and Class Hierarchies
Lists and Collections
Stacks
Queues
Recursion
Trees
Binary Search Trees
Heaps
Sets and Maps
Sorting
Graphs

Grading
Letter grades are assigned to students based on numerical scores for the following activities:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>10%</td>
</tr>
<tr>
<td>Programs</td>
<td>40%</td>
</tr>
<tr>
<td>2 Midterm Exams</td>
<td>30%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>20%</td>
</tr>
</tbody>
</table>

Curriculum Category Content (Credits)

<table>
<thead>
<tr>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Software Design</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Computer Architecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Structures</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Programming Languages</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>