COMPSCI 472: Object-Oriented Design Patterns
3-credit course with 3 lecture hours per week
Course Coordinator: James Buffenbarger

Textbook(s) and Supplemental Material
*Design Patterns*, Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, 1995.

Catalog Description
Reviews object-oriented design principles, explains the goals and form of design patterns, and examines several well-known patterns.

PREREQ: COMPSCI 342.

Elective

Goals for the Course
Successful students will be expected to:

- understand the meaning and benefits of software reusability
- know the form of a software design pattern
- understand domain-independent versus domain-specific patterns
- understand how to apply a pattern
- identify relationships between patterns
- recognize a taxonomy of several well-known object-oriented design patterns
- understand class patterns versus object patterns
- recognize the value of programming to an interface
- understand class versus interface inheritance
- understand inheritance versus composition
- understand delegation
- understand inheritance versus parameterized types
- distinguish between run-time versus compile-time structures
- perform object-oriented design and programming in C++ and Java

Outcomes Addressed
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
k. an ability to apply design and development principles in the construction of software systems of varying complexity

Outcomes Assessed: not applicable

Topics Covered
- Introduction
- Object-Oriented Design and Programming in C++ and Java
- Programming to an Interface
- Polymorphism
- Inheritance
- Class versus Interface Inheritance
- Implementations
Run-Time versus Compile-Time Structures
Reusability
Inheritance versus Composition
Delegation
Inheritance versus Generics
Attributes of a Pattern
Case Study: Lexi, a WYSIWYG Document Editor
A Taxonomy of Several Well-Known Object-Oriented Design Patterns
Selected Advanced Topics

Grading
At the end of the course, a letter grade is assigned to each student according to rank among classmates, which is determined from numerical scores assigned for performance of these activities:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>40%</td>
</tr>
<tr>
<td>Exam</td>
<td>25%</td>
</tr>
<tr>
<td>Final</td>
<td>35%</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></td>
<td></td>
</tr>
<tr>
<td>Software Design</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Computer Architecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Structures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programming Languages</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>