Catalog description:
This course extends the treatment of object-oriented methodologies, languages and tools begin in CSC201J. The emphasis is on the analysis of complex problems, particularly those involving multiple design alternatives, and the use of class libraries. Specific topics include inheritance, polymorphism, recursion, stream and file I/O, exceptions, and graphical interface programming. Style, documentation, solution robustness, and conformance with specifications are emphasized throughout. Three lecture hours and three hours of scheduled laboratory per week, plus extensive programming work outside of class.

Prerequisites: CSC201J with a grade of C+ or higher.

Goals:
The purpose of this course is to enhance and extend students= understanding of tools and techniques for object-oriented software development. Upon completion of the course, a student should be able to do the following:

CG01: analyze a problem statement for completeness and clarity;
CG02: use the methodology of object-oriented design to develop class diagrams (data descriptions and methods) for a problem solution;
CG03: convert this solution into source code in the designated high-level programming language in accordance with a well-defined set of style rules;
CG04: debug and test the program; and
CG05: provide clear documentation for the result.

Objectives:
CO01: Students will gain a deeper understanding of object-oriented design methodology.
CO02: Students will learn to recognize situations in which multiple design alternatives are possible.
CO03: Students will learn to recognize and apply design patterns
CO04: Students will gain experience in judging the effectiveness and cost of a software design.
CO05: Students will gain experience in choosing among competing design alternatives.
CO06: Students will gain experience in the use of the UML modeling language.
CO07: Students will extend their knowledge of an object-oriented programming language, including graphical user interfaces, event-driven programs, file-based input/output, and the use of libraries.
CO08: Students will produce full documentation for several completed projects, including formal class diagrams

<table>
<thead>
<tr>
<th></th>
<th>PO1</th>
<th>PO2</th>
<th>PO3</th>
<th>PO4</th>
<th>PO5</th>
<th>PO6</th>
<th>PO7</th>
<th>PO8</th>
<th>PO9</th>
<th>PO10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO01</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO02</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO03</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CO04</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO05</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO06</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO07</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO08</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Topics (using Java and UML):

- review of:
  - basic design concepts
  - Java syntax
  - UML
  - the concept of incremental development

PF1(1), PL4(1), PL5(1), SE3(0.5)
• designing for reuse  
• discovering and applying design patterns  
• subclasses and inheritance  
• file-based input and output  
• exceptions and exception handling  
• polymorphism  
• data organization and retrieval  
  ° sorting algorithms  
  ° searching algorithms  
  ° performance analysis  
  ° testing and validation  
• interfaces  
• recursion  
• survey of class libraries  
• programmer-developed windows and frames  
• applets  
• multidimensional arrays

**Programming assignments:** Approximately six programming assignments are given. One or more of these may be group projects. Each programming assignment involves the design, writing, testing and debugging of a program and the submission of an appropriate laboratory report. Each assignment has a specific due date, with a short grace period during which the assignment may be submitted for reduced credit. When the grace period has expired, the assignment will no longer be accepted. All programs must be coded in the programming language currently used for instruction in the CSC201J/202J sequence - no exceptions will be allowed. The version of the language being used will be the currently accepted standard version: any extensions or variations in student-owned compilers must be approved in advance by the instructor, who may choose to forbid their use.

**Laboratory exercises:** There will be short programming exercises to be completed during weekly scheduled laboratory sessions. Each exercise focuses on a specific language feature or programming technique presented in recent lectures. Performance on these exercises will be incorporated into the course grade.

**Exams and quizzes:** There will be a midterm examination and a comprehensive written two-hour final examination.

Final grades will be determined on the basis of the following approximate weights: examinations - 40%, programming assignments and lab exercises - 60%.

<table>
<thead>
<tr>
<th>CO01</th>
<th>CO02</th>
<th>CO03</th>
<th>CO04</th>
<th>CO05</th>
<th>CO06</th>
<th>CO07</th>
<th>CO08</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Test / Quiz Questions**  
**Homework Problems**  
**Programming Projects**  
**Lab Exercises**

**Bibliography:**  
Adams, Joel; Nyhoff, Larry; Nyhoff, Jeffrey.  

Bloch, Joshua.  