Syracuse University
L.C. Smith College of Engineering and Computer Science

CIS 454 – Software Implementation

Catalog Description
Ethics in software development. Detailed design of software, using tools such as structure charts. Implementation and software testing. Team projects and presentations.

Course Objectives
Working in teams, students should carry out the detailed design and implementation of a substantial piece of software, meeting (insofar as possible) the requirements set forth in a previously generated requirements specification. The resulting software should be well organized, well documented and thoroughly tested. Students will present progress reports to the class during the semester, coordinate their work with other teams, and present an overview of the work they have accomplished at the end of the semester.

Prerequisites
CIS 453: Software Specification and Design An understanding of requirements documents, software architectures and high-level software designs as represented by CIS 453 (Software Specification and Design). Programming, including object-oriented programming, algorithms and data structures as covered by CIS 351 and its prerequisites. (CIS 351 is a prerequisite for CIS 453).

Course Outcomes
After completion of the course, students should be able to:

- Develop specifications, design and documents [ABET (f)].
- Understand importance of clear interface specifications for software modules [ABET(b, i, k)].
- Appreciate the need for clear, well-organized software which can be understood and modified when necessary (whether written by the original implementors or by others) [ABET(c, i, j, k)].
- Be reasonably adept at working in development teams [ABET(d, f)].
- Be able to give a coherent explanation of the purpose, design implementation, and characteristics of a piece of software [ABET(e, f)].
- Recognize the need to learn new tools and methodologies in the evolving technological marketplace [ABET (h)].
- Demonstrate and document self-directed learning of available tools, research appropriate implementation methodologies and software engineering standards, towards completion of a student-driven project. [ABET(h)].
- Develop and document plans for potential enhancement of course projects [ABET(h)].

Outcome Measurement
Students spend one third of the time in a lab setting, with the full class, and the remaining two thirds of the time in small groups closely supervised by the instructor and the TA. Most of the written work for the course consists of programs and program documentation, with some report to check the correspondence between software and requirements. All students participate in preparing and delivering oral presentations to the class in the course of the semester and a final presentation at the end, but not every student makes an individual presentation.
Course Topics

Program modularity, interface specifications, documentation

- The importance of correct software
- Dealing with unanticipated inputs or responses, failures
- User interface design
- Appropriate topics in programming languages, algorithms, etc. as the need arises

CAC Category Content

<table>
<thead>
<tr>
<th>Credit</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>Data Structures</td>
</tr>
<tr>
<td>0.25</td>
<td>Algorithms</td>
</tr>
<tr>
<td>2.25</td>
<td>Software Design</td>
</tr>
<tr>
<td></td>
<td>Computer Organization &amp; Architecture</td>
</tr>
<tr>
<td>1.25</td>
<td>Programming Languages</td>
</tr>
</tbody>
</table>