Department: Computer Science

Assessment Coordinator: David Bover

Departmental Mission:
The mission of the Computer Science Department is to provide the highest quality education in computer science; to perform research that advances the state-of-the-art in computer science; to produce graduates that are knowledgeable, articulate, principled, innovative, confident, and able to think critically; to be engaged in local, state, and national issues to the benefit of both the public and private sector; and to maintain a diverse college community.

Departmental Student Learning Outcomes: Upon graduation, majors will be able to demonstrate:

1. An ability to analyze, design, implement and test significant computer-based projects, working individually and in small project teams.
2. Proficiency in the selection and use of data structures and algorithms, including an understanding of the underlying mathematical abstractions and the complexity and implementation of solutions.
3. Proficiency in the use of several programming languages and an understanding of different programming paradigms and their implementation in modern programming languages.
4. An understanding of theoretical computer science, software engineering and computer organization concepts and the ability to apply that knowledge in formulating and implementing computer-based solutions to problems.
5. An understanding of the concepts of computer operating systems and concurrent systems.
6. An ability to persuasively express their ideas orally and in writing.
7. An understanding of the social and ethical issues of computing and preparation to use their computing skills to make a positive impact on society.
**Student Learning Outcomes Assessed This Year:**

<table>
<thead>
<tr>
<th>Assessment Measures</th>
<th>SLO's Assessed</th>
<th>Use of the Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation of student performance in analysis, design and implementation of systems</td>
<td>1</td>
<td>Students were found to be weak in object-oriented design. It was felt that this is due to the lack of experience in OO design from previous courses.</td>
</tr>
<tr>
<td>for projects in CSCI 491, 492, 493</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Student programming ability in C-based languages used in CSCI 352, 367, 460</td>
<td>3</td>
<td>Students are not well prepared for significant programming projects in the C language encountered in senior courses.</td>
</tr>
<tr>
<td>Evaluation of student ability to apply theoretical computer science concepts from</td>
<td>4</td>
<td>There appears to be a disparity between students’ knowledge and understanding of theoretical computer science concepts and the application of those concepts to computer-based solutions.</td>
</tr>
<tr>
<td>CSCI 211 and 401.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluation of student understanding of computer organization concepts via Major Field</td>
<td>4</td>
<td>Results from the Major Field Test consistently suggest that our students do not have the same level of understanding of computer organization as in the other MFT categories.</td>
</tr>
<tr>
<td>Test.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Explicit questions on ethical issues in final exams for CSCI 491, 492</td>
<td>7</td>
<td>Students have a good understanding of the social and ethical issues associated with their project work.</td>
</tr>
</tbody>
</table>

**Program Changes Based on Assessment**

1. The required course CSCI 245 “Object Oriented programming in C++” is to be replaced by the new required course CSCI 345 “Object-Oriented Design”. It was felt that too much time was being spent in CSCI 245 on the C and C++
programming languages, leaving little time for discussion of OO design. The new course, CSCI 345, will concentrate wholly on design, without dependence on programming language.

2. The required course CSCI 227 “Computer Organization I” is to be replaced by the new required course CSCI 247 “Computer Systems I”, based on the textbook “Computer Systems: A Programmer’s Perspective” by Bryant and O’Hallaron. This approach will provide an earlier introduction of the C language and enable concentration on the programming aspects of computer hardware. It is expected that this approach will provide students with better preparation for the required courses CSCI 352, 367.

3. The required courses CSCI 211 “Discrete Structures and Functional Programming I” and CSCI 401 “Formal languages and Automata” are to be replaced by the new required course CSCI 301 “Formal Languages and Functional Programming”. The new course will introduce the discrete math concepts more in the context of their application and will include a weekly lab on functional programming applications of concepts.