**Minnesota State University, Mankato**

**Curriculum Proposal**

Please type or select the requested information. Print completed forms, add appropriate paper attachments, and route through MSU’s curricular process for recommendations and decisions.

| Proposal # | 311 |
| Effective Date of Change: | 06-07 |
| (For Office Use Only) | |

<table>
<thead>
<tr>
<th>Course Designator and Number</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 380</td>
<td>3</td>
</tr>
</tbody>
</table>

Include a course or program description for the Bulletin (30-40 words maximum for courses, 100 for programs):

This course focuses on machine level I/O and operating system file processing. Structure of systems programs including assemblers, linkers, and object oriented utilities and interfaces. Students will gain experience in writing utility programs and extensions to an operating system.

Pre: CS 111 or EE 107, and CS 320 F

**Rationale or Justification for change:**

This is part of the CS program redesign.

---

### ***For General Education or Cultural Diversity Courses Only***

<table>
<thead>
<tr>
<th>GE Category #</th>
<th>GE Category Name</th>
<th>(Maximum of 3 Categories)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

? For Writing Intensive Courses, attach a description of the kind and quantity of writing.

? For Upper Division Courses, include a description of the respects in which it is broad and general rather than narrow and specific, and so suitable as GE.

Attach paper copies of the following:

a. Syllabus or course outline.

b. Course's student learning outcomes associated with each GE competency or CD designation.

c. List of strategies to be used to assess students' achievement of each GE competency or CD designation.

---

### ***For New Courses***

<table>
<thead>
<tr>
<th>Instructional Type</th>
<th>Course will be offered:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>Fall Semester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grading Format</th>
<th>Pre- or Co-requisites:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grade</td>
<td>Prerequisites: CS 111 or EE 107, and CS 320</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other courses are being changed or eliminated.</th>
<th>(Explain.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course content or title is similar to courses in other departments.</th>
<th>(Attach copy of letter of agreement with other program(s) contacted. Indicate the nature of the discussions and/or resolution of differences or potential conflicts.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Attach paper copies of the following:

a. Syllabus or course outline.

b. Course's student learning outcomes.

c. A list of resources required to offer and support this course.

d. A description of how teaching this course will affect department staffing.

e. If 400/500 level course, an explanation of added expectations of graduate students.

---

Revised September 2002
***Signature Page***

**Department**
- [ ] Recommended (Category/ies________)
- [ ] Not Recommended (Category/ies________)

Comments:

_Signed_ David N. Nelles 10/17/06

Department Chair

**College Curriculum Committee**
- [ ] Recommended (Category/ies________)
- [ ] Not Recommended (Category/ies________)

Comments:

_Signed_ Karen C. Chen 11/2/06

Committee Chair

**College Dean**
- [ ] Recommended (Category/ies________)
- [ ] Not Recommended (Category/ies________)

Comments:

_Signed_ [signature] 11/6/06

Dean

**General Education Subcommittee**
- [ ] Recommended (Category/ies________)
- [ ] Not Recommended (Category/ies________)

Comments:

_Signed_ [signature] 8/1/07

General Education Subcommittee Chair

**Undergraduate Curriculum and Academic Policy Committee**
- [ ] Recommended (Category/ies________)
- [ ] Not Recommended (Category/ies________)

Comments:

_Signed_ [signature] 2/9/07

UCAP Faculty Chair

**Faculty Association Graduate Committee**
- [ ] Recommended
- [ ] Not Recommended

Comments:

_Signed_ [signature] 2/13/07

Faculty Association Graduate Chair

**Graduate Dean**
- [ ] Recommended
- [ ] Not Recommended

Comments:

_Signed_ [signature] 2/13/07

Graduate Dean

**Academic Affairs Council**
- [ ] Recommended (Category/ies________)
- [ ] Not Recommended (Category/ies________)

Comments:

_Signed_ [signature] 2/13/07

Assistant Vice President

**Senior Vice President and Vice President for Academic Affairs**
- [ ] Approved (Category/ies________)
- [ ] Not Approved (Category/ies________)

Comments:

_Signed_ [signature] 2/13/07

Sr. Vice President, Vice Pres. Academic Affairs
CS 360: Systems Programming (3 credits)

Course Description:
Machine level I/O and operating system file processing. Structure of systems programs including assemblers, linkers, and object oriented utilities and interfaces. Writing utility programs and extensions to an operating system.

3 lecture hours per week.

Prerequisites: CS 111 or EE 107, and CS 320

Proposed Text:

Schedule of Topics:
1) C Programming for System Calls (~1 wks)
2) File System (~1-2 wks)
3) Processes (~1-1.5 wks)
4) Fork and Exec (~1-1.5 wks)
5) The Shell and Signals (~1-2 wks)
6) Interprocess Communication (~1-1.5 wks)
7) Pipes (~1-1.5 wks)
8) Record Locking and Semaphores (~1-2 wks)
9) Message Passing and Shared Memory (~1-2 wks)

Student Outcomes.
Students who complete this course will be able to:
1) Understand the process of writing system software, from code to compilation and linking.
2) Understand the internal operation of Unix/Windows system software including assemblers, loaders, macro-processors, command language interpreters, and inter-process communication.
3) Develop medium to large C/C++ programs utilizing the C preprocessor, the debugger (gdb), make, source code revision systems (scs), etc.
4) Understand the representation of a program (written in C) in a system and accessing/manipulating storage information. L1
5) Understand the different classes of programming exceptions.
6) Understand different methods and the purpose of measuring the execution time of a program.
7) Understand the use of concurrent programming and the three main approaches.
8) Design and implement a substantial hardware / software systems project.
   Students can apply modern debuggers and programming techniques in the development of hardware/software systems

Grades will be assigned based on exams and assignments.

Required Resources & Departmental Staffing:
Resources currently in place within the department, the college, and the university library will support this new course. No new resources are required.

There is no impact on staffing requirements.