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.

(Check all that apply):
- College: Science, Engineering and Technology
- Department: Computer Science
- Program: Computer Science
- CIP #: 11.010104

Proposal #: 201
Effective Date of Change: 06-07
(For Office Use Only)

Type of Change: COURSE PROPOSALS
Proposed: New Course

Title Current: Computer Science II
Title Proposed: Computer Science II

24-Char. Abbrev: CS 111

Number of Credits: 4

Course Designator and Number:

(include a course or program description for the Bulletin (30-40 words maximum for courses, 100 for programs):
Continues the exploration of introductory Computer Science begun in CS 110. Focus is on developing basic knowledge of algorithms, programming skills and problem solving techniques. Topics include recursion, sorting, linked lists, stacks and queues.
Pre: MATH 115 or MATH 113, and CS 110
Fall, Spring

Rationale or Justification for change:
This is part of the CS program redesign and replaces COMS 212 for CS majors.

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

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

1 For Writing Intensive Courses, attach a description of the kind and quantity of writing.
2 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 ***

(Check all that apply):
- Course is an elective.
- Course is required for program.
- Pre- or Co-requisites
- Other courses are being changed or eliminated. (Explain.)

Grading Format:
- Grade
- P/N

Course will be offered:
- Fall Semester
- Spring Semester
- Summer Session

- Course content or title is similar to courses in other departments. (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.)

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
Minnesota State University, Mankato
Curriculum Proposal

***Signature Page***

<table>
<thead>
<tr>
<th>Department</th>
<th>Recommended (Category/ies)</th>
<th>Not Recommended (Category/ies)</th>
<th>Comments:</th>
<th>Department Chair</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Daniel A. Silva</td>
<td>10/17/06</td>
</tr>
</tbody>
</table>

College Curriculum Committee

<table>
<thead>
<tr>
<th>Recommended (Category/ies)</th>
<th>Not Recommended (Category/ies)</th>
<th>Comments:</th>
<th>Committee Chair</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Karen C. Chen</td>
<td>11/2/06</td>
</tr>
</tbody>
</table>

College Dean

<table>
<thead>
<tr>
<th>Recommended (Category/ies)</th>
<th>Not Recommended (Category/ies)</th>
<th>Comments:</th>
<th>Dean</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1/6/06</td>
</tr>
</tbody>
</table>

General Education Subcommittee

<table>
<thead>
<tr>
<th>Recommended (Category/ies)</th>
<th>Not Recommended (Category/ies)</th>
<th>Comments:</th>
<th>General Education Subcommittee Chair</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Undergraduate Curriculum and Academic Policy Committee

<table>
<thead>
<tr>
<th>Recommended (Category/ies)</th>
<th>Not Recommended (Category/ies)</th>
<th>Comments:</th>
<th>UCAP Faculty Chair</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1-21-07</td>
</tr>
</tbody>
</table>

Faculty Association Graduate Committee

<table>
<thead>
<tr>
<th>Recommended</th>
<th>Not Recommended</th>
<th>Comments:</th>
<th>Faculty Association Graduate Chair</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Graduate Dean

<table>
<thead>
<tr>
<th>Recommended</th>
<th>Not Recommended</th>
<th>Comments:</th>
<th>Graduate Dean</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Academic Affairs Council

<table>
<thead>
<tr>
<th>Recommended (Category/ies)</th>
<th>Not Recommended (Category/ies)</th>
<th>Comments:</th>
<th>Assistant Vice President</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3/9/07</td>
</tr>
</tbody>
</table>

Senior Vice President and Vice President for Academic Affairs

<table>
<thead>
<tr>
<th>Approved (Category/ies)</th>
<th>Not Approved (Category/ies)</th>
<th>Comments:</th>
<th>Sr. Vice President/Vice Pres. Academic Affairs</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2/13/07</td>
</tr>
</tbody>
</table>

Revised September 2002
CS 111: Computer Science II (4 credits)

Course Description:
Continues the exploration of introductory computer science begun in CS 110. Focus is on developing basic knowledge of algorithms, programming skills and problem solving techniques.

4 lecture hours per week.

Prerequisites: CS 110

Proposed Text:


Schedule of Topics:
1) Review of C++ fundamentals and big-O notation (~1 wk)
2) Recursion: counting, organizing data, array search, recursion and efficiency (~1-2 wks)
3) Sorting: mergesort(s), quicksort and friends, binary search (~1.5 wks)
4) More recursion: backtracking, simple recursive descent parsing, recursion and induction (~1.5-2 wks)
5) Data abstraction, adts and implementation (~1 wk)
6) Linked list adt: basics, programming with linked lists, implementations subclasses and inheritance. (~2 wks)
7) Iterators (~0.5 wks)
8) Stack adt: basics, programming with, implementations (including as an interface to list) (~1.5 wks)
9) Stacks and recursion (~0.5 wk)
10) Queue adt: basics, programming with, implementations (including list interface) (~1-1.5 wk)
11) Additional topics as time permits: tree basics, event-driven programming fundamentals, etc.

Student Outcomes:
Students who complete this course will be able to:
1) Use recursion to solve simple problems and implement those solutions.
2) Trace the execution of and argue the correctness of a recursive algorithm.
3) Describe the relationship between recursion and mathematical induction.
4) Describe and implement fast (n\log n) sorting algorithms.
5) Describe and implement fast (\log n) array searching.
6) Correctly determine the big-O category of simple iterative and recursive algorithms
7) Understand data abstraction including encapsulation and data-hiding and how to implement these things.
8) Understand classes, class hierarchies, inheritance, polymorphism and how these things are accomplished in C++.
9) Write code that implements and/or uses lists and their ilk; including stacks and queues.
10) Conduct themselves with integrity. They will be able to plan and manage their time. They will be able to understand descriptions of problems and concepts and will understand and conduct logical reasoning about a problem and its solution.

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.