Prefix, Number and Name of Course: MAT 366 Computer Mathematics II

Credit Hours: 3
In Class Instructional Hours: 3 Labs: 0 Field Work: 0

Catalog Description:
Prerequisites MAT 164, MAT 241 and MAT 270 or instructor’s permission.
Structured programming, verification of program validity, data structures, combinatorial problems, flow network algorithms, random number generators, simulation of random and nonrandom processes.

Reasons for Revision:
This is a revision of MAT 366. The revision is done in light of the Committee on Undergraduate Programs in Mathematics guidelines, and was necessary to update the literature and language used to describe the course, as well as to fit better with MAT 241 (Computer Mathematics I). This course will
1. provide mathematicians and scientists with the necessary knowledge to use computers effectively in their fields
2. provide an in-depth introduction to computer organization for mathematicians and scientists
3. provide an overview of computer languages and the mathematics behind them
4. provide further mathematics applications to computer science.

Student Learning Outcomes:

<table>
<thead>
<tr>
<th>Course Content References:</th>
<th>Assessment:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Students will be able to write computer programs to do mathematical modeling</td>
<td>1. Group work in class, individual homework assignments, projects, exams</td>
</tr>
<tr>
<td>2. Students will be able to describe computer programming organization and practices</td>
<td>2. Group work in class, individual homework assignments, projects, exams</td>
</tr>
<tr>
<td>3. Students will be able to determine the feasibility of solving a problem computationally</td>
<td>3. Group work in class, individual homework assignments, projects, exams</td>
</tr>
<tr>
<td>4. Students will be able to write and analyze algorithms.</td>
<td>4. Group work in class, individual homework assignments, projects, exams</td>
</tr>
<tr>
<td>5. Students will be able to use recursive functions effectively to create computational mathematical models</td>
<td>5. Group work in class, individual homework assignments, projects, exams</td>
</tr>
</tbody>
</table>
Course Content:

I. Elements of Program Design
   A. Overview of Programming Languages
   B. Influences on Programming Language Design
      1. Mathematical influences
      2. Other influences
   C. Procedural and Declarative languages
   D. Overview of programming paradigms
      1. Structured programming
      2. Object Oriented programming
   E. Program documentation
   F. Informal Semantics of programming languages
      1. Pseudo-code
      2. The contour model

II. Data structures with mathematical applications
   A. Arrays, files and sets
   B. Linked lists
   C. Stacks and queues
   D. Binary trees
   E. Heaps
   F. Graphs, digraphs, networks and their implementation

III. Recursion
   A. The substitution method
   B. The recursion-tree method
   C. The master method
   D. Proofs of methods
   E. Mathematical induction and recursion

IV. Sorting and searching algorithms
   A. Heapsort
   B. Quicksort
   C. Sorting in linear time
   D. Divide and conquer: Mergesort
   E. Bitonic sort

V. Algorithm analysis and problem complexity
   A. Probabilistic analysis and randomized algorithms
   B. Growth of functions: Asymptotic notation
   C. Algorithm complexity
   D. Problem complexity
   E. Introduction to np-completeness

Resources:

Classic Scholarship in the Field:

Cipra, B. (1993). What’s Happening in the Mathematical Sciences (V1). AMS.
Cipra, B. (1994). What’s Happening in the Mathematical Sciences (V2). AMS.
Current Scholarship in the Field:


Periodicals:

College Mathematics Journal
Discrete Mathematics and Computer Science Journal
Mathematics Magazine
The American Mathematical Monthly
The Journal of Undergraduate Mathematics and Its Applications (COMAP)

Electronic and/or Audiovisual Resources:

The On-Line Encyclopedia of Integer Sequences
(http://www.research.att.com/~njas/sequences/)
The electronic book A=B by Marko Petkovsek, Herbert Wilf and Doron Zeilberger
(http://www.cis.upenn.edu/~wilf/AeqB.html)
IEEE Annals of the History of Computing
(http://www.computer.org/portal/site/annals/)
The History of Computing Projects
(http://www.thocp.net/)
Decode Systems
(http://www.decodesystems.com/)
C/C++ Tutorial
(http://cplus.about.com/od/beginnerctutorial/l/cplusplus.html)
Online C++ Tutorial
(http://www.intap.net/~drw/cpp/0)