CSC 202
Computer Science II

Desired new catalog entry:
CSC 202 Computer Science II (4 cr.)
Examines fundamental data structures and analyzes algorithms. Covers abstract data types and essential data structures such as arrays, stacks, queues, linked lists, and trees; introduces searching and sorting algorithms and algorithm analysis. Prerequisites are CSC 201, Computer Science I, and MTH 173, Calculus with Analytic Geometry I; co requisite is MTH 174, Calculus with Analytic Geometry II. Lecture 4 hours per week.

1. Catalog Entry

CSC 202 Computer Science II (4 cr.)

Examines data structures and analyzes algorithms. Covers data structures (including sets, strings, stacks, queues, arrays, records, files, linked lists, and trees), abstract data types, algorithm analysis (including searching and sorting methods), and objects. Prerequisites are CSC 201, Computer Science I, and MTH 173, Calculus with Analytic Geometry I; corequisite is MTH 174, Calculus with Analytic Geometry II. Lecture 4 hours per week.

2. Entry-Level Competencies

Prerequsites are CSC 201, Computer Science I, and MTH 173, Calculus with Analytic Geometry I. Corequsite is MTH 174, Calculus with Analytic Geometry II

General Course Purpose

This course is the third course for the student majoring in Computer Science. There are four major purposes of the course. Here we begin a detailed study of data structures and data abstraction. We continue algorithmic analysis by examining various searching and sorting algorithms. Students study object-oriented programming in greater depth. .Finally, we continue the development of discipline in program design, in style and expression, and in testing and debugging. The student writes programs larger than those in the previous course. A high-level computer language is used for the students to implement their solutions on a computer.

3. Major Topics to be Included

I. Data abstraction and object-oriented programming (review)
   A. dynamic allocation
   B. overloading functions and operators
   C. constructors and destructors
II. Recursion
   A. analysis
   B. applications
III. ADT implementations (static and dynamic) and applications
   A. linked lists (singly, doubly, circular)
   B. stacks
   C. queues
   D. trees and tree traversal

IV. Searching
  A. analyzing and comparing searching methods
  B. relationships between methods and data structures
V. Sorting
  A. examples of order n^2 and order n log(n) sorts
  B. analyzing and comparing sorting methods
VI. Optional other topics
  A. hashing
  B. tables
  C. file structures
  D. sets
  E.Templates

5. Suggested Time Allocation per Topic

In order to standardize the core topics of CSC 202 so that a course taught at one campus is equivalent to the same course taught at another campus, the following student contact hours per topic are recommended. Of course, the topics cannot be followed sequentially. Many topics are taught best as an integrated whole, often revisiting the topic several times, each time at a higher level. There are normally 64 student-contact-hours per semester for a four-unit course. The last category, ***Other/Enhance***, leaves ample time for an instructor to tailor the course to special needs or resources.

Ref	Topic	Hours	Percent
I	Data abstraction and object-oriented programming	10	16
II	Concept of Recursion	2	3
III A	Linked lists/pointers	10	16
III B	Stacks	8	13
III C	Queues	8	13
III D	Trees	2	3
IV	Searching	4	7
V	Sorting	6	9
VI	***Other*** optional content or ***Enhance*** the above	8	13
	Exams and quizzes	6	9
	Total	64	100