CSC 1254: Computer Science II with C++

Credit Hours: 3 hours

Frequency: Fall and Spring semesters

Prerequisites:
> CSC 1253, MATH 1550 or registration in MATH 1435.

Prerequisites by Topics:
> On e course introductory course covering structured programming in C++
> Basic mathematics understanding

Catalog Course Description:
> Credit will not be given for both this course and CSC 1351. Develops solutions to problems using an object-oriented approach and emphasizes the concepts of: recursion; dynamic memory; data structures (lists, stacks, queues, trees); exception handling.

Course Outcomes
1. Be familiar with object-oriented design and coding.
2. Be familiar with algorithm efficiency.
3. Be familiar with data abstraction.
4. Be familiar with how to design and implement abstract data types.
5. Be familiar with text files and binary files.
6. Be familiar with recursion and recursive sorts.

Texts and Other Course Materials
> Foundations in C++ - Water Savitch 0-536-94800-3 PB Latest Addison

Major Topics
- Binary files,
- Dynamic memory,
- Constructors, copy constructors, destructors,
- Overloading,
- Virtual functions (late binding),
- Templates,
- Separate compilation,
- Namespaces (creating),
- Recursion and recursive sorts (meregand quick sorts),

- Exception handling and assertions,
- Robust verification of input data,
- Pointer arithmetic,
- Function pointers,
- Data abstraction, ordered list ADT, stack ADT, queue ADT, binary trees Insertion, deletion, and traversals of ADTs including binary search trees (BST),
- Asymptotic efficiency of the operations associated with an abstract data type,
- Classes and inheritance,
- Advanced linked lists: doubly linked list, circular linked list,
- Standard Template Library (STL) and preview of containers.

Assignments/Projects/Laboratory Projects/Homework
- Review array-based lists
- Menu-driven interfaces to manipulate a database
- Implement a linked list implementation of an array
- Use recursive sorts
- Use tree traversals

Curriculum Category Content (estimated in semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms	5	3	Data Structures	16	3
Software Design	4	1	Prog. Languages	1	0
Computer Arch.	2	0			

Relationship to Criterion 3 Outcomes

A	B	C	D	E	F	G	H	I	J	K
*	*	*		*	*	*	*		*	

Math Fundamentals:

Data Structures:
   Introduction to the ADTs (lists, stacks, queues, trees, BSTs) (18 hrs)
Algorithms and Software:
   Algorithm analysis (8 hrs),
   Software design (5 hrs)
Computer Organization and Architecture:
   Simulation of run-time stacks using activation record instances (2hrs)
Concepts of Programming Languages:
   Brief comparison of imperative languages and other languages (1 hr)

Social and Ethical Issues:

Credit code-author when using code re-use for ethical reasons and for good documentation practice.

Oral Communication (presentations) – none

Written Communication:

6-8 homeworks and projects, 5 pages per assignment

Course Coordinator: Dr. William Duncan

Last Modified: April 27, 2007