

COURSE IMPLEMENTATION DATE:[September 2000]
 COURSE TO BE REVIEWED DATE:[September 2004]
 (Four years after implementation date)

OFFICIAL COURSE OUTLINE INFORMATION

Students are advised to keep course outlines in personal files for future use.

Shaded headings are subject to change at the discretion of the department and material will vary
 - see course syllabus available from instructor

FACULTY/DEPARTMENT: COMPUTER INFORMATION SYSTEMS

COMP 251 COMP 175 4

COURSE NAME/NUMBER FORMER COURSE NUMBER UCFV CREDITS

DATA STRUCTURES AND ALGORITHMS

COURSE DESCRIPTIVE TITLE

CALENDAR DESCRIPTION:

This course is an introduction to the abstract data structures used in the solution of common computing problems, including stacks, queues, trees and graphs. Students will apply the concepts to problems using a modern object-oriented programming language. The course will also include a discussion and assignments on the correctness and efficiency of algorithms.

PREREQUISITES: COMP 155 or COMP 160; MATH 12 with C+ or better recommended and will be required Fall 2001

COREQUISITES: None

SYNONYMOUS COURSE(S)

(a) Replaces: N/A
 (Course #)
 (b) Cannot take N/A for further credit
 (Course #)

SERVICE COURSE TO:

(Department / Program)

(Department / Program)

TOTAL HOURS PER TERM: 56

STRUCTURE OF HOURS:

Lectures: 42 hrs
 Seminar: hrs
 Laboratory: 14 hrs
 Field Experience: hrs
 Student Directed Learning: hrs
 Other (Specify): hrs

TRAINING DAY-BASED INSTRUCTION

LENGTH OF COURSE: _____

HOURS PER DAY: _____

MAXIMUM ENROLMENT: 35

EXPECTED FREQUENCY OF COURSE OFFERING: _____

WILL TRANSFER CREDIT BE REQUESTED? YES _____ NO X

TRANSFER CREDIT EXISTS IN BCCAT TRANSFER GUIDE: YES _____ NO X

AUTHORIZATION SIGNATURES:

Course designer(s): _____

Chairperson: _____

Lee Chen

(Curriculum Committee)

Department Head: _____

Dean: _____

Paul Franklin

K. Wayne Welsh

PAC Approval in Principle Date: _____

PAC Final Approval Date: September 27, 2000

 COURSE NAME / NUMBER

LEARNING OBJECTIVES / GOALS / OUTCOMES/ LEARNING OUTCOMES:

Understanding of the design principles of various types of data structures and algorithms commonly used in software development. Basic concepts of data structures and algorithms are introduced through programming projects using the C++ programming language.

METHODS:

Lecture/lab

PRIOR LEARNING ASSESSMENT RECOGNITION (PLAR):

Credit can be awarded for this course through PLAR YES _____ NO X

METHODS OF OBTAINING PLAR:**TEXTBOOKS, REFERENCES, MATERIALS:**

Michael Maian & Walter Savitch, *Data Structures and Other Objects Using C++*, Addison Wesley, 1997.

SUPPLIES / MATERIALS:**STUDENT EVALUATION:**

Assignments	20%
Exam 1	25%
Exam 2	25%
Final exam	30%

COURSE CONTENT:

Applications and analysis, and the following constructs:

- C Preconditions/Postconditions
- C Simple Algorithm Analysis (Big-O Notation)
- C Container Classes
- C Pointer Usage
- C Dynamic Arrays
- C Introduction to Linked Lists, Building a Linked List Toolkit, Using the Linked List Toolkit
- C Template Functions, Template Classes, the Stack Template Class
- C Applications that use Stacks
- C Queues and Priority Queues
- C Introduction to Recursion
- C Fractals
- C Trees and Traversals, Binary Search Trees, Heaps and B-Trees
- C Sorting - Quadratic Sorting, Mergesort, Quicksort