San Bernardino Valley College
Curriculum Approved: January 24, 2005

I. COURSE INFORMATION:

A.

Division:	Science and Math
Department:	Computer Science
Course ID:	CS 265
Course Title:	Data Structures
Units:	3
Lecture Hours:	2
Laboratory Hours:	3
Prerequisite:	CS 190
Corequisite:	None
Dept. Advisory:	None

B. Catalog and Schedule Description: An introduction to data structures such as linked lists, stacks, queues, and trees. Topics include algorithm development, storage allocation, data organization, information retrieval system software, and language support features.

II. NUMBER OF TIMES COURSE MAY BE TAKEN FOR CREDIT: One time

III. EXPECTED OUTCOMES:

Upon successful completion of the course, the student should be able to:

A. Design an algorithm for programming projects using data structures.
B. Use object-oriented design and programming techniques.
C. Organize and process stacks in application programming techniques.
D. Organize and process queues in application programs and system programs.
E. Organize and process linked lists in application programs and system programs.
F. Organize and process tress in application programs and system programs.
G. Compare and contrast programming techniques such as sorting.
H. Compare and contrast programming techniques such as searching, hashing and recursion.
I. Recognize the significance of documentation.
J. Develop good programming style.
K. Transfer the techniques learned in this course into another programming course.

IV. COURSE CONTENT:

A. Data structures and Programming principles
   1. Structures of data and definitions
   2. Documentation and format
   3. Refinement and modularity
   4. Program revision and redevelopment
B. Stacks
   1. Lists and arrays
   2. Specifications and implementation
   3. Push and pop
   4. Case study: The Polish notation
C. Queues
   1. Circular implementation using C++
   2. Applications: simulations and random numbers
D. Linked stacks and queues
   1. Dynamic memory and pointers
   2. Declarations and operators
   3. Constructor and destructor
E. Recursion
   1. Recursive algorithm design
   2. Backtracking
   3. Stack frames for subprograms

4. Tree-structured programs

F. Lists and stings
1. Simple linked and double-linked lists
2. Contiguous implementation
3. Strings and string operations

G. Searching
1. Sequential search
2. Binary search
3. Comparison trees

H. Sorting
1. Insertion sort
2. Selection sort
3. Shell sort
4. Mergesort for linked lists

I. Tables and information retrieval
1. Tables of various shapes
2. Radix sort for tables
3. Hashing

J. Binary trees
1. Ordered lists and implementation
2. Tree search
3. Height balance
4. Splay trees

K. Graphs
1. Direct and undirected graphs
2. Traversal
3. Topological sorting
4. Minimum spanning trees

V. METHODS OF INSTRUCTION: (Please check all that apply and add any additional not listed.)

X	Lecture
X	Class and/or small group discussion
_____	Critical evaluation of texts, newspapers, journal articles, and other printed research
X	Critical evaluation of films, videotapes, audiotapes, or other media forms
_____	Classroom demonstrations
_____	Field trips
_____	Guest speakers
X	Other: Projects
X	Other: Examinations
_____	Other:

VI. TYPICAL OUT-OF-CLASS ASSIGNMENTS:

A. Read the first chapter on algorithms and concisely summarize the main topics. Pay special attention to how the examples affect algorithmic design.

B. Analyze the proposed algorithm on the web site and write a program that corresponds to it.

C. In the online course chat area, discuss the use of recursion as a problem-solving technique.

D. Assess the efficiency of the algorithm and suggest modifications to increase the efficiency of the process.

VII. EVALUATION:

A student's grade will be based on multiple measures of performance and will reflect the objectives explained above. A final grade of "C" or better should indicate that the student has the ability to successfully apply the principles and techniques taught in this course. These evaluation methods may include, but are not limited to, the following (Please check all that apply, and add additional ones not listed):

_____	Portfolios	
X	Projects	
X	Written papers or reports	
X	Presentations (oral and visual)	
_____	Work performance (internships or field work)	
X	Lab work	
_____	Comprehensive examinations (cumulative finals or certifications)	
_____	Peer evaluation	
_____	Self evaluation	
_____	Classroom participation	
_____	Homework	
X	Other: Programming projects	One project per week
X	Other: Examinations and quizzes	Two exams: midterm and final
X	Weekly quizzes on reading assignments	
_____	Other:	

VIII. TYPICAL TEXTS:

1. Data Structures and Other Objects Using C++, Main, Michael, et al.; Addison Wesley Professional, 2004.
2. Data Structures and Algorithms in C++, Drozdek; Course Technology Incorporated, 2004.
3. ADTs, Data Structures, and Problem Solving in C++, Nyhoff, Larry; Prentice Hall PTR, 2004.

IX. OTHER SUPPLIES REQUIRED OF STUDENTS: None