CS 11 Course Outline as of Fall 2009

CATALOG INFORMATION

Dept and Nbr: CS 11 Title: DATA STRUCT & ALGOR Full Title: Data Structures and Algorithms Last Reviewed: 3/27/2023

Units		Course Hours per Week		Nbr of Weeks	Course Hours Total	
Maximum	4.00	Lecture Scheduled	3.00	17.5	Lecture Scheduled	52.50
Minimum	4.00	Lab Scheduled	0	17.5	Lab Scheduled	0
		Contact DHR	3.00		Contact DHR	52.50
		Contact Total	6.00		Contact Total	105.00
		Non-contact DHR	0		Non-contact DHR	0

Total Out of Class Hours: 105.00

Total Student Learning Hours: 210.00

Title 5 Category:	AA Degree Applicable
Grading:	Grade Only
Repeatability:	00 - Two Repeats if Grade was D, F, NC, or NP
Also Listed As:	
Formerly:	CIS 11

Catalog Description:

Continued study of computer programming including specification and implementation of data structures, and analysis of associated algorithms. Topics include: abstract data types, dynamic memory, templated functions and classes, iterators, exception handling, linked lists, stacks, queues, recursion, trees, searching, sorting, and inheritance. Several significant programming projects are written in C ++.

Prerequisites/Corequisites: Course Completion of CS 10A (or CS 10 or CIS 10 or CIS 10A or CIS 10 or BDP 10)

Recommended Preparation:

Eligibility for ENGL 1A or equivalent

Limits on Enrollment:

Schedule of Classes Information:

Description: Continued study of computer programming including specification and implementation of data structures, and analysis of associated algorithms. Topics include: abstract data types, dynamic memory, iterators, linked lists, stacks, queues, recursion, trees, searching, sorting, and inheritance. Several significant programming projects written in C++. (Grade Only)

Prerequisites/Corequisites: Course Completion of CS 10A (or CS 10 or CIS 10 or CIS 10A or CIS 10 or BDP 10) Recommended: Eligibility for ENGL 1A or equivalent Limits on Enrollment: Transfer Credit: CSU;UC. Repeatability: Two Repeats if Grade was D, F, NC, or NP

ARTICULATION, MAJOR, and CERTIFICATION INFORMATION:

AS Degree: CSU GE:	Area Transfer Area			Effective: Effective:	Inactive: Inactive:
IGETC:	Transfer Area			Effective:	Inactive:
CSU Transfer	:Transferable	Effective:	Spring 1991	Inactive:	
UC Transfer:	Transferable	Effective:	Spring 1991	Inactive:	

CID:

CID Descriptor:COMP 132 SRJC Equivalent Course(s): Programming Concepts and Methodology II CS10C

Certificate/Major Applicable:

Certificate Applicable Course

COURSE CONTENT

Outcomes and Objectives:

Upon completion of the course, students will be able to:

- 1. Analyze algorithms for efficiency.
- 2. Use data abstraction as a tool for modeling.
- 3. Design the system life cycle through software engineering principles.
- 4. Construct linked lists, pointers, queues, stacks, and trees as abstract data types.
- 5. Design and construct iterative approaches to algorithm development.
- 6. Evaluate a variety of sorting and searching methods for efficiency.

Topics and Scope:

- A. Abstract data types: operator overloading
 - 1. Overloading arithmetic, shorthand arithmetic, and relational operators
 - 2. Overloading the insertion and extraction operators
 - 3. Overloading the pre and post decrement and increment operators
 - 4. Overloading the square brackets (subscript) operator
 - 5. Overloading operators as member functions, free (global) functions, and friend functions.
 - 6. Default parameters
- B. Pointers and Dynamic Memory
 - 1. Address operator

- 2. Dereference (indirection) operator
- 3. Pointer assignment
- 4. Arrow (dereference and select) operator
- 5. Arrays of pointers
- 6. NULL
- 7. Relationship of arrays and pointers
- "new" operator
 "delete" operator
- 10. Memory leaks
- 11. Reference types
- C. Dynamic memory in classes
 - 1. Assignment operator, including checking for self-assignment
 - 2. Copy constructor
 - 3. Destructor
- D. Container Classes
 - 1. Documenting member functions with pre/post conditions
 - 2. Using std::size_t
 - 3. Returning a reference
 - 4. User defined namespaces
- E. Linked lists
- F. Templated functions and classes
- G. Iterators, including user-defined
- H. Standard Template Library
- I. Exception handling
- J. Stacks, including expression evaluation
- K. Queues
- L. Recursion
 - 1.Development techniques
 - 2. Analysis techniques
- M. Trees: specification, implementation, and big-O analyses of
 - 1.Binary search trees
 - 2.Heaps
- N. Searching: specification, implementation, and big-O analysis of
 - 1.Sequential search
 - 2.Binary search
 - 3.Hashing
- O. Sorting: specification, implementation, and big-O analysis of
 - 1. Selection sort
 - 2. Insertion sort
 - 3. Bubble sort
 - 4. Merge sort
 - 5. Quicksort
 - 6. Heapsort
- P. Inheritance
 - 1. Contrasted with composition
 - 2. "is-a" relationship and "has-a" relationship
 - 3. Protected
 - 4. Constructors in inheritance
 - 5. Initializor lists
 - 6. Polymorphism and virtual functions
 - 7. The slicing problem
 - 8. Pure virtual functions and abstract classes

Assignment:

- 1. Read approximately 40 pages per week from text book.
- 2. Write computer programs.
- 3. Correct errors in programs with multiple runs of test data.
- 4. Write documentation for each program so that is easy to understand and use.
- 5. Take objective exams.

Methods of Evaluation/Basis of Grade:

Writing: Assessment tools that demonstrate writing skills and/or require students to select, organize and explain ideas in writing.

None, This is a degree applicable course but assessment tools based on writing are not included because problem solving assessments are more appropriate for this course.

Problem Solving: Assessment tools, other than exams, that demonstrate competence in computational or non-computational problem solving skills.

Computer programming assignments

Skill Demonstrations: All skill-based and physical demonstrations used for assessment purposes including skill performance exams.

None

Exams: All forms of formal testing, other than skill performance exams.

Multiple choice, True/false, Matching items, Completion,	,
short answer & programming	

Other: Includes any assessment tools that do not logically fit into the above categories.

None

Representative Textbooks and Materials:

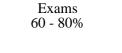
Data Structures and other Objects Using C++, 2nd Ed, by Michael Main and Walter Savitch - Addison Wesley Longman 2000

Writing 0 - 0%	

20 - 40%	U

Problem solving

Skill Demonstrations
0 - 0%



Other Category	
0 - 0%	