CIS 11 Course Outline as of Fall 2003

CATALOG INFORMATION

Dept and Nbr: CIS 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	5.00		Contact DHR	87.50
		Contact Total	8.00		Contact Total	140.00
		Non-contact DHR	0		Non-contact DHR	0

Total Out of Class Hours: 105.00 Total Student Learning Hours: 245.00

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:	

Catalog Description:
Intended as a second course for those pursuing computer programming. Topics include software engineering principles, the development of efficient algorithms, the design and selection of appropriate data structures, and an overview of computer science. Numerous programs are written in a suitable high-level language. Required for the Programmer certificate. Appropriate for those pursuing a four-year degree in computer science.

Prerequisites/Corequisites:
Course Completion of CIS 10B

Recommended Preparation:
Eligibility for Engl 1A or equivalent.

Limits on Enrollment:

Schedule of Classes Information:
Description: Intended as a second course for those pursuing computer programming. Topics include software engineering principles, the development of efficient algorithms, the design & selection of appropriate data structures, & an overview of computer science. Numerous programs are written in a suitable high-level language. (Grade Only)

Prerequisites/Corequisites: Course Completion of CIS 10B
Recommended: Eligibility for Engl 1A or equivalent.
Limits on Enrollment:
Transfer Credit: CSU;UC. (CAN CIS 11+CIS 11=CSCI24)(CIS 11+CIS 11=CSCI24)
Repeatability: Two Repeats if Grade was D, F, NC, or NP

ARTICULATION, MAJOR, and CERTIFICATION INFORMATION:

AS Degree:	**Area**	Effective:	Inactive:
CSU GE:	**Transfer Area**	Effective:	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 Programming Concepts and Methodology II
SRJC Equivalent Course(s): 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, trees and graphs as abstract data types.
5. Design and construct iterative approaches to algorithm development.
6. Evaluate a variety of sorting and searching methods for efficiency.
7. Describe the development of language compilers and operating systems.
8. Formulate software reliability through formal verification.
9. Compare the relationship of the halting problem to the quest for artificial intelligence.

Topics and Scope:

1. Algorithm Analysis.
   A. Program design.
   B. Big-O analysis.
2. Data Abstraction.
   A. Science as model building.
   B. Examples of abstract data types.
3. Software Engineering and the System Life Cycle.
   A. The life cycle and software reliability.

   B. Initial implementation.
   C. Maintenance.
4. Linked Lists.
   A. Linked lists as Abstract Data Types (ADTs).
   B. Linked lists implemented as arrays.
   C. Pointer variables.
   D. Strings and sparse tables.
5. Queues.
   A. Characterization and implementation.
   B. Queues as ADTs.
   C. Priority Queues.
6. Stacks and Recursion.
   A. Stacks as ADTs.
   B. Implementation of stacks.
   C. Stacks and recursion.
7. Trees and Graphs.
   A. Trees as ADTs.
   B. Implementation of trees.
   C. Graphs and networks.
   D. Heaps.
8. Analysis of Recursion.
   A. Recursion and generalized nested loops.
   B. Problem solving with backtracking.
   C. Tradeoffs associated with recursion.
9. Sorting Algorithms.
   A. Shell sort.
   B. Quick sort.
   C. Heap sort.
   D. Merge sort.
10. Search Algorithms.
   A. Serial Search.
   B. Binary Searh.
   C. Hashing.
11. Computer Science.
   A. Language compilers.
   B. Operating systems.
   C. Software reliability and formal verification.
   D. Artificial intelligence.
   E. The halting problem.

Assignment:

1. Read approximately 40 pages per week from text book.
2. Prepare hierarchy charts.
3. Design flowcharts, Chapin charts, pseudocode, or other design tools.
4. Write computer programs.
5. Correct errors in programs with multiple runs of test data.
6. Write documentation for each program so that is is easy to understand and use.
7. 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.	Writing 0 - 0%

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

Homework problems, Lab reports	Problem solving 20 - 40%

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

None	Skill Demonstrations 0 - 0%

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

Multiple choice, True/false, Matching items, Completion, SHORT ANSWER & PROGRAMMING	Exams 60 - 80%

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

None	Other Category 0 - 0%

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