CIS 10B Course Outline as of Spring 2002

CATALOG INFORMATION

Dept and Nbr: CIS 10B Title: COMP SCI FUNDAMENTALS II
Full Title: Computer Science Fundamentals II
Last Reviewed: 11/5/2001

Units		Course Hours per Week		Nbr of Weeks	Course Hours Total	
Maximum	3.00	Lecture Scheduled	2.00	17.5	Lecture Scheduled	35.00
Minimum	3.00	Lab Scheduled	0	8	Lab Scheduled	0
		Contact DHR	3.00		Contact DHR	52.50
		Contact Total	5.00		Contact Total	87.50
		Non-contact DHR	0		Non-contact DHR	0

Total Out of Class Hours: 70.00 Total Student Learning Hours: 157.50

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:
This course concludes the introduction of computer science and software engineering begun in CIS 10A. Topics include basic data structures, files, databases, artificial intelligence, theory of computation, abstract data types, arrays and pointers, recursion and inheritance. Eight to ten complete programs are written in C++. Serves as preparation for CIS 11: Data Structures & Algorithms.

Prerequisites/Corequisites:
Completion of CIS 10A (formerly CIS 10, BDP 10) with a grade of 'C' or better.

Recommended Preparation:
Eligibility for ENGL 1A or equivalent

Limits on Enrollment:

Schedule of Classes Information:
Description: Concludes the introduction of computer science and software engineering begun in CIS 10A. Includes basic data structures, files, databases, AI and computation theory, abstract data types, arrays and pointers, recursion and inheritance. Eight to ten complete programs are written in C++. Serves as preparation for CIS 11: Data Structures and Algorithms. (Grade Only)

Prerequisites/Corequisites: Completion of CIS 10A (formerly CIS 10, BDP 10) with a grade of 'C' or better.
Recommended: Eligibility for ENGL 1A or equivalent
Limits on Enrollment:
Transfer Credit: CSU;UC. (CAN CIS 10+CIS 10B=CSCI22)
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:	Fall 2000	Inactive:	Summer 2010
UC Transfer:	Transferable	Effective:	Fall 2000	Inactive:	Summer 2010

CID:

Certificate/Major Applicable:
Certificate Applicable Course

COURSE CONTENT

Outcomes and Objectives:
Students will:
1. List common data models and the structures employed in their implementations.
2. Identify appropriate secondary storage techniques for a variety of applications.
3. Contrast flat-file, relational, and object-oriented database designs.
4. Relate and contrast natural and artificial intelligence.
5. Write simple Turing machine programs.
6. Identify computable and non-computable problems.
7. Describe the complexities of selected problems.
8. Produce multi-file programs using an integrated development environment.
9. Implement Abstract Data Types in C++.
10. Produce complete classes for use by client programmers.
11. Use recursion, dynamic allocation and inheritance when implementing software designs.

Topics and Scope:

A. Computer Science
  1. Data Organization
    a. Data Structures
      1) Arrays
      2) Lists

3) Stacks & Queues
4) Trees
5) ADTs
6) Pointers

b. Files
1) Low-level access
2) Sequential & Random Access
3) Text Files
4) Indexing & Hashing

c. Databases
1) Tables
2) The Relational Model
3) Object-Oriented Databases
4) Database Integrity

2. Algorithmic Machines

a. Artificial Intelligence
1) Machine Intelligence
2) Images
3) Reasoning
4) Neural Networks
5) Genetic Algorithms
6) Applications

b. Theory of Computation
1) An Example Minimal Language
2) Turing Machines
3) Computable & Non-computable Functions
4) Complexity

B. Programming

1. Abstract Data Types
a. Interface Files
b. Implementation Files
c. Friend Functions
d. Constant Parameters
e. Operator Overloading
f. Constructors and Automatic Type Conversion

2. Arrays and Strings
a. Declaring & Referencing Arrays
b. Initializing Arrays
c. Arrays as Arguments
d. Returning Arrays
e. Arrays & Classes
f. C-Style Strings
g. The C++ Standard String Class

3. Pointers & Dynamic Memory Allocation
a. Pointer Variables
b. Arrays and Pointers
c. Pointer Arithmetic
d. Storage Classes
e. Dynamic Structures
f. Copy Constructors
g. Destructors
h. Overloading Assignment Operators

4. Recursion
   a. Tail Recursion
   b. Induction
   c. Stack Frames
   d. Recursion Elimination
   e. Design Techniques
5. Inheritance
   a. Derived Classes
   b. Constructors & Initialization
   c. Virtual Functions & Polymorphism
   d. Abstract Classes

Assignment:

1. Maintain a reading schedule for the text(s).
2. Write programs using the C++ programming language.
3. Test and debug programs.
4. Write program documentation.

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.

Written program documentation	Writing 10 - 20%

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

Homework problems, Exams, LABORATORY ASSIGNMENTS	Problem solving 30 - 60%

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, Programming exercises	Exams 30 - 60%

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

None	Other Category 0 - 0%

Representative Textbooks and Materials:

"Problem Solving with C++: The Object of Programming", by Walter Savitch - Addison-Wesley Longman 1999

"Computer Science: An Overview", by J. Brookshear - Addison-Wesley Longman 1997