

**CIS 110A Course Outline as of Spring 2004****CATALOG INFORMATION**

Dept and Nbr: CIS 110A Title: PROGRAMMING CONCEPTS

Full Title: Concepts for Beginning Programmers

Last Reviewed: 9/10/2018

Units		Course Hours per Week		Nbr of Weeks	Course Hours Total	
Maximum	1.50	Lecture Scheduled	3.00	8	Lecture Scheduled	24.00
Minimum	1.50	Lab Scheduled	0	8	Lab Scheduled	0
		Contact DHR	0		Contact DHR	0
		Contact Total	3.00		Contact Total	24.00
		Non-contact DHR	0		Non-contact DHR	0

Total Out of Class Hours: 48.00

Total Student Learning Hours: 72.00

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:**

A brief introduction to computer programming concepts using languages appropriate to such learning. This course is designed for students who do not intend to pursue further study of computer programming or who intend to continue in the field but desire a gentle introduction to the subject. Taught in a hands-on environment.

**Prerequisites/Corequisites:****Recommended Preparation:****Limits on Enrollment:****Schedule of Classes Information:**

Description: A brief introduction to computer programming concepts using languages appropriate to such learning. This course is designed for students who do not intend to pursue further study of computer programming or who intend to continue in the field but desire a gentle introduction to the subject. Taught in a hands-on environment. (Grade Only)

Prerequisites/Corequisites:

Recommended:

Limits on Enrollment:

Transfer Credit:

Repeatability: Two Repeats if Grade was D, F, NC, or NP

## **ARTICULATION, MAJOR, and CERTIFICATION INFORMATION:**

<b>AS Degree:</b>	<b>Area</b>	Effective:	Inactive:
<b>CSU GE:</b>	<b>Transfer Area</b>	Effective:	Inactive:

<b>IGETC:</b>	<b>Transfer Area</b>	Effective:	Inactive:
---------------	----------------------	------------	-----------

<b>CSU Transfer:</b>	Effective:	Inactive:
----------------------	------------	-----------

<b>UC Transfer:</b>	Effective:	Inactive:
---------------------	------------	-----------

**CID:**

**Certificate/Major Applicable:**

Certificate Applicable Course

## **COURSE CONTENT**

### **Outcomes and Objectives:**

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

1. Create correct algorithms
2. Design and implement computer programs that employ conditional and repetitive control structures.
3. Design and implement computer programs that employ subprograms.
4. Classify program errors.
5. Design and implement computer programs using stepwise refinement.
6. Design and implement computer programs that employ simple recursion.

### **Topics and Scope:**

- I. Primitive instructions
- II. Basic program structure
- III. Classifying programming errors
- IV. Creating and calling subprograms
  - A. The Correctness of subprograms
  - B. Stepwise refinement
  - C. Program design techniques
  - D. Advantages of using subprograms
  - E. Writing understandable programs
- V. Conditional Execution
  - A. If statements
  - B. If/else statements
  - C. Nested if statements
  - D. Complex conditions
  - E. When to use conditional execution
  - F. Transformations for simplifying if statements

- G. The dangling else
- VI. Repetitive Execution
  - A. While statements
  - B. Errors to avoid with repetitive execution
  - C. Nested while statements
  - D. Reasoning about while statements
  - E. When to use repetitive execution
- VII. Simple recursion

### Assignment:

1. Maintain a reading schedule for the text(s), approximately 20 pages per week.
2. Write computer programs.
3. Test and debug computer programs.
4. Take objective examinations.

### 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.

Writing computer programs

Problem solving  
30 - 70%

**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, Computer programming questions

Exams  
30 - 70%

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

None

Other Category  
0 - 0%

### Representative Textbooks and Materials:

"A Gentle Introduction to the Art of Programming, Second Edition",  
by Richard E. Pattis - John Wiley & Sons, Inc., 1995