

CS 110A Course Outline as of Fall 2013**CATALOG INFORMATION**

Dept and Nbr: CS 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	1.50	17.5	Lecture Scheduled	26.25
Minimum	1.50	Lab Scheduled	0	4	Lab Scheduled	0
		Contact DHR	0		Contact DHR	0
		Contact Total	1.50		Contact Total	26.25
		Non-contact DHR	0		Non-contact DHR	0

Total Out of Class Hours: 52.50

Total Student Learning Hours: 78.75

Title 5 Category: AA Degree Applicable

Grading: Grade or P/NP

Repeatability: 00 - Two Repeats if Grade was D, F, NC, or NP

Also Listed As:

Formerly: CIS 110A

Catalog Description:

A brief introduction to computer programming concepts. Topics include algorithms, decision structures, repetition structures, and subprograms.

Prerequisites/Corequisites:**Recommended Preparation:**

Eligibility for ENGL 100 or ESL 100

Limits on Enrollment:**Schedule of Classes Information:**

Description: A brief introduction to computer programming concepts. Topics include algorithms, decision structures, repetition structures, and subprograms. (Grade or P/NP)

Prerequisites/Corequisites:

Recommended: Eligibility for ENGL 100 or ESL 100

Limits on Enrollment:

Transfer Credit:

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:		Effective:	Inactive:
UC Transfer:		Effective:	Inactive:

CID:

Certificate/Major Applicable:

Not Certificate/Major Applicable

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:

1. Primitive Instructions
2. Basic Program Structure
3. 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
4. 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
5. Repetitive Execution
 - a. While statements, do-while statements, and for statements
 - b. Errors to avoid with repetitive execution
 - c. Nested while statements

- d. Reasoning about while statements
- e. When to use repetitive execution

Assignment:

1. Read approximately 30 pages per week.
2. 2-8 computer programming assignments.
3. Test and debug computer programs.
4. 2-6 objective quizzes and/or 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.

Computer programming assignments, including testing and debugging 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:

Absolute C++ (5th). Savitch, Walter. Addison-Wesley: 2012
 Python Programming: An Introduction to Computer Science (2nd). Zelle, John. Franklin, Beedle & Associates Inc: 2010