

**CS 110A Course Outline as of Fall 2019****CATALOG INFORMATION**

Dept and Nbr: CS 110A Title: CODING FOR BEGINNERS

Full Title: Coding for Beginners

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:**

This course is designed to teach basic computer programming concepts to anyone – no programming experience required. Using simple tools, students will learn the building blocks of computer programs in a stress-free environment. This introduction to coding will guide students through the process of creating simple programs, starting with graphical coding tools.

**Prerequisites/Corequisites:****Recommended Preparation:**

Eligibility for ENGL 100 or ESL 100

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

Description: This course is designed to teach basic computer programming concepts to anyone – no programming experience required. Using simple tools, students will learn the building blocks of computer programs in a stress-free environment. This introduction to coding will guide students through the process of creating simple programs, starting with graphical coding tools.  
(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:**

<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:**

Not Certificate/Major Applicable

## **COURSE CONTENT**

### **Student Learning Outcomes:**

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

1. Design and implement computer programs that employ basic computer programming concepts.

### **Objectives:**

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

1. Describe the fundamental components of a computer program.
2. Create simple coding projects using drag-and-drop tools.
3. Understand the correct usage of conditionals, variables, and loops.

### **Topics and Scope:**

- I. What is Coding/Programming
- II. The Fundamental Components of a Program
  - A. Conditionals
    1. When to use
    2. How to use conditionals
  - B. Variables
    1. What are variables
    2. Variable types
    3. When to use
    4. How to use variables
  - C. Loops
    1. Different types of loop structures
    2. When to use
    3. How to implement loops
- III. Dealing with Input and Output

- A. Input
  - 1. Button presses
  - 2. Keyboard input
  - 3. Sensor input
    - i. Motion
    - ii. Environmental
- B. Output
  - 1. LEDs
  - 2. Screen
  - 3. Sounds
  - 4. Servos

#### IV. Creating Subprograms

#### Assignment:

1. Read approximately 20 pages per week
2. Coding assignments (2 - 8)
3. Test and debug computer programs
4. One to three 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  
50 - 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  
20 - 40%

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

Participation and attendance

Other Category  
0 - 10%

**Representative Textbooks and Materials:**

The Official BBC Micro:Bit User Guide. Halfacree, Gareth. Wiley. 2017