CS 10A Course Outline as of Fall 2021

CATALOG INFORMATION

Dept and Nbr: CS 10A Title: INTRO TO PROGRAMMING Full Title: Introduction to Programming Concepts and Methodologies Last Reviewed: 2/8/2021

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	3.00	6	Lab Scheduled	52.50
		Contact DHR	0		Contact DHR	0
		Contact Total	6.00		Contact Total	105.00
		Non-contact DHR	0		Non-contact DHR	0

Total Out of Class Hours: 105.00

Total Student Learning Hours: 210.00

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:	CS 10

Catalog Description:

Specification, design, implementation, testing, debugging, maintenance, and documentation of computer programs. Topics include algorithms, languages, software engineering, control structures, functions, and arrays. Numerous programs are written in C++. Intended for the absolute beginner, both computer science majors and those seeking a general introduction to computer programming.

Prerequisites/Corequisites:

Recommended Preparation:

Eligibility for ENGL 1A or equivalent or appropriate placement based on AB705 mandates

Limits on Enrollment:

Schedule of Classes Information:

Description: Specification, design, implementation, testing, debugging, maintenance, and documentation of computer programs. Topics include algorithms, languages, software engineering, control structures, functions, and arrays. Numerous programs are written in C++. Intended for the absolute beginner, both computer science majors and those seeking a general

introduction to computer programming. (Grade or P/NP) Prerequisites/Corequisites: Recommended: Eligibility for ENGL 1A or equivalent or appropriate placement based on AB705 mandates Limits on Enrollment: Transfer Credit: CSU;UC. Repeatability: Two Repeats if Grade was D, F, NC, or NP

ARTICULATION, MAJOR, and CERTIFICATION INFORMATION:

AS Degree: CSU GE:	Area Transfer Area	I		Effective: Effective:	Inactive: Inactive:
IGETC:	Transfer Area	l		Effective:	Inactive:
CSU Transfer	:Transferable	Effective:	Spring 1989	Inactive:	
UC Transfer:	Transferable	Effective:	Spring 1989	Inactive:	

CID:

CID Descriptor:COMP 122	Programming Concepts and Methodology I
SRJC Equivalent Course(s):	CS10A OR CS10B
CID Descriptor:COMP 112	Introduction to Programming Concepts and Methodologies
SRJC Equivalent Course(s):	CS10A

Certificate/Major Applicable:

Both Certificate and Major Applicable

COURSE CONTENT

Student Learning Outcomes:

At the conclusion of this course, the student should be able to:

- 1. Describe the principles of structured programming and be able to describe, design, implement, and test structured programs using currently accepted methodology.
- 2. Explain what an algorithm is and its importance in computer programming.

Objectives:

At the conclusion of this course, the student should be able to:

- 1. Create correct and efficient algorithms.
- 2. Describe the software-development life cycle.
- 3. Employ the basic elements of the C++ language.
- 4. Implement algorithms using C++ flow-control constructs.
- 5. Write descriptive and helpful program documentation.
- 6. Implement algorithms using arrays.

Topics and Scope:

I. Software Life-Cycle

- A. Design
- B. Development
- C. Styles
- D. Documentation

E. Testing

F. Maintenance

- II. Procedural Versus Objected Oriented Programming Survey of Current Languages
- III. Program Design Tools and Programming Environments
- IV. Documentation
- V. Coding Conventions
- VI. Data Types, Variables, Expressions, Sequential Processing
- VII. Arrays
 - A. Declaring and allocating arrays
 - B. Multiple-subscripted arrays
- VIII. Control Structure
 - A. Selective structures: if and switch
 - B. Repetitive structures: loops
- IX. Algorithms
 - A. Simple Sorting
 - B. Searching
- X. File I/O
 - A. Files and streams
 - B. Sequential access files
- XI. Error Handling
- XII. Passing Parameters by Value and by Reference
- XIII. Principles of Testing and Designing Test Data

All topics are covered in both the lecture and lab parts of the course.

Assignment:

Lecture-Related Assignments:

- 1. Read approximately 30 pages per week
- 2. Complete 2-8 examinations including final exam

Lab-Related Assignments:

1. Complete 10-15 programming assignments, with documentation, using the C++ programming language

Methods of Evaluation/Basis of Grade:

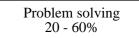
Writing: Assessment tools that demonstrate writing skills and/or require students to select, organize and explain ideas in writing.

Written program documentation

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

Programming assignments

Writing 10 - 20%



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

F	
None	Skill Demonstrations 0 - 0%
Exams: All forms of formal testing, other than skill performance exams.	
Exams, Final Exam: (Multiple choice, true/false, matching items, completion, programming problems)	Exams 20 - 60%
Other: Includes any assessment tools that do not logically fit into the above categories.	
None	Other Category 0 - 0%

Representative Textbooks and Materials:

Starting Out with C++ From Control Structures through Objects. 9th ed. Gaddis, Tony. Pearson. 2017

0 - 0%