

**SADDLEBACK
COLLEGE**

Saddleback College

Course Outline

CS 1A - INTRODUCTION TO COMPUTER SCIENCE I

Catalog Id 192025.00

State Id CCC000591127

Division/School Science, Technology, Engineering and Math

Department Computer Science

Program CMPSCI - Computer Science

Subject Computer Science

History and Status

Course Status

Active (Fully Approved)

Course Originator Perez, Larry

- Tech Review Approval
07/20/2022
- Division Approval
07/20/2022
- Curriculum Committee Approval
07/20/2022
- Board of Trustees
08/29/2022
- State Approval
03/07/2018
- Technical Change Date
10/25/2024

Technical Change Comment

2/12/07-SAM changed fr "D"-----5-6-04-add lab fee fr CS 999; 3/7/18- fr CCC000546314 to CCC000591127; 12/12/18-chngd mtrl fee fr 0 to 6; 10/25/24-added C-ID descriptor

Brief Description

Short Title INTRO TO COMP SCI I

Full Title INTRODUCTION TO COMPUTER SCIENCE I

Catalog Description

The first course in the Introduction to Computer Science series. Designed as a hands-on introduction to the field of computer science for students preparing to enter the field or who are looking for a general interest course. Includes an overview of the history of computers, their hardware, and systems software. Emphasizes algorithm development using modern design methodologies and programming concepts up to and including looping algorithms, functions, and arrays. Programs implemented in a contemporary high-level-language currently C++.

Course Requisites

Course Functions

Course Prior To Y - Not applicable

Course Classification Y - Credit Course

Degree Transfer Applicability

CB04: Credit Status D - Credit - Degree Applicable

Proposed Transfer Types

Acceptable to CSU, UC or Private

Explanation

General Education

UC Approval Date

CSU Approval Date

GE Approval Date

IGETC Approval Date

Local GE Approval Date

Comparable Irvine Course(s)

Comparable Transfer Courses

1. Introduction to Programming

Course ID

CPSC 120

Articulation College System

CSU

4-yr Institution

CSU Fullerton

2. Introduction to Programming

Course ID

I&C SCI 31

Articulation College System

UC

4-yr Institution

UC Irvine

SC/IVC Code Yes

- NA - Not Applicable

Rationale

Approval Date

Cal-GETC No

CSU GE Yes

- TR - Transferable as an elective-does not fit GE pattern

Approval Date

IGETC Yes

- NA - Not Applicable

Approval Date

UC Transferable Course L - UC credit limitations (see UC list)

General Education Status Y - Not Applicable

TOP Code 070600 - Computer Science (Transfer)

SAM Code E - Non-Occupational

CAN Code

C-ID

1. COMP 122

Course Options

Grading Option Letter Grade or Pass/No Pass

Open Entry No

This course is variable No

Repeatable No

Is this a cross listed course? No

Course Values

Method of Instruction Lecture/Lab Combination

Maximum Enrollment 35

Average Enrollment 30

Maximum WSCH

175.000

Average WSCH

150.000

Total Min Units/Hours Calculation

	Lecture	Lab	Learn Ctr	Total
Weekly Faculty Contact Hours	3.00	2.00	0.00	5.00
Total Contact Hours	49.80	33.20	0.00	83.00
Lecture Hour Equivalent	3.00	1.67	0.00	4.67
Full Time Equivalent Faculty	20.00	11.13	0.00	31.13
Units	3.00	0.50	0.00	3.50
Outside of Class Hours				99.60
Total Student Learning Hours				182.60

Schedule Description

Exploration of computer hardware and software. Emphasis on problem solving and programming in C++.
Provides necessary foundation for CS 1B.

Course Content - Topics Covered

Lecture Topics

1. Overview of computer hardware and software
2. Overview of the Von Neumann architecture
3. History of the development of computing machinery
4. Design, implement, test, and debug simple programs in C++
5. Phases of Software Development Lifecycle
6. Problem Solving approaches
7. Overview of operating systems and compilers
8. Algorithm development, flowcharts and pseudocode
9. Algorithms using basic control logic structures: sequence, selection, repetition (loop) structures, and functions.
10. Diagnostics - testing and debugging
11. Documentation
12. Brief history of C/C++
13. Basic syntax and semantics of C++
14. Variables, types, expressions, and assignment
15. Iteration
16. Coding conventions
17. Data types, variables, expressions, sequential processing
18. File I/O: Files and streams; sequential access files
19. Error handling
20. Principles of testing and designing test data
21. Programming in C++
22. Binary and hexadecimal number systems and conversions
23. Introduction to Functions
24. Introduction to Arrays
25. Best Practices in Software Development
26. Style and programming documentation

Lab/Learning Center Content

Through lab experiences, students will practice skills related to the course content in the following areas:

1. Basic concepts of computer functional unit - hardware and software.
2. Introduction to Networking.
3. General computer organization and information representation.
4. Programming fundamentals.
5. Basic software engineering and debugging.
6. Algorithm design and problem-solving techniques.
7. Introduction to programming using a high level language (C++).
8. Basic data type, assignment, arithmetic.
9. Flow control: selection, logical operators.
10. Character strings.
11. Basic sequential and direct access files.
12. Flowcharts and pseudocode design, and editors.
13. Using compilers, and IDEs for first programs.
14. Programs using simple arithmetic expressions.
15. Programs using basic data types.
16. Formatted I/O and arrays.
17. Selection, multiway selection, and logical operators.
18. Pre- and post-test repetition structures.
19. Functions, call by value, call by reference.
20. String processing.
21. Design, implement, test, and debug a program that use basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions
22. Describe the mechanics of parameter passing
23. Introduction to Pair Programming

Course Content - Learning Objectives

Students participating in this class will:

1. Describe the mechanics of parameter passing. Describe various data representations.
Describe the software development life-cycle. Explain the function of the statements that form
2. the basis of a programming language, using examples in several different languages.
Describe the principles of structured programming and be able to describe, design, implement,
3. and test structured programs using currently accepted methodology.
4. Explain what an algorithm is and its importance in computer programming. Represent the
solution to a problem in flowchart and pseudocode form.
5. Describe various data representations.
6. Utilize text editors, compilers and IDEs.

- Break down a problem into the steps necessary for its solution. Translate those solutions into
7. code (created to solve the same problem) in C++.
 8. Demonstrate number system conversion to and from binary, decimal and hexadecimal.
 9. Discuss basic fundamental units of digital computers.
 10. Describe computer organization and operating system features.
 11. Analyze and design efficient algorithms for problem solving.
 12. Utilize appropriate data types and structures.
 13. Write, organize and assemble program documentation.
 14. Create correct code, and debug simple errors in C++.
 15. Define terms relating to computers and computer science.
 16. Describe the historical development of computing machinery.
 17. Create a worksheet using electronic spreadsheet software.

Course Content - Methods of Evaluation

Evaluation of the student will be based upon the following items:

Writing Assignments

- short answers

Description

- laboratory reports

Description

- Other (please describe)

Description

Students will demonstrate their understanding of computing systems, the history of computers, and software development as well as their underlying knowledge of how programming constructs work.

Problem Solving Demonstrations

- exams

Description

- quizzes

Description

- homework problems

Description

- laboratory report(s)

Description

- Other (please describe)

Description

These skills will be demonstrated through labs and assignments. Students will demonstrate their ability to develop algorithms using digagramming tools and translate algorithms to code utilizing an IDE.

Skill Demonstrations

- performance (exam)

Description

- Other (please describe)

Description

Through assignments and lab students will demonstrate their ability to write algorithms and develop code using the principles discussed in class.

Examinations

- multiple choice, true/false

Description

- completion

Description

- Other (please describe)

Description

Students will demonstrate their understanding of computing concepts through T/F and multiple choice questions on an exam as well as open response questions.

Other

- Other (please describe)

Description

Instructor-prepared examinations, written and oral assignments requiring students to interpret, analyze, apply, and solve problems using the skills and concepts included in the topics covered in this course.

Course Content - In and Out-of-Class Assignments

Typical Reading Assignments

College-level text

Typical Writing Assignments

1. Describe the main thesis of a selected article in a computer-related publication. 2. Describe the decision- making process used in the design of an original document, spreadsheet, and database using application software. 3. Write fifteen to twenty sections of code in a high level programming language.

Typical Oral Assignments

Students will participate in weekly class discussions and three or four small-group discussions during the

semester.

Typical Other Assignments

Course Content - Other Requirements

Textbooks / Supplies

D.S. Malik , C++ Programming: Program Design Including Data Structures, 8 Ed. Cengage Learning . 2018

Material Fee 0.00

Requisite Validation
