CS 007A: COMPUTER SCIENCE I

Originator
mflora

Justification / Rationale
Periodic Update.

Effective Term
Fall 2022

Credit Status
Credit - Degree Applicable

Subject
CS - Computer Science

Course Number
007A

Full Course Title
Computer Science I

Short Title
COMPUTER SCIENCE I

Discipline

Disciplines List
Computer Science

Modality
Face-to-Face
100% Online
Hybrid

Catalog Description
This course is an introduction to computer programming and is designed primarily for computer science and related transfer majors. Its main objective is to teach principles and practices of computer science, but students will also engage in problem solving using the C++ programming language. Topics include structured procedural programming with program control structures (sequence, selection, iteration), modular program structures (functions and parameter passing), data types (primitive types, arrays, files and structures) and an intro to object-oriented programming.

Schedule Description
An introduction to computer programming using the C++ language for students with no programming experience. Prerequisite: MATH 005

Lecture Units
3

Lecture Semester Hours
54

Lab Units
1

Lab Semester Hours
54

In-class Hours
108

Out-of-class Hours
108

Total Course Units
4

Total Semester Hours
216

Prerequisite Course(s)
MATH 005

Required Text and Other Instructional Materials

Resource Type
Book

Open Educational Resource
No

Author
Gaddis, Tony

Title
Starting Out with C++ from Control Structures to Objects

Edition
9th

Publisher
Pearson

Year
2018

College Level
Yes

ISBN #
9780134498379

Resource Type
Book

Open Educational Resource
No

Author
Vahid, Frank

Title
Programming in C++

Publisher
zyBooks

Year
2021

College Level
Yes

Class Size Maximum
28

Entrance Skills
Demonstrate proficiency with polynomial analysis including the Fundamental Theorem of Algebra and its implications.

Requisite Course Objectives
MATH 005-Use the concept of a function by identifying and describing a function graphically, numerically and algebraically.
MATH 005-Apply the six basic transformations of functions to graph translated functions, including the quadratic functions.

Entrance Skills
Demonstrate proficiency with analysis of trigonometry functions, including amplitude, periodicity, phase shifts, and basic identities.

Requisite Course Objectives
MATH 005-Graph the six trigonometric functions and demonstrate the ability to predict the corresponding graphic behavior of changes in parameters that modify amplitude, period, and phase.
MATH 005-Use trigonometric functions to model periodic behavior.

Entrance Skills
Demonstrate proficiency with a wide variety of mathematical relations, including rational functions, root functions, symmetric functions, and the quadratic relations used in modeling shifted conics.

Requisite Course Objectives
MATH 005-Use the concept of a function by identifying and describing a function graphically, numerically and algebraically.
MATH 005-Determine when a function has an inverse (one to one functions) and find the inverse function graphically or algebraically.
MATH 005-Form new functions through addition, subtraction, multiplication, division and composition.

Entrance Skills
Solve word problems leading to systems of linear and/or non-linear equations in 2 or more variables using methods of elimination and substitution.

Requisite Course Objectives
MATH 005-Represent a word problem (especially a geometric problem) with a function, including the use of functions to model real world applications.

Course Content

I. Programming Fundamentals (PF) PF1. Fundamental programming constructs

Topics

1. Basic syntax and semantics of a higher-level language
2. Variables, types, expressions, and assignment
3. Simple I/O
4. Conditional and iterative control structures
5. Functions and parameter passing
6. Structured decomposition

PF2. Algorithms and problem-solving

Topics

1. Problem-solving strategies
2. The role of algorithms in the problem-solving process
3. Implementation strategies for algorithms
4. Debugging strategies
5. The concept and properties of algorithms

II. Programming Languages (PL) PL1. Overview of programming languages

Topics

1. History of programming languages
2. Brief survey of programming paradigms

3. Procedural languages
4. Object-oriented languages

PL4. Declarations and types

Topics

1. The conception of types as a set of values together with a set of operations Declaration models (binding, visibility, scope, and lifetime)
2. Overview of type-checking

Lab Content

1. Complete programming assignments incorporating design elements and code.
2. Consult with the teacher and classmates in small groups to tackle special programming tasks that arise as part of (1), above.

Course Objectives

	Objectives
Objective 1	Demonstrate effective use of a program development environment by entering/editing and executing programs.
Objective 2	Demonstrate mastery of the conception of types as a set of values together with a set of operations on those values.
Objective 3	Use the basic syntax and semantics of C++ to declare and define variables of specific types, and to form expressions, and assign these expressions to other variables of compatible types using simple Input/Output with conditional and iterative control structures.
Objective 4	Design functions using structured decomposition/modularization and pseudo-code with the "check/expect" model of development that checks that all input expressions evaluate to the value of the expected output expression.
Objective 5	Synthesize problem-solving strategies to design algorithms implementing step-wise refinement for improving algorithmic efficiency and correctness by debugging.
Objective 6	Interpret declaration models and the binding, visibility, scope, and lifetime of variables, especially by distinguishing "pass by value" and "pass by reference" function parameters.
Objective 7	Compare and contrast features of different programming languages by conducting a brief survey of programming paradigms: procedural, object-oriented and functional.
Objective 8	Employ with deliberate effect scalar data including int, double, char, boolean and string types in one and two-dimensional arrays.
Objective 9	Adhere to the Association of Computing Machine's Code of Ethics and Professional Conduct principles of contributing to human well-being, avoiding harm and behaving in an honest and trust-worthy manner and to maintain high standards of professional competence, conduct, and ethical practice.
Objective 10	Design iterative loops through the use of flow charts, pseudo code, or tracing of code.
Objective 11	Determine validity of a Boolean Logic expression.
Objective 12	Design and implement various types of loops appropriate to a given problem.
Objective 13	Design and implement programs that use complex Boolean expressions in conditional statements, as well as, in loop conditions.
Objective 14	Convert decimal numbers to binary form and binary numbers to a decimal.
Objective 15	Determine how the size (number of bytes) of a variable determines its range.

Student Learning Outcomes

	Upon satisfactory completion of this course, students will be able to:
Outcome 1	Describe the functionality of the basic file input and output operations using file streams. Write and debug the code that opens a file, checks for an open file error, reads from the file and checks for the end-of-file condition.
Outcome 2	Write and predict the results of code with numeric and Boolean expressions, if-statements and loops, including nested control structures.
Outcome 3	Design, write and predict the results of code using functions that have parameters (both call by reference and call by value) and return values.
Outcome 4	Design and develop code to process arrays.

Methods of Instruction

Method	Please provide a description or examples of how each instructional method will be used in this course.
Laboratory	Students will practice developing the design principles introduced in lecture by writing programs that solve problems of varying difficulty. Typically, students may be assigned to work either individually or in small groups to address the problem of writing code to accept input in a specific format and analyze that input produce a desired output.
Lecture	Programming design practices and principles are introduced in concept and by example.
Collaborative/Team	Take turns role-playing as designer/tester/developer in solving programming challenges to produce software meeting prescribed input/output specification.

Methods of Evaluation

Method	Please provide a description or examples of how each evaluation method will be used in this course.	Type of Assignment
Mid-term and final evaluations	There will be a midterm and a final exam, generally in written format, but this may be combined with some computer work. (4 hrs)	In Class Only
Tests/Quizzes/Examinations	There will be a sequence of short quizzes to gauge student understanding of new concepts. (3 hrs)	In and Out of Class
Group activity participation/observation	Three or more major projects encompassing at least two weeks for development of complex solutions to complex tasks. Typical problems involve assignments such as solving triangles, investigating the behavior of generalizations of the Collatz conjecture, generalizing the Babylonian algorithm for computing nth roots in the complex plane, exploring knight's tours on toroidal chess board, simulating the Game of Life and writing a calculator that will parse arithmetic and/or algebraic expressions. (10 hrs)	In and Out of Class
Computational/problem-solving evaluations	Branching and looping structures are employed in algorithms such as that logistic iteration x(n+1) = cx(n)(1-x(n)), testing for bifurcation values, etc. (1 hr/wk)	In and Out of Class
Laboratory projects	Often these will require students to solve problems from the ICPC (International Collegiate Programming Contest) while using concepts introduced in lecture. (3 hrs/wk)	In Class Only

Assignments

Other In-class Assignments

1. Participate in discussion.
2. Develop original programs to solve given problems.

Other Out-of-class Assignments

1. Read the text. (2 hrs/wk)
2. Write descriptions of programs in pseudocode. (0.5 hrs/wk)
3. Finish unfinished lab work. (2 hrs/wk)
4. Take quizzes. (0.5 hrs/wk)

Grade Methods

Letter Grade Only

Distance Education Checklist

Include the percentage of online and on-campus instruction you anticipate.

Online %
100
On-campus %
0

What will you be doing in the face-to-face sections of your course that necessitates a hybrid delivery vs a fully online delivery?
Although the course can be offered entirely online, it may also be offered hybrid to take advantage of collaboration activities that are more suited to in-person interaction.
Examinations can be given in a controlled location.

Lab Courses

How will the lab component of your course be differentiated from the lecture component of the course?
Lab assignments involve more interaction. For example, they may require students collaborate with a classmate, utilize a tutoring resource, or interview someone who is not part of the course.

From the COR list, what activities are specified as lab, and how will those be monitored by the instructor?
Lab activities are discussions and assignments that involve solving problems or exploring concepts with other students, with people not part of the course, or under the guidance of the professor or instructional support assistant. Discussions and other assignments that are completed in Canvas are monitored and evaluated by the professor. Assignments that do not take place in Canvas are evaluated by the professor based on write-ups (which may include summaries and feedback from the participants). Anonymous and non-anonymous feedback opportunities will be available to students to allow the professor further monitor effectiveness and appropriateness of activities that take place somewhere other than on the course LMS.

How will you assess the online delivery of lab activities?
Reports and other forms of write-ups will be submitted on the course LMS for evaluation and feedback.

Instructional Materials and Resources

If you use any other technologies in addition to the college LMS, what other technologies will you use and how are you ensuring student data security?
Depending on the textbook used, the professor may choose to use Pearson MyLab and Mastering, zyBooks, WebAssign, Replit, or GitHub. All of these are considered to be safe for use in education for both faculty and students. All can also be integrated with the college LMS (Canvas), which decreases the amount of times students will need to sign-in-and-out of accounts and open them up to data breaches.

If used, explain how specific materials and resources outside the LMS will be used to enhance student learning.
Professors who choose to use Pearson MyLab and Mastering, zyBooks, WebAssign, Replit, or GitHub do so in order to assign pre-written or instructor-created problems that are more complicated than those that can be created in Canvas while still receiving instantaneous feedback.

Effective Student/Faculty Contact

Which of the following methods of regular, timely, and effective student/faculty contact will be used in this course?

Within Course Management System:
Chat room/instant messaging
Discussion forums with substantive instructor participation
Online quizzes and examinations
Private messages
Regular virtual office hours
Timely feedback and return of student work as specified in the syllabus
Weekly announcements

External to Course Management System:
Direct e-mail
Posted audio/video (including YouTube, 3cmediasolutions, etc.)
Synchronous audio/video
Telephone contact/voicemail

For hybrid courses:
Scheduled Face-to-Face group or individual meetings

Briefly discuss how the selected strategies above will be used to maintain Regular Effective Contact in the course.

Faculty will regularly contact students individually and as a group through Canvas messages and/or COD email. Students will also receive regular announcements with information about the course, COD as a whole, or other relevant information.
In discussions and through other lab assignments, students will communicate with each other and their professor regularly and frequently.

If interacting with students outside the LMS, explain how additional interactions with students outside the LMS will enhance student learning.

Students may prefer to contact their professor via email or on the phone, which allows for an improved experience for those who communicate better in those contexts. The professor may direct students to access free supplemental resources as well.

Other Information

Comparable Transfer Course Information

University System
CSU
Campus
CSU San Bernardino

Course Number
CSE 2010
Course Title
Computer Science I

Catalog Year
2021

University System
UC
Campus
UC Santa Barbara

Course Number
CMPSC 16
Course Title
Problem Solving with Computers I

Catalog Year
2021

University System
UC
Campus
UC Riverside

Course Number
CS 10A
Course Title
C++ Programming I

Catalog Year
2021

University System
UC

Campus
CSU Los Angeles

Course Number
CS 2011

Course Title
Introduction to Programming I

Catalog Year
2021

MIS Course Data

CIP Code
11.0701 - Computer Science.

TOP Code
070600 - Computer Science (transfer)

SAM Code
E - Non-Occupational

Basic Skills Status
Not Basic Skills

Prior College Level
Not applicable

Cooperative Work Experience
Not a Coop Course

Course Classification Status
Credit Course

Approved Special Class
Not special class

Noncredit Category
Not Applicable, Credit Course

Funding Agency Category
Not Applicable

Program Status
Program Applicable

Transfer Status
Transferable to both UC and CSU

General Education Status
Y = Not applicable

Support Course Status
N = Course is not a support course

C-ID
COMP 122

Allow Audit
No

Repeatability
No

Materials Fee
No

Additional Fees?
No

Approvals

Curriculum Committee Approval Date
11/18/2021

Academic Senate Approval Date
12/9/2021

Board of Trustees Approval Date
1/21/2022

Chancellor's Office Approval Date
6/03/2020

Course Control Number
CCC000523776

Programs referencing this course
Engineering AS Degree (http://catalog.collegeofthedesert.eduundefined/?key=24)
Liberal Arts: Business and Technology AA Degree (http://catalog.collegeofthedesert.eduundefined/?key=27)
Mass Communication A.A. Degree (http://catalog.collegeofthedesert.eduundefined/?key=273)
Liberal Arts: Math and Science AA Degree (http://catalog.collegeofthedesert.eduundefined/?key=29)
Computer Science AS-T Degree (http://catalog.collegeofthedesert.eduundefined/?key=35)
Mathematics AS Degree (http://catalog.collegeofthedesert.eduundefined/?key=68)