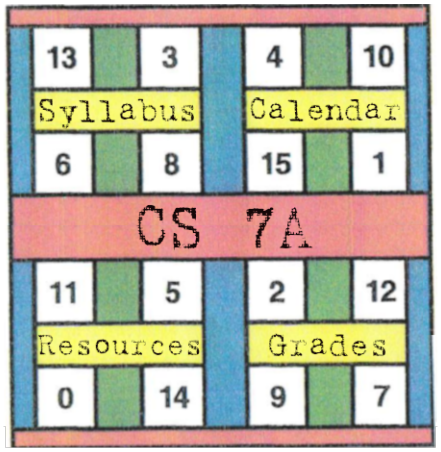


CS 7A - Section 3496 - Fall, 2019

Computer Science I

TR 9:30-12:35, DM24

ghagopian@collegeofthedesert.edu
geofhagopian.net
Math 12
Office Hours: MW 4:30-5:15, TR 8:30-9:30, or by appointment.
(760) 776-7223

Course Description: This introductory course in computer programming and computer science is designed primarily for computer science and related transfer majors. Its primary objective is to teach fundamentals of computer science and 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.
Note: No prior knowledge of computer programming is required.
Prerequisite: MATH 005 (Trigonometry) with a minimum grade of C.
Text: *Programming Principles and Practice Using C++* (2nd ed). ISBN: 978-0-321-99278-9,by Bjarne Stroustrup (Addison-Wesley, 2014).

Course Objectives:

At the completion of this course, students will be able to:

1. Demonstrate the grammar, punctuation, and vocabulary of C++ by composing original programs to meet specified design conditions.
2. Demonstrate good program design principles to do problem solving, creating programs that are correct, efficient, and easily read, modified, and repaired.
3. Describe the sequence of steps to go through, in designing, writing, testing and debugging a C++ program.
4. Synthesize application-specific knowledge (e.g. to write a program that draws geometric shapes on the screen, you have to know something about geometry.)

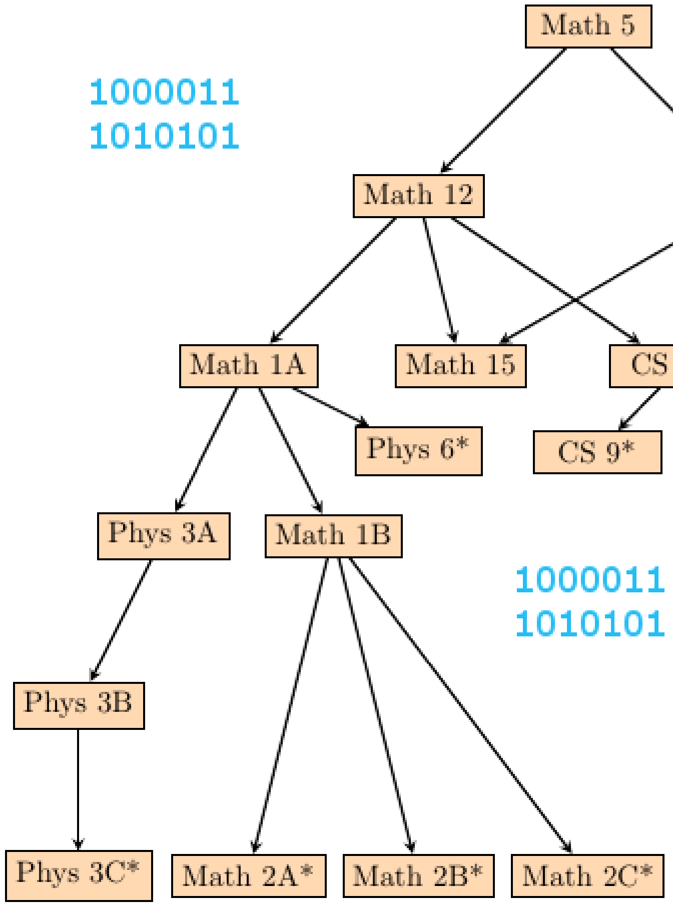
Grade Distribution:

Quizzes	10%
Midterms	25%
Final Exam	15%
Projects	50%

Letter Grade Distribution:

$\geq$ 90.00	A	70.00 - 79.99	C
80.00 - 89.99	B	60.00 - 69.99	D
.	.	$\leq$ 59.99	F

Computer Science Program Prerequisites Map



1000011
1010101

Note: Both Math 12 and CS 7A are required for Math15 and CS7B.

Math 5: Trigonometry
Math 12: Precalculus
CS7A: Computer Science I
CS7B: Computer Science II
CS8: Computer Architecture
Math 15: Discrete Math
Math 1A: Calculus I
Math 1B: Calculus II
Physics 3A: Engineering Physics I
Physics 3B: Engineering Physics II
* Choose at least 2 of these courses.

Course Policies:

- **Calendar:** Refer regularly to the calendar at to monitor the progress of lecture, homework, quizzes, tests, and projects.

- **Homework:** Homework will involve close reading (and re-reading) course materials, working exercises from the text, and developing/completing projects.

- **Quizzes:** There will be a sequence of frequent quizzes to gauge your understanding of the homework.

- **Exams:** There will be 2 midterms exams and a final exam at the end of the term.

- **Projects:** A large part (half) of your evaluation in this course is based our your ability to complete various projects in a timely manner. Some of these will be individual, and for some you will work in pairs.

  You'll want to start the projects as soon as possible and, when you get stuck on something, bring your questions to class and/or see me in my office.

- **Lecture vs. Lab and Computers**

  - Learning is best accomplished by combining the processes of listening, looking and writing–each occupies a relatively independent neural network, the whole of which is greater than the sum. Therefore it is recommended that you take notes during lecture.
  - Access to a computer with an Visual Studio 2017 (Community Edition) IDE outside of class is essential–you will need this to be able to modify code and test your changes in a timely manner. See the Resources Guide.
  - To support your foundation of computer science theory and good programming habits, this course involves writing and rewriting C++ code to accomplish various computing tasks. The programs you will be asked to write will be relatively short and simple. In the first part of the course, programs will be only console-oriented (input and output involved in the program will be limited to ascii text data, not graphics), while extending into the realm of GUI (graphics user interface) and programs interfacing with a simple graphics environment in the latter part of the course.

    The programming assignments are designed to be interesting, challenging and to build incrementally on your existing skills. A big part of writing code involves debugging - that is, rewriting your code to eliminate various kinds of "bugs", improving the efficiency of algorithms and/or improving the way the code provides flexibility in interaction with other code and coders. This can be quite time-consuming and, well, frustrating at times. That's normal. You are encouraged to consult with me and your classmates–but ultimately you must invent your own code and to understand it for yourself.

    If your work duplicates in whole or in significant parts the work of someone else, both will receive a grade of 0. If there is doubt about authorship, you may be required to defend your work. If you submit work copied from the internet, that's plagiarism. According to the Student Conduct Code in the College Catalog, plagiarism "shall constitute good cause for discipline, including but not limited to the removal, suspension or expulsion of a student."

    The programming projects are listed on the calendar page. These should be emailed to my cod email address attaching the file named using the format ⟨your initials⟩_⟨assignment name⟩ and the file type either .txt or .cpp depending on whether you are submitting plain text or C++ code. It his helpful to zip multiple files together in a bundle, in which case the file type will be .zip. This will build a portfolio which I will evaluate periodically. Your programming work will be evaluated according to the rubric shown on the next page

- **Attendance and Absences** Attendance is expected and will be noted. If you're not there, you missed it. Students are responsible for all missed work, regardless of the reason for absence. It is also the absentee's responsibility to get all missing notes or materials.

Academic Honesty Policy

In addition to skills and knowledge, College of the Desert aims to teach students appropriate ethical and professional standards of conduct. The college catalog specifies that students are expected to "Integrate universally accepted values such as honesty, responsibility, respect, fairness, courage and compassion into judgments and decision-making." and that, "Students are expected to act in an honest and trustworthy manner. Work performed on examinations or other forms of evaluation must represent an individual's own work, knowledge and experience of the subject matter. Students are expected to follow the classroom rules established by each instructor." Any attempt to deceive a faculty member or to help another student to do so will be considered a violation of this standard.

Trait	Exceptional	Acceptable	Fledgling	Unsatisfactory
Specifications	The program works and meets all of the specifications	The program works and produces the correct results and displays them correctly. It also meets most of the other specifications	The program produces correct results but does not display them correctly.	The program is producing incorrect results.
Readability	The code is exceptionally well organized and very easy to follow.	The code is fairly easy to read.	The code is readable only by someone who knows what it is supposed to be doing.	The code is poorly organized and very difficult to read.
Reusability	The code could be reused as a while or each routine could be reused.	Most of the code could be reused in other programs.	Some parts of the code could be reused in other programs.	The code is not organized for reusability.
Documentation	The documentation is well written and clearly explains what the code is accomplishing and how.	The documentation consists of embedded comment and some simple header documentation that is somewhat useful in understanding the code.	The documentation is simply comments embedded in the code with some simple header comments separating routines.	The documentation is simply comments embedded in the code and does not help the reader understand the code.
Delivery	The program was delivered on time.	The program was delivered within a week of the due date.	The code was within 2 weeks of the due date.	The code was more than 2 weeks overdue.
Efficiency	The code is extremely efficient without sacrificing readability and understanding.	The code is fairly efficient without sacrificing readability and understanding.	The code is brute force and unnecessarily long.	The code is huge and appears to be patched together.