

Syllabus

Course: Introduction to Computer Science I (CSE 201)
School: CSUSB
Quarter: Fall 2013
Lecture: Mon/Wed 12:00 - 1:15 JB 113
Lab 1: Mon 1:30 - 3:20 JB 359
Lab 2: Wed 1:30 - 3:20 JB 359
Prerequisite: Some exposure to programming
Textbook: Big C++, Second edition by Kay Horstmann and Timothy Budd
(or other equivalent book)
Instructor: David Turner
Office Hours: 3:30 - 5:30 Mon/Wed

Course Format

You should carefully study the first 6 chapters of the textbook or equivalent introductory book on C++. During our weekly lab sessions, you will perform activities to improve your understanding of the material presented in lecture and in the textbook. I will assign programming problems that you will complete inside and outside of lab. There will be quizzes and final exam.

All required work is detailed in the course schedule, published on the Web. I will announce in class or by email changes to required work, points, deadlines and quiz dates.

I encourage you to collaborate with other students to complete the labs and assignments. However, you must submit assigned work individually and you are required to understand what you submit. I will use quizzes and a final exam to assess your understanding of submitted work for labs and assignments.

There are 2 ways to submit labs and assignments: by email or through a revision control system called Git. I will provide instruction on how to use Git for this purpose.

Course Goals

- Students will learn how to write programs in the C++ language.
- Students will improve computational thinking skills, which can be used to solve a wide range of problems.

Learning Objectives

- Learn how to devise algorithms that solve computing problems.
- Learn about primitive data types.
- Learn about user-defined data types.
- Learn about flow control constructs.
- Learn about functions.
- Learn about arrays and vectors.
- Learn about pointers (into computer memory).
- Learn how to compile code into executable files.
- Learn about version control systems.
- Learn how to use the Linux command line interface.

Labs/Assignments

In this course, you will complete a sequence of assignments and labs. The assignments involve research, programming and problem solving. The labs also involve research, programming and problem solving, but I will generally show you their solutions in advance of their deadlines.

The work in this course can be completed using the lab computers in JB 359 and JB 358. The work can also be completed outside the labs on your personal computer in any of the 3 common operating system environments: Linux, Windows, and OS X. You have 2 choices when working from home: work on a lab machine through a remote connection, or work completely on your own computer. If you work on your own computer, you will need to set up a C++ development environment.

Programs submitted with compilation errors receive no credit.

Programs that are incorrect or do not solve the stated problem will lose some or all points.

Work that is submitted late will lose some or all points.

If you submit copies of other people's work, you will lose all points for that work. However, if you work with a partner, you can submit identical programs, but you need to refer to your partner by name in a comment at the top of your source code file.

Writing a program to produce required behavior is not good enough for a full score in this class: you must also write code that is readable by humans. Program readability is important because real-world programs are read over and over again in the process of fixing bugs and adding new functionality. Program readability will be evaluated according to the following set of criteria.

Table 1: Readability Criteria

| Criterion | Description |
|-----------------------------|---|
| Organization | Is source code well-organized? |
| Cleanliness | Have unnecessary variables and logic been removed from the code? |
| Logical indentation | Does indentation show logical structure? |
| Consistent indentation | Does the indentation follow a consistent policy? |
| Logical spacing | Does the spacing show logical structure? |
| Consistent spacing | Does the spacing follow a consistent policy? |
| Separation of concerns | Are concerns separated when possible? |
| Responsibilities | Are the responsibilities of functions and classes clear and consistent? |
| Expressive and clear naming | Does the code contain names for variables, functions and classes that clearly express their purpose in the program? |
| Necessary comments | Are comments included when needed? |
| Unnecessary comments | Are superfluous comments omitted? |

Grading

Labs, assignments, quizzes and final exam have point values, which are shown on the course schedule. Your percentage score will be computed by dividing the total of all points earned by the total possible points. The normal scale will be used to assign a letter grade.

Table 2:
Grading Scale

| Percent | Grade |
|----------------|--------------|
| 90-100 | A |
| 80-89 | B |
| 70-79 | C |
| 60-69 | D |
| 0-59 | F |

Students with Disabilities

If you are in need of an accommodation for a disability in order to participate in this class, please let me know as soon as possible, and also contact Services to Students with Disabilities at UH-183, (909)537-5238. You are advised to establish a buddy system and alternate in the class if you require assistance in the event of an emergency. Individuals with disabilities should prepare for an emergency ahead of time by instructing a classmate and the instructor.

Academic Regulations and Procedures

See the CSUSB Bulletin of Courses for the University's policies on course withdrawal, cheating, and plagiarism.

Computer Science and Engineering Club

The CSUSB Computer Science and Engineering Club is a student-run organization that uses a combination of email and campus meetings to plan events, ask and answer technical questions, post job and internship openings, and discuss other topics of interest to computing majors at CSUSB. Club-sponsored events include seminars, workshops, tutoring and fun activities.