

CIS 22B INTERMEDIATE PROGRAMMING METHODOLOGIES IN C++

Instructor	Victor Yu
Meeting Hours	Lectures: Monday ~ Thursday: 12:30 pm-1:30 pm Lab Hours (optional): Monday ~ Thursday: 1:40pm – 2:20pm <u>Online</u> : Tuesday, Thursday: 8:30PM-9:00PM
Email	yulinyun@fhda.edu
Course Site	http://elearning.ebookriter.com (Enrollment key: HackC++@2)
Self-Test	MyProgrammingLab (Section Access Code: DEAN-19788-RKFV-25)

COURSE DESCRIPTION

This course is an introduction to computer programming. Its primary objective is to teach problem solving using the C++ programming language. Emphasis will be placed on structured procedural programming with an introduction to object-oriented programming. This course is designed primarily for computer science and related transfer majors.

Upon successful completion of this course, students should be able to:

- Read, analyze and explain intermediate level C++ programs and their efficiency.
- Design solutions for intermediate level problems using appropriate design methodology incorporating intermediate programming constructs including structures and objects.
- Create algorithms, code, document, debug, and test intermediate level C++ programs.

REQUIRED MATERIALS

Starting Out with C++: From Control Structures through Objects, 8th Edition, By Tony Gaddis, 8th Edition, 2014. Publisher: Addison-Wesley, ISBN-13 976-0376939-5 ISBN-10: 0-13-376939-9 Available for purchase or rent on [Amazon](#),

COURSE REQUIREMENTS

Attendance	Your attendance is expected in all lectures, because some of the materials presented in class may not be in the textbook. If you miss any class, you are still responsible for completing all work assigned in this class in a timely fashion. You are expected to do the assigned readings before each session and to come prepared for the discussion. Lab attendance, however, is optional. If you have a computer at home with a C++ development environment installed, you may choose to work on your labs from there.
Scholarly Conduct	Discussion and exchange of ideas on assignments are strongly encouraged. However, each person is expected to complete his/her own computer work. Identical solutions will be given a zero grade. Copying or cheating during an exam will result in a zero being assigned to the test grade for both parties.
Participation & Assignments	Assignments are important component of the course. You will not learn by attending lecture in lieu of completing assignments. Assignments consist of: <ul style="list-style-type: none"> • Required reading assignments • Labs

- Exams** There will be 1 midterm and 1 final exam.
- Make up for the midterms rarely allowed, unless for emergency reasons or prior approval. Prior approval must be obtained at least one week before the scheduled exam.
 - The final exam must be taken during the scheduled time, there is no early or late exam taking.
- Both exams are comprehensive.

ACADEMIC RESOURCES AND SUPPORT

Tutoring De Anza's Tutorial Center is located in L47. You may receive tutoring services including drop-in help, regularly scheduled tutoring sessions, and/or online tutoring assistance. These and many other academic support services are part of the Student Success Center: <http://www.deanza.edu/studentsuccess/>

Library Support Many library materials can be accessed from an off-campus computer. Go to the library webpage at <http://www.deanza.edu/library>. You will need either your 14 digit library number, posted on the front of your DASB card or your 8 digit student identification number to verify your student status. See the library webpage for more details.

GRADING

Grading is based on the percentage of the total points obtained:

Participation & Contributions	5%
Labs	50%
Midterm exam	25%
Final exam	25%

A	90-100%
A-	86-89%
B+	80-85%
B	76-79%
B-	70-75%
C+	66-69%
C	60-65%
F	<=59%

Tentative Schedule

<i>Week</i>	<i>Topics</i>	<i>Chapter Readings</i>	<i>Exams</i>
1	Pointers and Dynamic Allocation	Ch 9	
2	C-Strings, C++ string class, Structure, File I/O	Ch 10, 12	
3	Introduction to Classes, Object-oriented Design	Ch 12	Midterm Exam
4	Function Overloading, ADT, Friend Functions, Operator Overloading	Chapter 13, 14	
5	Inheritance and Polymorphism STL	Chapter 15, 16	
6	Linked Lists	Chapter 17	Final Exam