

### Syllabus

**11309 \* CIS -022B-61Y** *Intermediate Programming Methodologies in C++*

**Instructor:** Shu-huar Yeh

**Email** : [yehshuhuar@fhda.edu](mailto:yehshuhuar@fhda.edu) ([Links to an external site.](#))[Links to an external site.](#)

**Office Hour** : No official office hour. Use email and message.

**Class meetings:** AT203 Monday, Wednesday 6:00PM – 9:50PM

**LAB** : ONLINE See the ONLINE LAB schedule and login page

**Midterm** : AT203 Monday, July 23 during the lecture hours

**Final Exam** : AT203 Wednesday, August 8, 2018 6:15PM - 8:15PM

**Prerequisites:** CIS 22A.

**Course Description:** A systematic approach to the design, construction and management of computer programs, emphasizing design, programming style, documentation, testing and debugging techniques. Strings, multidimensional arrays, structures, and classes. Pointers: their use in arrays, parameters and dynamic allocation. Introduction to linked lists. Software engineering and computer science students are the targeted group.

#### Student Learning Outcome Statements (SLO)

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

#### Course Objectives:

Upon completion of the course, the student will

- Know how to construct and develop good intermediate level C++ programs.

- Appreciate well-organized and well-documented programs.
- Have learned the usage of pointers in various ways: arrays, functions, and dynamic memory allocation.
- Have learned how to process text data using C-strings and **string** objects.
- Have learned how to process structured data.
- Have learned how to use classes for problem solving.
- Know various topics about classes: access specifications, constructors, destructors, static members, friends, operator overloading, object conversion, object aggregation, inheritance, and polymorphism.
- Know how to write code that can handle exceptions.
- Know how to enable code reuse through templates.
- Be inspired to learn more about software development technologies.

**Text required:** Gaddis, Tony Standard Version of Starting Out With C++ from Control Structures through Objects(9th Edition), 2018 ISBN: 9780134498379 (8th edition of the same book is also good to use.)

**Attendance policy:** You are expected to attend all lecture and the online lab sessions. If you must be absent from class, arrange with another student to share class notes for that session. You should plan on spending at least another 4 hours per week in the open lab. **If you wish to drop the class, it is your responsibility to do so.** An unauthorized withdrawal from class without following official procedures will result in your being assigned a grade of "F" (or "NC" if you have selected the Credit /No Credit option).

**Scholarly conduct:** In order to be successful in this class you will have to make a commitment to studying, reading the text, doing your homework, writing your lab assignments, attending class, and taking notes. Worthwhile contribution (both in the classroom and the forum) and regular attendance can positively affect the grades. You are expected to do your own work. **Copying or cheating during a test will result in a zero** being assigned for that test. In programming classes, students often confer with one another on approaches to solving the problem: however, your solutions to lab problems must represent your own individual work. Do not copy solutions from any source outside of this class. **Any copied solutions will result in a zero grade for both parties**, and may result in a failing grade. It may also result in dismissal from class. Please check the current Schedule of Classes to learn more about academic integrity, other policies, and Student Standards of Conduct.

**Homework:** Homework is not to be run on the computer (unless you wish to). The purpose of the homework is to help clarify the material for you as we proceed and to prepare you for tests, therefore, you are strongly encouraged to do it.

**Tests:** There will be pop quizzes throughout the course, a midterm and a final. The points you earn from the quizzes are counted as extra credit. There will be **no make-up** for the quizzes you have missed. You must take the midterm and the final exam in the classroom and log in to Canvas using the computers in the classroom. The midterm and final exams are open book, open notes, no cameras, no internet access, no personal devices. You must obtain a user ID to the lab computers for the tests. Seats may be reassigned. Midterm and final dates are shown on the calendar. Final exam papers, if any, will be retained for a period of 90 days from the exam date.

**Laboratory assignments:** You will be given individual lab assignments. All assignments must be turned in on or before the due date. Partial credit will be given for incomplete assignments. For more information on the grading of lab assignments, see Lab Requirements

**Extra-credit** may occasionally be given throughout the course.

**Grading:** 200 points are available (+ up to 6 points extra-credit). Your final grade is based on the percentage of the total you earn; however, if you fail the final exam your final grade will be lowered by at least one letter grade.

Activities	Occurrences	Points
Labs and CodeLab Assignments	See lab assignments	80
Midterm	1	50
Final	1	70
Total		200
Extra (quizzes, etc.)		6

Grading scale:

Percentage	Grade
------------	-------

<b>90 - 103</b>	A-, A, A+
<b>80 - 89.9</b>	B-, B, B+
<b>70 - 79.9</b>	C, C+
<b>60 - 69.9</b>	D
<b>0 - 59.9</b>	F

Both A and A+ require that the programming portion of the final exam is assessed at 90% or above.

### **Useful Links**

Important Dates (i.e., Drop date, etc.): <http://www.deanza.edu/calendar/springdates.html>

Resources On Campus: [Tutorial](#), [EDC](#), [Counseling](#).

Classroom Conduct: [Academic Integrity](#).

[Mutual Respect Policy](#).

[Student Grievance Procedure](#).

[Student Rights & Responsibilities](#).