CIS 22B-02Y – Intermediate Programming Methodologies in C++

Instructor	Manish Goel
Class Hours	TTh: 1:30 pm – 3:20 pm, AT205
	M: 3:30 pm – 4:45 pm ONLINE
Office Hours	MTWTh: 11:00 am – 12:00 pm or by appointment
Phone	(408) 799-9170 – turnaround time can be 24 hours
Email	goelmanish@fhda.edu – this is the best way to reach me

Text *Starting Out with C++: From Control Structures through Objects*
8th edition, by Gaddis   ISBN: 0-133-77877-0

Class website Please log into Catalyst

Course Description A systematic approach to the design, construction and management of computer programs, emphasizing design, programming style, documentation, testing and debugging techniques. Strings, multi-dimensional arrays, structures and classes. Pointers: their use in arrays, parameters and dynamic allocation. Introduction to linked lists.

Prerequisites Computer Information Systems 22A (formerly Computer Information Systems 71B). (Students may receive credit for either {CIS 22A and 22B} or {CIS 27}, but not both.)

Student Learning Outcomes Upon the completion of this course, students will 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 using classes and objects.
- Create algorithms, code, document, debug and test intermediate C++ programs.

Attendance Any student who is a No-Show on first day of class will be dropped.

After the first class, it is *your responsibility to drop the class before the last day to drop.* Otherwise, you will receive an appropriate grade at the end of the quarter.

This hybrid course has 4 lecture / lab hours on campus in addition to online reading and assignments. Regular and punctual attendance is expected during the quarter. Lectures will be the main source of information.

Class Decorum In class, you are expected to pay attention, participate, not conduct personal conversations, and use the computer for class work only. Disruptive behavior is not tolerated, and any student with excessive disruptive behavior will be asked to leave and administrative follow-up may result. On the other hand, worthwhile contribution and regular attendance can positively affect your grade.

Scholarly Conduct Discussion and exchange of ideas on lab assignments are strongly encouraged. However, each person is expected to complete his/her own computer work. **Identical solutions will be given a zero grade to all parties. DO NOT SHARE EITHER SOFT OR HARD COPY OF YOUR CODE WITH ANYONE. Copying or cheating during an exam will result in a zero being assigned to the test grade for both parties and may result in a failing grade. ANY SUCH ACTIVITY WILL BE REPORTED FOR DISCIPLINARY ACTION.**

Lab Assignments   There will be 5 lab assignments – worth 20 points each:

- All labs have to be turned in as a soft copy via Catalyst by their due date.
- Partial credit will be given for incomplete labs based on corresponding grading rubrics.
- Labs turned in after the due date will receive a 20% *per calendar day* penalty.
- Labs turned in more than 5 days after the due date will not be graded.
- Labs could build on prior ones, so missing any labs could be hard to make up.

Exams   There will be 2 short quizzes, 1 midterm and 1 final.

  * All exams are open book, open notes, no electronic devices.

* The quizzes will be multiple choice, fill in the blank questions or code correction questions.

* Midterm and final will include programming questions for which code submission will be required similar to lab assignments.

* You must pass the final exam in order to pass the class.

* Make up for the midterm will be allowed only with proof of emergency reasons or prior approval. Make up exam will be given no later than one week after the mid-term, will be administered after a class session and will have a 25% penalty.

* Final exam must be taken only during the scheduled time – there will be no make up.

Code Lab   CodeLab is an online learning tool and its use is worth 60 points. The questions will be assigned and required to be completed in class. Late submission of solutions will be allowed of up to 24 hours and will incur a 50% penalty.

Project   There will be a project due on the last day of class. This project is to be done in teams of 3 members with work divided equitably between all team members. Details of the project will be provided after the midterm.

Extra Credit   There will be other opportunities to earn extra credit – these will be determined later. You must be present in class to earn the extra credit

Grading   Grading is based on the percentage of the total points obtained as below. All grades will be reflected in Catalyst. There will be no curve.

     Lab assignments: 100 points (5x20 points)
     Quizzes: 40 points (2x20 points)
     Project: 60 points
     Midterm: 50 points
     Final: 50 points
     Total: 300 points

The passing grade at De Anza is a C or higher – for students taking this course for a Pass/No-Pass.

A+: 97-100%	B+: 87-89%	C+: 77-79%	D+: 67-69%	
A : 93-96%	B : 83-86%	C : 70-76%	D : 63-66%	
A- : 90-92%	B- : 80-82%		D- : 60-62%	F : 0-59%

_____

Lab Submission Format

Assignments that do not follow these instructions will receive a penalty of 2 points.

- IMPORTANT - All labs MUST WORK on Microsoft Visual Studio 2013 because they will be graded on MSVC13. If you use Mac or Linux machines or any other Windows compiler, please ensure your labs compile and run on MSVC13 in the ATC lab computers. Code that doesn't run on MSVC13 will earn a 25% penalty.

- Each lab project should be named **"Lab_nn_yourfullname"**. The CPP file inside the project containing your "main" should be name**d "Lab_nn_yourfullname.cpp"**. Other CPP files can be named according to their designed purpose.

- Each assignment must begin with a documentation block with the following information – failure to include the header name block will earn a 10% penalty:

        // CIS 22B
        // Lab nn: One line description of the lab
        // Name:_____

- Assignments will be graded for Correctness, Structure, Style, Clarity and Documentation.

- Labs may require screen shots of your program output. The screen shots should be taken such that the output is clearly readable – remember the grading will take place on a different machine so zoom screen shots to improve readability before submitting.

- Some labs will require file output. Most likely, the format of the file output will be similar to the screen output but will be specified along with the lab.

- All labs should have minimum documentation as follows:
    1. Name block at the top of the code file.
    2. Pseudocode – right after the name block, and at other necessary places in your code.
    3. Comments in-line with the Program listing as necessary to make your code understandable.
    4. Name block in your Program Output.

- Remember to adequately test your program. When test data is specified, output based on the test data is <u>required</u>.

- Remember to collect all necessary files of your work – *it is your responsibility to ensure I have complete assignments*. At a minimum, your output will consist of:
    1. Your code files as *.h and *.cpp files.
    2. Your screen output as a JPG or PNG – it could be one or more files.
    3. If any flowchart is required, it should be in a PDF format with clear readability, whether hand-drawn or created using any drawing application.
    4. If any file output is required, it should be in a *.txt format.
    5. Zip all your files into a single compressed folder of type *.zip – your zip file should be named **"Lab_nn_yourfullname.zip"**.
    6. Only include the required files, do not create any sub-folders inside the zip file – this is especially important for MAC users.
    7. Upload your zip file to the Catalyst link for the corresponding lab only – do not email them to me.
    8. Not following all of the above or submitting labs in any other zip format (*.gz, *.tar, *.7z etc.) will earn a 10% penalty.