

Programming Lab Practicum

CPSC 301-03

Fall 2017

Description & Objectives

Intensive programming covering concepts learned in lower-division courses. Procedural and object oriented design, documentation, arrays, classes, file input/output, recursion, pointers, dynamic variables, data and file structures.

Prerequisites

Prerequisite: CPSC 131; declared major/minor in CPSC, CPEN or CPEI.

Notes

Go to class the first meeting to sign up to take the EPP (Examination in Programming Proficiency). If you do not pass the EPP, the Computer Science Department will assist those with priority to register for CPSC 301 on a space available basis. Please see http://www.fullerton.edu/ecs/cs/_resources/pdf/CS301Info.pdf for more information.

Enrollment restricted to those students who have met the prerequisite(s).

Instructor

Professor William McCarthy

Phone: 657-278-7305 or 213-294-3177

Email: wmccarthy@fullerton.edu

Office: CS 429

Office Hours: Monday & Wednesday 1300-1500

& by appointment. During final exam week, office hours are by appointment only.

Meeting Information

Room: CS 101

Time: Monday 10:00-12:45 (Lab)

Wednesday 10:00-12:45 (Lab)

Important Dates

CSUF's Academic Calendar is posted online at «<http://apps.fullerton.edu/AcademicCalendar/>». The Academic Calendar contains all the campus closures and holidays of which you should be aware.

CSUF's Admissions Calendar is posted online at «<http://www.fullerton.edu/admissions/Resources/Calendars.asp>». The Admissions Calendar contains all the major dates with respect to adding, dropping, and withdrawing from your classes.

September 4 Labor Day, no class

September 20 Midterm Exam #1

October 25 Midterm Exam #2

November 20-23 Fall break (Thanksgiving), no class

EPP Exam as final at the end of the course

Textbooks

Required

- *C++, How to Program, 10th Ed.*, by Deitel and Deitel, Pearson, 2017, ISBN: 978-013-444823-7.

Optional

- *C++ Primer, 5th Ed.*, Stanley Lippman, Josee Lajoie, and Barbara Moo, Pearson, 2013, ISBN: 978-0-321-71411-4.
- *Ruminations in C++*, Andrew Koenig and Barbara Moo, Addison-Wesley, 1997, ISBN:

978-0-201-42339-6.

- *Effective C++, 55 Specific Ways to Improve your Programs and Designs*, Scott Meyers, Addison-Wesley, 2005, ISBN: 978-0-321-33487-9.

Many popular technical books may be read online through the campus's subscription to Safari Books Online. From outside of the campus network, the campus library's WWW proxy will grant you access, «<http://www.library.fullerton.edu/asp/ipcheck.aspx?url=http://proquest.safaribooksonline.com/?unicode=calstate>». The Safari Books Online service can be accessed directly from any computer on the campus network, «<http://proquest.safaribooksonline.com/>».

Development Tool Resources

Students interested in using Microsoft[®] development tools may request a Dreamspark account at «<http://dsreqform.ecs.fullerton.edu/>». A student may, at no monetary cost, download full featured versions of Microsoft Visual Studio.

Students interested in using Apple[®] development tools can freely download Xcode through the App Store application bundled with OS X. Students may download Xcode directly from «<https://developer.apple.com/xcode/>».

A Debian-based GNU/Linux OS virtual machine ready for students use and Debian-style installation scripts are posted online at «<https://gamble.ecs.fullerton.edu/resources/>».

A CentOS-based shell server is available through secure shell (ssh) and secure file transfer protocol (sftp). The hostname is ecs.fullerton.edu. If your email address is malcolm@csu.fullerton.edu, then your username is ACAD\malcolm. If you are using a command-line ssh client, then your command to connect to ecs.fullerton.edu will be `ssh 'ACAD\malcolm@ecs.fullerton.edu'`. Your password is the same password as your CSUF Portal password.

Please consider adopting a package management system for your personal computer to facilitate adding, updating and removing the various software development tools you may wish to use.

Apple OS X

- MacPorts «<http://www.macports.org/>»
- Fink «<http://www.finkproject.org/>»
- Homebrew «<http://brew.sh/>»

Microsoft Windows

- Chocolatey NuGet «<https://chocolatey.org/>»
- Cygwin «<http://www.cygwin.com/>»
- Npackd «<https://npackd.appspot.com/>»

GNU/Linux OS

- dpkg «<https://www.debian.org/doc/manuals/debian-faq/ch-pkgtools.en.html>»
- rpm «<http://fedoranews.org/alex/tutorial/rpm/>»

Learning Goals

- Solve complex problems using a powerful, flexible programming language
- Know what the compiler does for you by default when designing classes
- Know how to use modern C++11 and C++14's idioms correctly
- Be able to correctly implement classes that manage dynamic memory
- Effectively use copy c'tors/assignment operators, and move c'tors/move assignment operators
- Be able to create template, container classes from scratch, similar to those used in the STL
- Know how to use common, STL template classes: `std::vector<T>`, `std::list<T>`, `std::map<T>`
- Be thoroughly familiar with the STL's algorithms, including `find`, `remove`, `replace`,
- Be able to correctly overload operators, and create classes that use them effectively, such as `matrix<T>` and `vector<T>` classes
- Know how to build and maintain balanced Red-black trees
- Be able to create hash-tables with linked-lists to handle hash-collisions
- Know how to implement the major sorting algorithms: Heapsort, Shellsort, Mergesort, and Quicksort
- Be familiar with common Design Patterns: Decorator, Singleton, Composite
- Know how to create classes that use Inheritance (is-a), and Composition (has-a)
- Know how to use abstract classes, and multiple inheritance using pure virtual classes

G.E. Requirements

This class does not meet any CSU General Education requirements.

Course Outline

- Review of EPP: classes, template classes, container classes
- EPP exam: Wednesday 23 Aug, and Monday 28 Aug.
- Introduction to Classes, Objects, Member Functions, and Strings
- Review of methods the C++ compiler writes for you
- Review of Recursion, and some classic recursion problems (n!, gcd, fibonacci, binary search)
- Class Templates, Arrays and `std::vector<T>`, Catching Exceptions
- Pointers, References, Pointers to pointers and References to pointers
- Class constructors, destructors, telescoping (delegating) constructors, throwing exceptions from c'tors, the this pointer

First Midterm: Thursday, Sep 21

- Operator overloading, dynamic memory allocation, copy c'tors/operator=, and move c'tors
- Inheritance hierarchies, c'tors/destructors in derived classes, Object slicing, Multiple inheritance
- Polymorphism and Abstract classes, Virtual and pure virtual methods, virtual destructors, checklist for class authors
- Design patterns: Decorator, Observer, and Composite
- Stream input/output, I/O file streams, stream manipulators and object serialization
- Standard library containers and iterators, `std::vector<T>`, `std::list<T>`, `std::deque<T>`, `std::map<T>`, overloading operator<, operator==

Second Midterm: Thursday, Oct 26

- STL algorithms, iterators/custom iterators, lambda functions, functors, fill, generate, equal, remove, replace, conditionals (remove_if, replace_if,...), searching/sorting
- Exception handlers, throw and re-throwing exceptions, stack unwinding, RAI (resource acquisition is initialization)
- Creating a custom Stack<T> class
- Creating custom container classes: List<T>, Queue<T>, Tree<T>
- Linear search, Binary search, Insertion, Selection, Merge, Shell sorts, and Red-black trees
- Design Pattern: Decorators, inheritance and composition combined
- **pImpl** Pattern, hiding implementation behind pointers to implementation classes
- ATM Case study using CRC (Class, responsibility, and collaboration) cards, and the UML (Unified modeling language)

Final project

EPP Exam as Final

Technical Proficiency

Technical proficiency in programming and software engineering should correspond to the prerequisite(s) of the course. Students are expected to be intimately familiar with their development platform of choice and be able to write/debug code in C++ at a level of proficiency corresponding to the course prerequisites. Technical proficiency with information technology, such as, but not limited to, the use of web-based online services, sending and receiving electronic mail, and desktop computer file systems, is assumed.

Grading

Plus and minus grading is not used when determining final grades.

Final grades are computed by first finding the average score in each category described in the table below on the right. All scores are normalized to a scale of 0 to 100 before being averaged. The average score for each category is then used to compute the weighted average, using the weights in the second table below.

Grade	% of Total Points
A	90 - 100 %
B	80 - 89 %
C	70 - 79 %
D	60 - 69 %
F	Below 60 %

Category	% of Final Grade
Assignments	40 %
Final Exam	25 %
Midterms	25 %
Participation	10 %

Graduate Grading

Graduate students that use this course on a graduate study plan must perform additional work and will be evaluated on a separate grading scale vis-à-vis their undergraduate counterparts.

An additional programming project is mandatory for all graduate students. The project is proposed by the student and approved by the instructor. Graduate students must have a project approved by the tenth week of the semester or face a penalty of –10% for each week it has not been approved similar to course rule. The final project is due on the last class meeting of the 14th week of instruction.

Plus and minus grading is not used when determining final grades.

Final grades are computed by first finding the average score in each category described in the table below on the right. All scores are normalized to a scale of 0 to 100 before being averaged. The average score for each category is then used to compute the weighted average according to the weights in the table below on the right.

Grade	% of Total Points
A	90 - 100 %
B	80 - 89 %
C	70 - 79 %
D	60 - 69 %
F	Below 60 %

Category	% of Final Grade
Assignments	30 %
Final Project	20 %
Final Exam	20 %
Midterms	20 %
Participation	10 %

Assignments

Programming and written assignments will be discussed in class and posted to the course website in advance of their due dates. Each assignment description will include the assignment's grading rubric. Reading assignments are outlined in the syllabus and it is the responsibility of the student to stay up to date with the reading.

Written assignments must be typeset and presented in a professional manner. Presentation, spelling and grammar can be worth up to 30% of a written assignment's grade.

All programming assignments must be written in the C++ programming language. Coding style must conform to professional norms. At a minimum, code must be commented, have descriptive names for identifiers, and contain a comment at the top of each file with pertinent information such as the student's name, email address, and assignment name. A plain text README.TXT must be included with each assignment submission summarizing and documenting the work submitted. For students unfamiliar with coding style, Google's style guides are an excellent starting point, <<https://github.com/google/styleguide>>, particularly their C++ style guide, <<https://google.github.io/styleguide/cppguide.html>>.

At the start of the semester, the instructor will detail the platform and tools used to grade student assignments. It is the student's responsibility to ensure that the assignments execute to his or her satisfaction on the instructor's grading platform.

There are approximately:

- 5 programming assignments
- 2 written assignments
- 14 weeks of reading assignments

Exceptions are made on a case by case basis given enough time and evidence to weigh the merits of the application.

Attendance Policy

Attending class is mandatory. Missing class is not allowed unless it is excused by the instructor. Missing class as part of a documented accommodation is guaranteed to be excused. The ADA accommodated student must make a reasonable effort to coordinate any absences with the instructor.

Make Up Policy

Exams and quizzes cannot be taken after they have been given in class. Due to an act of nature, personal medical emergency, a family crisis, an act of terrorism, severe civil unrest, etc. students have 10 calendar days to petition the instructor to retake any exam/quiz or submit an assignment without late penalty. Exceptions shall be made on a case by case basis, provided there is time to evaluate the merits of such an application.

Participation

In the context of this course, participation is defined as the following:

- Arriving to class prepared and on time.
- Taking notes.
- Actively listening to the lecture and asking questions when appropriate.
- Annotating code listings and handouts.
- Bringing any required materials to class.
- When needed/desired, seeking assistance to complete assignments.
- Barring emergency, not leaving the class session early without instructor consent.
- Not distracting oneself or others with smartphones, games, online diversions, etc.
- Respecting and treating the instructor and the student's peers civilly.

Required Material

- A writing instrument
- A notebook
- A USB memory stick
- A personal computer with the requisite development tools or regular access to a computer lab

Academic Dishonesty

Students are encouraged to assist one another and discuss the course materials with your peers. It is your responsibility to be aware of and follow the spirit of CSU Fullerton's academic honesty policy which can be found at «http://www.fullerton.edu/senate/publications_policies_resolutions/ups/UPS%20300/UPS%20300.021.pdf». Academic dishonesty will not be tolerated. The University Catalog and the Class Schedule provide a detailed description of Academic Dishonesty under *University Regulations*.

By submitting work for evaluation, you acknowledge that you have adhered to the spirit of the university's academic honesty policy and that your submission is an original work by you unless otherwise directed to work in groups. Failure to follow the spirit of the academic honesty policy will result in a severely negative evaluation of the work in question and may result in involving the Department Chair and the Judicial Affairs office to seek a disciplinary remedy.

ADA Accommodations

Any student who, because of a disability, may require special arrangements in order to meet course requirements must register with the Office of Disability Support Services within the first week of classes. The Office of Disability Support Services' website is «<http://www.fullerton.edu/DSS/>». They can be reached by phone at 657-278-3117 or TDD at 657-278-2786. Their email address is «dsservices@fullerton.edu». Their office is located in University Hall, room 101. The instructor may request verification of need from the Dean of Students Office. Students requesting accommodations shall inform their instructors during the first week of classes about any disability or special needs that may require specific arrangements/accommodations related to attending class sessions, completing course assignments, writing papers or quizzes, tests or examinations.

Student Resources

Any student who wishes to discuss any concern may contact the assistant deans of the college. Assistant deans are student advocates who will help you navigate the university's policies and procedures and assist with resolving any conflicts.

Assistant Dean for Student Affairs Carlos Santana
CS-206A (657) 278-4407 «csantana@fullerton.edu»

Assistant Dean International Programs and Global Engagement Lillybeth Sasis
CS-206A (657) 278-4881 «lsasis@fullerton.edu»

Emergency Procedures

For your own safety and the safety of others, each student is expected to read and understand the guidelines published at «<http://prepare.fullerton.edu/campuspreparedness/>». Should an emergency occur, follow the instructions given to you by faculty, staff, and public safety officials. An emergency information recording is available by calling the Campus Operation and Emergency Closure line at 657-278-4444.

Instructional Continuity

Due to an event such as an epidemic or a natural disaster that disrupts normal campus operations, students must monitor the course Titanium site and their campus email address for any instructions and assignments that the instructor announces.

Laboratory Safety

Safety is no accident. Learning and following the appropriate safety practices and protocols is an integral part to all laboratory courses. Following the appropriate safety practices and protocols minimizes the chances of repetitive stress injuries, mishandling hazardous materials, and injury to self and others. Additional campus laboratory safety information regarding hazardous materials is online at «<http://riskmanagement.fullerton.edu/laboratorysafety/>».

Extra Credit

There are no opportunities for extra credit.

Recording & Transcription of Class Content

Recording class content is governed by UPS 330.230, «http://www.fullerton.edu/senate/publications_policies_resolutions/ups/UPS%20300/UPS%20330.230.pdf». Each instructor must permit class content to be recorded or transcribed by students when mandated to do so by the Americans with Disabilities Act or by other federal or state laws. Any recording of class content is for private use and study and shall not be made publicly accessible without the written consent of the instructor and students in the class.

Course Rules & Classroom Management

Unless an agreement or accommodation is reached between the student and the instructor, these rules must be followed.

- Attendance at all regularly scheduled lecture and discussion section is mandatory.
- Do not eat during lecture.
- If it makes noise, silence it.
- Computer use is not allowed in lecture except for taking notes.
- The student is responsible to be aware of any course announcements including changes to due dates and requirements.
- Homework, programming assignments, etc. may not be submitted late.
- Third party work (code, artwork, etc.) may not be used in student work without prior instructor consent. Failure to gain and document instructor consent will be construed as willful academic dishonesty.
- When a third party's work is incorporated into student work after gaining instructor consent, failure to wholly document the work's origin, copyright and license will be construed as willful academic dishonesty.