

Object-Oriented Programming

CPSC 121 - 17/18

Fall 2019

(Rev. 1-20190809)

Description

This course is designed for students with basic programming knowledge. It introduces them to advanced programming concepts such as recursion, memory allocation, and object-oriented programming. Most software development projects use object-oriented programming to facilitate the development of large projects, improve code readability, and foster code reuse. Students' experience using this programming paradigm in class prepares them for future career opportunities. Students who complete this course will not only understand advanced programming concepts but also acquire knowledge that enables them to learn other programming languages.

Learning Objectives

At the end of the course you are expected to:

1. Design and implement software that makes appropriate use of fundamental programming constructs and data structures (e.g., expressions, conditions, loops, functions, primitive data types, arrays).
2. Design and implement software that makes appropriate use of object-oriented concepts (e.g., classes, objects, methods, composition, inheritance, abstract classes, polymorphism).
3. Design and implement software that makes appropriate use of advanced programming concepts (e.g., recursion, pointers, memory allocation, STL, file input/output, exception handling).
4. Demonstrate proper use of encapsulation and decoupling.
5. Demonstrate proper use of coding standards.
6. Write appropriate comments that help other developers understand and reuse code.
7. Demonstrate the ability to utilize IDEs to optimize software development.

Prerequisite

CPSC 120 or passing score on the Computer Science Placement Exam



G.E. Requirements

This class does not meet any CSU General Education requirements.

Meeting Information

	Tuesday	Thursday
Lecture (CS102-B)	9:00 - 9:50 am	9:00 - 9:50 am
Lab (CS408)	10:00 - 11:50 am	

Instructor

Chary Vielma

Phone: 562-383-2285

Email: cvielma@fullerton.edu

Office: PLN 320-4

Office Hours: Thursdays 10:00am - 11:00am

You can also send an email to schedule an appointment. During final exam week, office hours are by appointment only.

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 you should be aware of.

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.

The final exam schedule is [determined by Admissions and Records](#). Makeup exams are only available by advance request for documented exceptional circumstances.

Tentative Midterm exam: Oct 22/24, 2019

Tentative Final exam: Dec 19, 2019 9:00 -10:50 am CS 300

Textbook

Gaddis, T., Walters, J., Muganda, G. (2017). *Starting Out with C++: Early Objects* (9th ed.). Boston, MA: Pearson

Online resources

1. <http://cplusplus.com>
2. <http://pythontutor.com/cpp.html>

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/?uicode=calstate»](http://www.library.fullerton.edu/asp/ipcheck.aspx?url=http://proquest.safaribooksonline.com/?uicode=calstate). The Safari Books Online service can be accessed directly from any computer on the campus network, [«http://proquest.safaribooksonline.com/»](http://proquest.safaribooksonline.com/).

Tuffix Development Environment

Students enrolled in CISC 120, CISC 121, and CISC 131 are recommended to use the Computer Science Department's official Linux development environment, Tuffix. Tuffix is Tuffy the Titan's Linux distribution.

Both CS 104 and CS 408 provide this development environment through Virtual Box. You are encouraged to use the same development environment on your own computers to ensure that codes work as expected.

Instructions on how to install Tuffix or a Tuffix based VM are online at [«http://csufcs.com/tuffixinstall»](http://csufcs.com/tuffixinstall). The Tuffix Titanium Community for Students, <https://communities.fullerton.edu/course/view.php?id=1547> is the best venue to receive help with Tuffix. You can self-enroll in the community using the instructions at [«http://titaniumhelp.fullerton.edu/m/FAQ/1/492060-how-do-i-enable-self-enrollment-for-my-community»](http://titaniumhelp.fullerton.edu/m/FAQ/1/492060-how-do-i-enable-self-enrollment-for-my-community).)

The Tuffix home page is [«https://github.com/kevinwortman/tuffix»](https://github.com/kevinwortman/tuffix).

Grading Source Code

Student source code will be evaluated and graded for correctness using a semi-automated process based on a GNU/Linux-based environment. Source code will be collected using GitHub [«www.github.com»](http://www.github.com), a web-based code management service. You should create a free account on [«www.github.com»](http://www.github.com) with your csu.fullerton.edu email address. You can add your csu.fullerton.edu email address to your GitHub account if you already have one. Detailed instructions for submitting source code will be given along with the programming exercise or project.

You should write portable code that works in all contemporary C++ version 17 environments. For grading purposes, code is considered to work to the extent that it works in GNU/Linux.

Course Outline (Subject to change)

Week (Days of the Month)	Topic	Book Reference
Week 1 (Aug 27, 29)	Class introduction and administration, C++ review	Ch. 2
Week 2 (Sep 3, 5)	C++ expressions and operators	Ch. 3
Week 3 (Sep 10, 12)	Conditions and Loops	Ch. 4, Ch. 5
Week 4 (Sep 17, 19)	Files and Functions	Ch. 6
Week 5 (Sep 24, 26)	Classes: concepts and terminology	Ch. 7.1-7.5
Week 6 (Oct 1, 3)	Arrays and Search	Ch. 8, Ch. 9.1-9.2
Week 7 (Oct 8, 10)	Recursion	Ch. 14
Week 8 (Oct 15, 17)	Pointers and dynamic memory	Ch. 10.1 - 10.9
Week 9 (Oct 22, 24)	Midterm Exam	
Week 10 (Oct 29, 31)	Classes: constructors, destructors, and composition	Ch. 7.6-7.10
Week 11 (Nov 5, 7)	Classes and Pointers	Ch. 10.11
Week 12 (Nov 12, 14)	Inheritance	Ch. 11.12
Week 13 (Nov 19, 21)	Polymorphism, Abstract classes	Ch. 15.2, 15.3
Week 14 (Nov 26, 28)	Spring Recess	
Week 15 (Dec 3, 5)	Templates, STL	Ch. 16.2, 16.3, 16.5
Week 16 (Dec 10, 12)	Exceptions	Ch. 16.1
Week 17 (Dec 19)	Final Exam	

Take note, that the course outline and schedule may change. In this case, the syllabus will be updated, and you will be informed of the changes accordingly.

Technical Proficiency

You are expected to be intimately familiar with fundamental programming concepts. The first part of the course will review related topics, but you are responsible for additional review on those prerequisite topics. You can consult your supplemental instruction leaders or me if you have questions.

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

Final grades are computed by first finding the average score in each category described in the first table. All scores are normalized to a scale of 0 to 100 before averaging. The average score for each category is added to the final grade according to the weights listed under % of Final grade. Plus and minus grading is not used when determining final grades.

Category	% of Final Grade
Participation	10%
Quizzes	15%
Lab exercises	15%
Project	15%
Midterm	20%
Final Exam	25%

Letter Grade	Final Grade
A	90–100%
B	80–89%
C	70–79%
D	60–69%
F	Below 59%

I reserve the right to *decrease* grade boundaries, but *will not increase* them. I may adjust boundaries downward, awarding higher letter grades, to compensate for assessments that were too difficult. Grade boundaries are not changed unless the unadjusted class average is significantly lower than the department's GPA average.

Participation

Each week will consist of two lecture classes to discuss the topic for the week (Tu and Th) and a laboratory session to supplement what is learned through hands-on programming practice (Tu or Th). Although it is not a requirement, you are encouraged to read the assigned book references and corresponding slides to prepare for the lecture and to answer questions you may have about the discussions, quizzes, lab exercises, and projects.

Participation will be graded equally between attendance and classroom activities. It is important that you attend lecture and lab sessions not only to learn from lectures, but also to ask questions, to participate in class

activities, to take quizzes, and to engage in hands-on programming practice. These activities are important because research shows that active learning leads to better learning compared to simply reading the book or listening to lectures. You are allowed two absences to accommodate unforeseen issues (medical or otherwise). More details are provided in the attendance policy section.

Classroom activities expand your knowledge and help practice your skills learned from lectures. Some examples of activities include group discussions, mini-presentations, group programming exercises, and so forth. Your grade will be based on participation in these activities and not just performance.

Quizzes

Quizzes are given on the **first five minutes of the first lecture class** of each week (Tu). It focuses on the previous week's discussion, but may also cover earlier topics. Quizzes will help you evaluate your current understanding and help you recall information we need for the new topic of the week.

There will be a maximum of 12 quizzes throughout the course as no quizzes will be given on the first day of class and lecture classes after an exam. Your **two lowest-scoring quizzes will not be counted** towards the final grade to accommodate unforeseen absences. Quizzes are part of the final grade, so you are encouraged to arrive in class on time.

Quizzes will be conducted on Titanium. You have the option to use your *mobile phone*, a *tablet*, or your *laptop*. Many students use their laptops because it is easier to see the questions and submit answers. If you do not have a laptop available, the library can loan you a laptop for the entire term for free. See the device rental section for more details.

Lab exercises

We will have lab sessions every week devoted to hands-on programming exercises. This will help you practice what we learned from the class discussions. Instructions for each lab exercise will be provided with the programming problem.

A lab exercise will contain around five programming problems, but you will be required to **answer at least two problems correctly** before moving to the next lab exercise. All your solutions to programming problems will need to pass the unit test, style checker, and format checker to get full points. You will be able to run all three tests yourself to check if you can move to the next lab exercise. Details about running these tools will be provided with the programming problem.

You are given **two weeks to complete** each lab exercise. However, it is highly suggested that you finish the lab exercise during lab session it is released to ensure you get full points. Otherwise, you will only get a grade for the lab exercises that you completed. Partial points will be given depending on how much you completed the lab exercise. You are encouraged to use our lab time to complete the lab exercises because you will likely use your time outside of class to work on your projects.

Your **two lowest-scoring lab exercises will not be counted** towards the final grade to accommodate unforeseen absences. However, you are encouraged to arrive in class on time because lab exercises are part of the final grade. I want everyone to have enough time to complete the lab exercise.

More details about grading and other instructions are discussed in the lab exercise guide.

Projects

We will have one project for the course, but you are required to submit up to eight separate milestones on specific dates throughout the semester. This format will help to ensure that you finish the project on time and give you frequent feedback on your performance. Generally, **each milestone will be due every two weeks** and will be submitted through GitHub. Make sure you submit your project on time to get credit for your work.

Each milestone will contain skeleton code and unit tests that will help you check whether you implemented all the required features and if you implemented them correctly. Your grade will be based on successfully implementing the required features, the design of your code to achieve those requirements, and documentation. For example, you will not get full points if you manually assign values to each array element (brute force solution) instead of using a loop.

Please follow proper coding style and provide sufficient internal documentation. Your project must be written in C++ using a coding style that conforms to professional norms. At a minimum, code must be commented. 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>».

I will use the Tuffix development environment for grading. It is your responsibility to ensure that your project executes in this environment. Exceptions are made on a case by case basis given enough time and evidence to weigh the merits of the application. All milestones will count toward your final grade.

If you are unable to complete the milestone on time, make sure you still submit it to get partial points. I will share my solution a few days after the deadline so you can see how I would solve it. You are encouraged to use your own solution as a starting point for the next milestone, but you can adopt my code if you wish to do so. More details about grading and other instructions are discussed in the project guide.

Examinations

The midterm and finals are not cumulative, but it will require your knowledge of previous topics. For example, the finals might ask questions about inheritance which requires the understanding of classes. Exams are **closed books and closed notes**. There will be a **conceptual and hands-on part** in both exams.

Grade exceptions

Please make sure that you work hard, get help when necessary, attend class, complete all the projects, and do well on exams especially if there is a lot riding on your grade. Please note that there is one syllabus for the course; all students are graded based on the requirements outlined in the syllabus, and nothing more. There are no special deals, relaxed standards or extra opportunities based on class standing or other factors. Your grade is based on your graded work, and that alone. That's an essential part of a fair grading system.

You have the right to ask if your grade was given in error. I will be happy to check your scores to verify that no clerical error was made. In case of errors, grade changes will be corrected promptly. Please consult me early, so we avoid issues later when there is limited time to make corrections.

Grade calculations

I record grades in Titanium. Please check them for accuracy weekly. Titanium calculates your grade automatically. These calculations are based only on the grades that are currently available. So, for example, the grade calculation will ignore the project category until I've graded the first project. Titanium automatically drops low scores where appropriate, but only once we're far enough along that at least one score will be counted.

Device Rental

Our university offers free rentals of laptops and mobile devices that you can use for our class. Students who rented laptops in previous semesters shared with me that the machines provided more than enough resources required to run the applications we used for class. Kindly follow the links below to know more about renting devices.

Long-term laptop checkout «<https://www.fullerton.edu/it/students/equipment/longtermlaptop.php>»

Long-term smartphone checkout «<https://www.fullerton.edu/it/students/equipment/smartphone.php>»

Attendance Policy

As mentioned previously, you **may choose to be absent twice in a lecture session and twice in a lab session** during the semester without needing to justify the absence to accommodate unforeseen issues. Attendance will be tracked via the iFullerton app and your participation in class or lab activities. If you logged your attendance on iFullerton but did not participate in a class activity, for example, then you will still be considered absent. You are expected to log your attendance twice; once for the lecture and once for the lab session. In case you are unable to log your attendance in the iFullerton app, please let me know so I can mark your attendance manually.

Allowed absences will not result in grade deductions for participation, but **quizzes and lab exercises may be affected depending on your prior attendance and scores**. You will be responsible for catching up on what you missed from class and I will not be obligated to give make-up lectures or activities for that day.

Absences outside of the two lecture and lab sessions will affect your grade. You will not receive the corresponding participation, quiz, or lab exercise grade for the day of the unexcused absence.

Makeup Policy

Outside of the aforementioned two-day allowed absence, students who miss quizzes, lab exercises, or exams due to valid reasons can still request to retake it within 10-calendar days of the absence. Some valid reasons

may include religious events, university-approved activities, acts of nature, personal medical emergencies, family crisis, acts of terrorism, severe civil unrest, and so forth. In some cases, such as university-approved activities, and medical and personal reasons, you may be asked to provide an official letter from a supervisor, physician, or guardian.

Missing class as part of a documented accommodation is guaranteed to be excused. An ADA accommodated student must make a reasonable effort to coordinate any absences with the instructor. Exceptions shall be made on a case by case basis, provided there is time to evaluate the merits of such an application.

Administrative drops

Any student who misses the first class meeting may be dropped from the class unless they contact the instructor or Computer Science department within 24 hours.

Communication

You have a CSUF-supplied email account, and that is the only way I have of reaching you outside of class. Check that account daily for important class announcements and individual messages. I try to respond to all emails within two working days but occasionally may take longer than that. Plan accordingly, especially around deadlines.

We have a class forum on Piazza that I highly encourage you to use. You and your classmates will surely have similar questions so feel free to post them there. You may see your question already answered and if not, you might get a quicker reply from your classmates than me.

However, please refrain from posting answers to graded activities or asking questions about those answers on the forums unless I have finished grading them or I have specifically allowed the class to post about it.

Academic Dishonesty

You are encouraged to assist one another and discuss the course materials with your peers. However, 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 unless otherwise directed to work in groups. Plagiarism and cheating are serious academic offenses with serious consequences. Anyone discovered engaging in such behavior will automatically receive a zero for the activity on the first warning. Receiving a second warning will 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, you must monitor the course Titanium site and your campus email address for any instructions and assignments that I announce.

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

You are requested to follow the guidelines listed below to facilitate learning and to foster a supportive and inclusive learning environment:

- Arriving to class prepared, with any required materials, and on time.
- Actively listening to the lecture, taking notes, and asking questions when appropriate.
- Not distracting oneself or others with smartphones, computers, games, online diversions, etc.
- Respecting and treating the instructor and peers civilly.
- Barring an emergency, not leaving the class session early.
- When needed/desired, seeking assistance to complete assignments. Assistance does not include asking someone to give you the answers, copying/pasting someone else's code or code from the internet.
- Being aware of course announcements including changes to due dates and requirements.
- Getting prior instructor consent and providing proper documentation for using third-party work (code, artwork, etc.).

Acknowledgment

Portions of this syllabus draw from syllabi authored by Professors David Falconer, Matt Huffman, David A. Mix-Barrington, Mariko Molodowitch, Michael Shafae, and Kevin Wortman.

Supplemental Instruction (SI)

Supplemental Instruction (SI) study sessions are offered for this course. SI sessions meet two to three times a week, throughout the semester. Supplemental Instruction is an academic assistance program which provides peer-led group study sessions to assist students in traditionally difficult courses.

SI sessions are led by a SI leader who has already mastered the course material and has been trained to facilitate group sessions where students can meet to improve their understanding of course material, review and discuss important concepts, develop study strategies and prepare for exams. *SI is for everyone, and open to all students enrolled in this class; not just those students who are struggling.* Attendance at SI sessions is free and voluntary. Students who attend SI sessions weekly, typically earn higher final course and exam grades than students who do not participate in SI. Please bring your lecture notes, books, and questions with you.

SI sessions for this class will meet at the following days/times:

Sec 05/07: MoWe 10:00am - 11:15am CS 209 Andrew Lopez

Sec 13: MoWe 10:30am - 11:45am CS 209 Brittany Kraemer

Sec 09/11: TuTh 8:00am - 9:15am CS 209 Ean McGilvery

SI location: **CS-209**

SI leader for this class: **TBD**

For additional information on the SI Program at CSUF, please visit our website at:

<http://www.fullerton.edu/SI>