

CS 162: Introduction to Computer Science II

Summer 2019

Credits: 4

Terms Offered: All terms

Instructor Name: Luyao Zhang

Instructor Email: zhangluy@oregonstate.edu

Course Description

Basic data structures; Computer programming techniques and application of software engineering principles; Introduction to analysis of programs.

Prereqs: CS 161 or EECS 161

Textbook (required)

Gaddis et al, **Starting Out with C++: Early Objects**, 9th Edition (ISBN-13: 978-0134400242, ISBN-10: 0134400240)

Canvas

This course will be delivered via **Canvas** where you will interact with your classmates and with your instructor. Within the course Canvas site, you will access the learning materials, such as the syllabus, assignments and quizzes. For technical assistance, please visit [Ecampus Technical Help](#).

Communication Policy

Please post all the course-related questions on **Piazza Discussions** forum so that the whole class may benefit from our conversation. Please email me for matters of a personal nature. Please include the tag **[CS162 Summer 2019]** in your email subject and use your **OSU email** instead of the Canvas email. I will reply to course-related questions and emails within 48 hours. I will also send out the course updates through **Canvas Announcement** to the whole class and you will receive an email notification on each announcement. It is your responsibility to check your OSU email and the Canvas course website regularly to keep updated.

For questions and requests about grading and re-grading, please post privately on Piazza and @ your grading TA. Please don't post any grade-related request on Slack. TAs will hold office hours online (WebEx or Slack) from week 1 to week 10 (to week 7 for summer term). The office hour schedule and link will be posted on Canvas in the "Start here" module. TA's contact information and who will grade your assignment will also be posted in the "Start here" module. These contact information will be updated each term within week 1.

Measurable Student Learning Outcomes

At the completion of the course, students will be able to:

1. **Design** and **implement** programs that require:
 - (a) **multiple classes**, structures
 - (b) hierarchies of classes that use **inheritance** and **polymorphism**
 - (c) understanding of abstraction, modularity, separation of concerns, exception handling
2. **Construct** and **use** basic **linear structures** (arrays, stacks, queues, and various linked lists) in programs, and be able to describe instances appropriate for their use.
3. **Classify** moderately complicated **algorithms** in these complexity classes: $O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, and $O(n^2)$.
4. **Develop** test-data sets and testing plans for programming projects
5. **Produce recursive** algorithms, and choose appropriately between iterative and recursive algorithms.

Graded Course Works

- **Tests and Quizzes**

There will be a policies quiz at the beginning of the course, one course survey at the end of the term, and four open-book tests. Each test will have around 20 questions (T/F and multiple choices) on Canvas. You may take the test a second time, but only the score of the **final attempt** will count. Each attempt will be timed. You may not get the same questions each time.

- **Projects**

There are 3 larger programming project assignments. An assignment generally includes understanding the requirements, designing the program, implementing and testing the code, and writing a reflection document. Programs are graded on how well

they solve the assigned problem, meet specifications, use proper formatting and documentation. All the 3 projects will be individual work.

- **Labs**

Labs are small development projects that reinforce the topics presented each week. You will have one lab each week, and 7 labs in total. All the labs will be individual work.

- **Group Activities**

Activities include two types of activities that support the course objectives. You will discuss certain topics on some concepts and techniques on Piazza with your classmates. There is a final reflection discussion (week 6 & 7) about your journey through 161-162 to learn how to program. This discussion will be on Canvas and will be available for about 2 week periods. There will be two group project peer reviews during the 4th-6th weeks of the term and you need to form a team to review each other's code and give comments.

- **Final Project**

There will be a final project instead of a final exam. It will be similar in the format to the projects but will cover concepts from the entire course. The final project is also individual work.

Grading Policies

The list below indicates how the course learning outcomes will be measured:

- Projects – 30%
- Labs – 30%
- Group activities – 10%
- Quizzes/Test – 15%
- Final project – 15%
- Total – 100%

Grade letter	Percentage floor
A	≥ 93
A-	≥ 90

Grade letter	Percentage floor
B+	≥ 87
B	≥ 83
B-	≥ 80
C+	≥ 77
C	≥ 73
C-	≥ 70
D+	≥ 67
D	≥ 63
D-	≥ 60
F	< 60

REMINDER: A passing grade for classes in CS is a **C** or above. A C- in a CS course is not considered a passing grade toward a CS degree or as a prerequisite for future CS classes.

Your grade for each assignment will be posted on **Canvas** (generally after one week of the due date). Canvas is used to simply record the scores. The final score displayed is only **approximate**. At any time, if you want a better estimate of your current grade in the course, please **email the instructor**.

Re-grading: If you have a question about an assignment grade, you must contact your TA by **post privately on Piazza** within **ONE WEEK** of receiving your grade. You can use words like "regrading request lab 1 @ TA's name" in the post subject so we can see it right away. After one week, you will not be able to dispute your grade.

Late policy:

- The **final project, last lab, group activities and test MUST** be completed by the deadline, and **no late submission** would be accepted for grading purpose.

- Projects and labs should be completed by the due date. If you do not submit the assignment by the due date, there is **late penalty**:
- Late **<= 1 day**: **10%** penalty;
- **1 day < Late <= 3 days**: **20%** penalty;
- **3 days < Late <= 7 days**: **30%** penalty;
- **Late > 7 days**: **not be accepted**.

Work submitted after 7 days will not be accepted. It is your responsibility to manage your time. If there are extenuating circumstances, please contact the instructor as soon as possible **before the deadline**.

- You have **three bonus days** in the entire term to apply to any labs/project. You can use it all at once for one assignment (if you are late for 3 days), or split it and use one day each for three assignments (no "half" day).
- **How to apply the bonus day**: leave a comment on Canvas under that assignment submission, saying that you would like to apply x bonus days for this late submission, and you have y bonus days left after that. When TAs are applying late penalty, they will look at your submission time and your comments. If you don't leave a comment there, TAs will directly apply the late penalty.
- If you have a really tough situation that might affect your progress a lot (illness, job duties, family emergency...), you should contact me immediately and ask for extensions other than the 3 Bonus days. **Don't wait until the due date or even past the due date to explain your personal situations and ask for extensions.** If you are not sure whether to ask for it, better do it.

Course Work Submission

- All work must be submitted before **23:59 (Pacific Time Zone)** on the date they are due.
- All the files need be archived in a zip file
- Your submission must be named in the following format: assignment name + your last name + your first name for individual assignment (eg: Lab1_Jones_Adam).
- All the projects and labs must be submitted on Canvas.
- Programs must **compile and run** on the EECS server (flip) or they will not be graded. Programs must include a **makefile**.
- If you are not sure you submit the correct file, **download it and check it**. Make sure you include everything in your submission!

Program Re-submission: assignment re-submission is not allowed in general. If you are not 100% sure your code could run on flip, don't submit it. If your code does not compile on the flip, TAs will not grade it. If you couldn't get the code work and just want to get partial credits for the effort, leave a comment on Canvas and TAs will grade according to that. If you submit a wrong file and find it out later on, resubmit it on Canvas. TA will always grade the latest version. If the re-submission is past the due

time, don't touch the original file and show us that you finish it on time. Otherwise, late penalty will be applied accordingly.

Topics by Weeks

Week	Topic(s)	Due
1	Pointers Review, Separate Compilation and Makefile	Lab 1 , Course policies quiz
2	Software Design, Testing and Debugging	Lab 2, Project 1,Test 1
3	Classes and Inheritance	Lab 3
4	Polymorphism and Virtual Functions	Lab 4, Project 2,Test 2, project peer review 1
5	Recursion	Lab 5
6	File I/O and Linked Lists,ADT: Stacks and Queues	Lab 6, Project 3, Group Project, Test 3,project peer review 2
7	Complexity Analysis: Searching and Sorting, STL, Templates and Exceptions	Lab 7, Test 4
8		Final project, Reflections, Course survey

You can see the assignment due dates on Canvas directly. You can also download it here:

Programming assignments in this course are considered **Take Home Programming Tests**. You must do your own work, entirely.

To Do & Not To Do:

You **MAY** discuss the meaning of assignments, general approaches, and strategies with other students in the course.

You **MAY** show your code to the TAs or instructor for feedback and help.

You **MAY NOT** ask another student for help debugging your assignment code.

You **MAY NOT** use or copy code from any other source, including the internet.

You **MUST** write your own code for your assignments. That doesn't mean we don't allow you to search solutions online when you meet problems. That means, if you learn how to implement some function from online recourse, do not just copy/paste the code. Make sure you cite the resources properly if the code is exactly the same, and also make sure you could write the code by yourself now without looking at the original source.

We use **plagiarism-detection software** to check your code against the code from other students. It is quite sophisticated and can easily see through **variable name changes and formatting differences**. If you are found in violation of any of the above policies, whether you are the giver or receiver of help, you will receive a zero on the assignment or fail the course (Instructor's decision). For further information, visit [Academic or Scholarly Dishonesty](#), or contact the office of Student Conduct and Community Standards (SCCS) at 541-737-3656.

Statement Regarding Students with Disabilities

Accommodations are collaborative efforts between students, faculty and [Disability Access Services \(DAS\)](#), with accommodations approved through DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at 541-737-4098.

NOTE for Disability Access Services (DAS) – If you have accommodations through DAS for extra time on your exams or quizzes it is very important that you communicate with your instructors as soon as possible. Ask the instructor to double check all timed exams and quizzes to make sure that extra time has been given to you for each exam. The instructor has to do this for each timed exam or quiz manually

If you start an exam and do not see your extended time, please have your proctor call Ecampus Testing or try to contact your instructor for assistance. We can give you extra time while you are still taking the exam if we can be contacted before the exam submits.

After you read all the details in the syllabus and start here module, please take the **syllabus quiz** and get full points to unlock the week 1 module.

