

SYLLABUS
CS 162: Introduction to Computer Science II
Winter 2017

Credits: 4

Terms Offered: All terms

Instructor Name: Luyao Zhang

Instructor Email: zhangluy@oregonstate.edu

Office: KEC 1099

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, projects, and quizzes. For technical assistance, please visit [Ecampus Technical Help](#).

Communication

Please post all 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 Winter 2017]** in your email subject. I will reply to course-related questions and emails within 24-48 hours. For questions about assignments and grading, please contact the TA that grade your assignment.

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 4 tests for this course. There will be a test about every 2 weeks. The tests will not be proctored. Each test will have around 20 questions (T/F and multiple choices) in 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 4 larger programming project assignments. An assignment generally includes: understanding, design, implementation, testing, and reflection. Programs are graded on how well they solve the assigned problem, meet specifications, use proper formatting and documentation. All the 4 projects will be individual work.

- **Labs**

Labs are small development projects that reinforce the topics presented each week.

- **Group Activities**

Activities include two types of activities that support the course objectives. You will be given some questions, and as a group, discuss what each other has found. There is a final reflection discussion about your journey through 161-162 to learn how to program. This discussion will be in Canvas and will be available for about 2 week periods. There will be one group project during the 5th-7th weeks of the term and you need to form a team to work together and submit one work.

- **Final Project**

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

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	Score
A	≥ 92
A-	≥ 90
B+	≥ 87
B	≥ 82
B-	≥ 80
C+	≥ 77
C	≥ 72
C-	≥ 70
D+	≥ 67
D	≥ 62
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** from your OSU email account.

Re-grading: If you have a question about an assignment grade, you must contact your TAs through **EMAIL** within **ONE WEEK** of receiving your grade. After one week, you will not be able to dispute your grade.

Late policy:

- Group activities, the final project and quiz/test **MUST** be completed by the deadline, and **no late submission** would be accepted.
- 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 ≤ 24 hours: **10%** penalty;
 - $24 \text{ hours} < \text{Late} \leq 48$ hours: **20%** penalty;
 - $48 \text{ hours} < \text{Late} \leq 72$ hours: **30%** penalty;

- **Late > 72 hours: not be accepted.**

Work submitted after 72 hours 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. Check the note about late policy on Canvas for more details.

Course Work Submission

- All work must be submitted before **23:59 (Pacific Time Zone)** on the date they are due.
- If your submission includes more than one file, all the files need be archived in a **zipfile**.
- Your submission must be named in the following format: assignment name + your last name + your first name for individual assignment (eg: Lab 1 John Carter).
- Programs must **compile and run** on the EECS server (flip) or they will not be graded. Programs must include a **makefile**.
- If you submit **multiple times**, only the **most recent submission** to flip will be used for grading. Make sure you include everything in your submission!

Topics by Weeks

Week	Topic(s)	Due
1	Introduction, separate compilation and makefile	Lab 1 , Course policies quiz
2	Software Design and Testing	Lab 2, Assignment 1 Group Activity 1
3	Classes and Inheritance	Lab 3, Test 1
4	Recursion	Lab 4, Assignment 2
5	Polymorphism	Lab 5, Test 2
6	Complexity, Searching and Sorting	Lab 6, Assignment 3, Group Activity 3
7	Intro to Linear Data Structure, Linked Lists	Lab 7, Test 3
8	Stacks and Queues	Lab 8, Assignment 4, Group Activity 4
9	STL, Templates, exceptions	Lab 9
10	Recursion vs Iteration	Lab 10, Reflections
Final		Final Project

Final Project Due: **Tuesday, March 21, 2016**

Academic Integrity

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.

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.