

**SYLLABUS**  
**CS 162: Introduction to Computer Science II**  
**Fall 2016**

---

**Credits: 4**

**Terms Offered: All terms**

**Instructor Name: Luyao Zhang**

**Instructor Email: [zhangluy@oregonstate.edu](mailto:zhangluy@oregonstate.edu)**

**Office: KEC 1099**

**Office Phone: (541) 737-3468**

**Course Description**

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

**Prereqs:** CS 161 or EECS 161

**Textbook (required)**

Gaddis et al, **Starting Out with C++: Early Objects**, 8<sup>th</sup> Edition (ISBN-13: 978-0133360929, ISBN-10: 013336092X) **or** 9<sup>th</sup> 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, class discussions, assignments, projects, and quizzes. For technical assistance, please visit [Ecampus Technical Help](#).

**Communication**

Please post all course-related questions in **Canvas 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 Fall 2016]** in your email subject. I will reply to course-related questions and emails within 24-48 hours.

**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.

## Course Works

- **Tests and quizzes**

There will be 1-2 quizzes and 4 tests for this course.

There will be a policies quiz over the syllabus. This will be a 15 question (T/F and multiple choice) quiz in Canvas. You can take it as many times as you need, but you **MUST** have a score of 100% to continue with the course.

There be will a test about every 2 weeks. The tests will not be proctored. Each test will have around 20 questions (T/F and multiple choice) 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.

- **Assignments**

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.

- **Labs**

Labs are small development projects that reinforce the topics presented each week. They may be broken into mini-labs which may be standalone or combined into a single program. Some mini-labs will be completed in a peer-led study session each week. If you are not able to attend a study session, then you will be required to turn in your work through TEACH. For students completing their exercise as a group in a peer-led study session, you will automatically receive the full points for completing the work at the session. Those students who do not attend a peer-led study session to complete their exercise, will be graded on their individual work.

- **Group Activities**

Activities include many different types of small activities that support the course objectives.

There will be an initial group activity. You will work on problems (puzzles) in small groups. You will be asked to describe the process you use or develop in solving the problems. You will, as a group, discuss what each other has found or discovered and put together a single document describing the results. Developing this process is as important as solving the problems provided.

There is a final reflections discussion about your journey through 161-162 to learning how to program. This discussion will be in Canvas and will be available for designated 2 week periods.

- **Final Project**

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

### **Grading Policies**

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

- Quizzes/Test – 20%
- Assignments – 30%
- Labs – 20%
- Group activities – 10%
- Final project – 20%
- Total – 100%

<b>Grade</b>	<b>Score</b>
<b>A</b>	<b>&gt;= 92</b>
<b>A-</b>	<b>&gt;= 90</b>
<b>B+</b>	<b>&gt;= 87</b>
<b>B</b>	<b>&gt;= 82</b>
<b>B-</b>	<b>&gt;= 80</b>
<b>C+</b>	<b>&gt;= 77</b>
<b>C</b>	<b>&gt;= 72</b>
<b>C-</b>	<b>&gt;= 70</b>
<b>D+</b>	<b>&gt;= 67</b>
<b>D</b>	<b>&gt;= 62</b>
<b>D-</b>	<b>&gt;= 60</b>
<b>F</b>	<b>&lt; 60</b>

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.

### Course Work Submission

- You will submit all coursework (labs, assignments, reports) in **TEACH** unless otherwise directed. 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) or assignment name + group number for group work (eg: Lab 1 Group 1).
- 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!

### Late policy:

- Discussions, activities, the final project and tests **MUST** be completed by the deadline, and no late submission would be accepted.
- Assignments and labs should be completed by the due date. If you do not submit the assignment or lab by the due date, there is **late penalty**:
  - Late  $\leq$  **24** hours: **10%** penalty (original score \* 90% = score after late penalty);
  - 24 hours < Late  $\leq$  **48** hours: **25%** penalty;
  - Late > 48 hours: **not be accepted**.
- Work submitted after 48 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.

### 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 (verbally) 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** share assignment code, pseudocode, or documentation of any kind with any other student in the course.

You **MAY NOT** show your assignment code to another student in the course for any reason.

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). The academic dishonesty charge will be documented and sent to your school's dean and the Office of Student Conduct. The first offense results in a warning; the second offense results in an academic dishonesty charge on your transcript, a disciplinary hearing, and possible expulsion. 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.