



## **Ecampus** SYLLABUS

**Course Name: CS 162 – Introduction to Computer Science II**

**Credits: 4**

**Instructor name: Terry Rooker**

**Instructor email:** [rookert@eecs.oregonstate.edu](mailto:rookert@eecs.oregonstate.edu)

**Office:** KEC 2099

**Phone:** For a number of reasons, do not expect me to answer if you call my office phone. Email is the best way to get in touch with me.

### **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

### **Learning Resources**

Gaddis et al, *Starting Out with C++: Early Objects* (8th Ed.) . ISBN-13: 978-0133360929 ISBN-10: 013336092X Digital: 9780133449198

### **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. To preview how an online course works, visit the [Ecampus Course Demo](#). For technical assistance, please visit [Ecampus Technical Help](#).

### **Measurable Student Learning Outcomes**

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. an understanding of abstraction, modularity, separation of concerns, and 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.

**Grading:**

You will submit all coursework (modules, assignments, reports) in TEACH unless otherwise directed. All work must be submitted before 23:59 (**Pacific Time Zone**) on the date they are due.

Late submissions. **Discussions, activities, modules, and tests MUST be completed by the deadline.** Assignments and the final project should be completed by the due date. If you do not submit the assignment or project by the due date, but within 24 hours there is a 10% penalty. If it is submitted no more than 48 hours late then there is a 25% penalty. Work submitted any later will not be accepted. It is your responsibility to manage your time. If there is a last minute glitch and you miss the deadline by a minute, it is still late. If there are extenuating circumstances please contact the instructor as soon as possible, preferably before the deadline.

Many students ask to see code from their peers, but few volunteer. **The grader may select any program submitted for grading to post anonymously for discussion.** These examples may be chosen, as they are interesting or elegant. In some cases, an example may be chosen because, while not wrong, indicates a misunderstanding of a concept or an inefficient way to solve the problem. Sharing has no bearing on the grade for the work!

**Modules - 20%**

Modules are small development projects that reinforce the topics presented each week. They may be standalone or combined into a single program.

Each Module will be graded S/U. Feedback will be provided by the grader. The student is expected to correct significant errors. Many of the Modules will build on the previous smaller programs so these corrections are essential.

At the end of the quarter the final large program will be graded. This final grade will be used in your final score calculation.

**Programs must compile and run on the EECS server (flip) or they will not be graded. Programs must include a makefile and be archived in a zipfile. The zip archive must be submitted to TEACH before the deadline.** Only the most recent submission to flip will be used for grading. Make sure it is complete!

**Group Activities - 10%**

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 1 or 2 week periods.

#### **Assignments - 30%**

There are 4 longer programming projects. 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

The assignments build on previous assignments and modules to produce a final comprehensive program to demonstrate understanding of the course objectives.

**Programs must compile and run on the EECS server (flip) or they will not be graded. Programs must include a makefile and be archived in a zipfile. The zip archive must be submitted to TEACH before the deadline.** Only the most recent submission to flip will be used for grading. Make sure it is complete!

#### **Final Project - 20%**

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

**Programs must compile and run on the EECS server (flip) or they will not be graded. Programs must include a makefile and be archived in a zipfile. The zip archive must be submitted to TEACH before the deadline.** Only the most recent submission to flip will be used for grading. Make sure it is complete!

#### **Tests and quizzes - 20%**

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. 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.

The tests will not be proctored.

#### **Grading Scale:**

Grade	Score
A	$\geq 92.5$

A-	$\geq 90$
B+	$\geq 87.5$
B	$\geq 82.5$
B-	$\geq 80$
C+	$\geq 77.5$
C	$\geq 72.5$
C-	$\geq 70$
D+	$\geq 67.5$
D	$\geq 62.5$
D-	$\geq 60$
F	less than 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. .

Canvas is used to simply record the grades. The final score displayed is only approximate. At any time if you want a better estimate of you current grade in the course please email the instructor from an OSU email account.

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**.

### Academic Dishonesty

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

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 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 discretion). 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.

Please, read the [department](#), [college](#), and university dishonesty policy, [OAR 576-015-0020 \(2\) Academic or Scholarly Dishonesty](#)

## **Students with Disabilities**

"Accommodations are collaborative efforts between students, faculty and Disability Access Services (DAS). Students 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."

Students with documented disabilities who may need accommodations, who have any emergency medical information the instructor should be aware of, or who need special arrangements in the event of evacuation, should make an appointment with the instructor as early as possible, and no later than the first week of the term. Class materials will be made available in an accessible format upon request.

## **Student Assistance:**

### **Getting assistance:**

- Your first line of assistance should be to take a break, skim through the book, lectures, notes, and Internet,
- If you cannot find the answer yourself after some searching, you should then communicate with your fellow classmates, (remember that I want you to learn the basics in whatever way works best for you!)
- Please contact your TA or me via email to bring your question to our attention if you remain stuck on finding a solution,
- Remember to form study groups,
- We have several methods of communicating, but I would prefer we use a discussion board so that we can refer back to our previous discussions and citations.

**Technical Assistance** — If you experience computer difficulties, need help downloading a browser or plug-in, assistance logging into the course, or if you experience any errors or problems while in your online course, contact the OSU Help Desk for assistance. You can call (541) 737-3474, email [osuhelpdesk@oregonstate.edu](mailto:osuhelpdesk@oregonstate.edu) or visit the OSU Computer Helpdesk online.

(you can also clearly ask in discussion with the class and we can try to work through it for the benefit of the class)

## **Ground Rules for Online Communication & Participation:**

- *Online threaded discussions* are public messages, and all writings in this area will be viewable by the entire class or assigned group members. If you prefer that only the instructor sees your communication, send it to me by email, and be sure to identify yourself and the class.
- Posting of personal contact information is discouraged (e.g. telephone numbers, address, personal website address).
- *Online Instructor Response Policy*: I will check email somewhat frequently and will respond to course-related questions within 24 hours if possible.
- *Observation of "Netiquette"*: All your online communications need to be composed with fairness, honesty and tact. Spelling and grammar are very important in an online course. What you put into an online course reflects on your level of professionalism. Here is a reference
  - netiquette: <http://www.albion.com/netiquette/corerules.html> .
- Please check the Announcements and the course syllabus before you ask general course "housekeeping" questions (i.e. how do I submit assignment 3?). If you don't see your answer there, then please contact someone through chat or discussion boards.
- (Adapted from statements provided by Becky Warner, SOC)

## **Guidelines for a productive and effective online classroom**

- The discussion board is your space to interact with your colleagues related to current topics or responses to your colleague's statements. It is expected that each student will participate in a mature and respectful fashion.
- Participate actively in the discussions, having completed the readings and thought about the issues.
- Pay close attention to what your classmates write in their online comments. Ask clarifying questions, when appropriate. These questions are meant to probe and shed new light, not to minimize or devalue comments.
- Think through and reread your comments before you post them.
- Assume the best of others in the class and expect the best from them.
- Value the diversity of the class. Recognize and value the experiences, abilities, and knowledge each person brings to class.
- Disagree with ideas, but do not make personal attacks. Do not demean or embarrass others. Do not make sexist, racist, homophobic, or victim-blaming comments at all.
- Be open to be challenged or confronted on your ideas or prejudices.

*(Adapted from a statement provided by Susan Shaw, WS)*