University of Washington
Computer Science & Engineering 142: Introduction to Programming I
Course Syllabus, Autumn 2019

Instructor Email Office	Brett Wortzman brettwo@cs.uw.edu CSE 542
Office Hours	Tuesdays 12:00am-1:30pm Wednesdays 1:30pm-3:00pm
Course Website	https://cs.uw.edu/142 — All course materials and resources will be posted here. Check this site frequently for updates.
Lectures	A Lecture: MWF 11:30am – 12:20pm, KNE 130 B Lecture: MWF 3:30pm – 4:20pm, KNE 120
Course Administrator Email Office	Pim Lustig cse142@uw.edu CSE2 173 Pim is your best resource for registration questions, such as switching sections or changing to or from S/NS grading.

Course Overview

This course provides an introduction to programming using the Java programming language. We will explore common computational problem-solving techniques useful to computer scientists, but also to anyone who has large data sets, repetitive processes or other needs for computation. No prior programming experience is assumed, although students should know the basics of using a computer (e.g., using a web browser and a text editor) and should be comfortable with math through Algebra 1. Students with significant prior programming experience should consider skipping CSE 142 and taking CSE 143 (we allow this without any special permission) or CSE 143X.

Inclusion

All students are welcome in CSE142 and are entitled to be treated respectfully by both classmates and the course staff. We strive to create a challenging but inclusive environment that is conducive to learning for all students. If at any time you feel that you are not experiencing an inclusive environment, please contact the course staff or the CSE academic advisors. You should feel free to email any member of the course staff or the advisors at any time, and anonymous feedback can be sent to the course staff via the form linked on the course website.

Accommodations

Please refer to university policies regarding accommodations for purposes of disability or religion. Accommodations must be formally requested through the proper channels according to the relevant policies.

Lectures

Lectures will be the place where you are first introduced to new concepts. It is not expected that will have mastered the material by the end of lecture because mastery requires practice. To encourage students to participate actively in their learning during lectures, we will be using PollEverywhere polls in class as opportunities to solve problems that help you check your understanding; these polls are **not** graded on participation or correctness.

Students are expected to keep talking and electronics usage to a minimum during lecture. You are welcome to use a laptop or tablet to take notes or code along, but please remain on task. If you expect to use your device in ways not related to the course, please sit only in the last four rows of the classroom. This is to ensure a productive learning environment for all the students in the classroom. TAs will periodically enforce this policy during lecture.

Discussion Sections

You will be expected to participate in a Thursday discussion section, held at various times (see course web site for details). The TA who runs your section will grade your homework assignments. In section, we will answer questions, go over common errors in homework solutions, and discuss sample problems in more detail than in lecture.

You will receive up to **3 points** for each section you participate in, up to a maximum of 20 points. Two points are awarded for being present and participating in section each week. The third point is awarded for completed short homework problems that are due at the **start** of each week's section. Homework will not be accepted late for any reason and cannot be submitted electronically. You cannot earn the one point for completing homework if you do not attend section that week. Section homework is expected to take no more than 60 minutes each week and is graded on effort and completion. Total section points are worth about as much as one programming assignment.

Getting Help

Please don't be afraid to ask for help if you don't understand something. Brett holds office hours multiple times a week, and the TAs staff the IPL (see below) for most of the day Monday through Friday and a few hours on weekends. At office hours (or in the IPL), you can ask for a review or explanation of lecture material, clarification of a homework specification, help with a frustrating part of the assignment, or many other things. You can even show up just to tell us you're frustrated and ask for emotional support! Here are some first steps on how to get help:

- Post on Piazza asking a question
- Ask someone on course staff questions before/after lecture, before/after section, etc.
- Come to Brett's office hours
- Visit the IPL

The IPL

The department operates an Introductory Programming Lab (IPL) in **room 334 of Mary Gates Hall**. TAs will be available at the lab to help students with problems. It is open almost every day of the week, and you can find a link to the exact schedule on the course website.

Grading and Assessment

Grades for CSE 142 are calculated based on three categories, weighted as follows:

45%	programming assignments, written reflections, and section participation	
20%	midterm exam	Friday, May 10th (tentative)
35%	final exam	Wednesday, June 12th (tentative)

Final course grades are converted to the 4.0 scale roughly as follows:

90%: at least 3.5	80%: at least 2.5	70%: at least 1.5	60%: at least 0.7

Homework consists of weekly programming assignments completed individually and submitted electronically from the course web site. Programs will be graded on both "external correctness" (behavior) and "internal correctness" (adherence to design, usage, and style guidelines) and are generally graded on a 20-point scale. Disputes about homework grading must be made within 2 weeks of receiving the grade using the process described on the course web site.

Exams (both the midterm and final exam) are closed-book and closed-note. You may not bring any written material, but a standard "cheat sheet" will be provided as part of the exam. No electronic devices, including calculators or smart devices, may be used.

Students are expected to take exams as scheduled excepted in extreme circumstances. Requests for an alternate exam due to known conflicts (e.g. another class or a work obligation) must be made before the deadline posted on the course website to be considered. Students in emergency circumstances (including illness or injury) must contact Brett ***before the exam*** to explain the situation and provide clear documentation of their inability to take the exam. Make-up exams will not be given except in cases of a serious emergency. No make-up or alternate exams will be granted for personal reasons (e.g. travel) under any circumstances.

Reflection assignments will be given for each homework assignment (post-reflection) and for both exam (pre-reflection for both, post-reflection for midterm). Reflections are short written assignments that will help you engage in metacognition and assess the strengths and weaknesses in both your understanding and your process. These assignments will be graded entirely on effort and completion. You will earn 2 points per completed reflection, up to a maximum of 20 points.

Late Days

Each student receives **5 "late days"** for use on programming assignments. A late day allows you to submit an assignment up to 24 hours late without penalty. For example, you could use 2 late days and submit an assignment due Tuesday 9pm on Thursday by 9pm with no penalty. Once a student has used up all their late days, each successive day that an

assignment is late will result in a loss of 1 point on that assignment. Regardless of how many late days you have, **you may not submit an assignment more than 3 days after it is due** or after the last day of class. *Students will not be given extensions except under extenuating circumstances. Students requesting an extension must contact the instructor as soon as possible, and before the "lock date" of the assignment, to explain the situation.*

Academic Integrity and Collaboration

Programming assignments must be completed **individually**; all code you submit must be your own work. You may discuss general ideas of how to approach an assignment, but never specific details about the code to write. Any help you receive from or provide to classmates, tutors, friends, or other outside individuals should be limited and should never involve details of how to code a solution. **In particular, violations of this policy include, but are not limited to:**

- Working as a partner with another student on an assignment.
- Showing another student your solution to an assignment, or looking at another student's solution (past or present), for any reason.
- Having another person "walk you through" an assignment, describe in detail how to solve it, or sit with you as you code; or providing such help to another student. This includes current or former students, tutors, friends, TAs, paid consultants, people on the Internet, or anyone else.
- Copying *any amount* of code (even just a few lines or a single method) from anywhere other than resources explicitly allowed by the instructor.
- Posting your homework solution code, in part or in whole, online to ask others for help. This includes public message boards, code repositories, forums, file sharing sites and services, or any other online system.

Under our policy, a student who gives inappropriate help is equally guilty as one who receives it. Instead of providing such help to someone who does not understand an assignment, please refer them to class resources such as lecture examples, the textbook, the IPL, or a TA or instructor. You must not share your solution and ideas with others. You must also ensure that your work is not copied by others, such as by making sure to log out of shared computers, not leaving printouts of your code in public places, and not emailing your code to other students or posting it on the web.

If you are retaking the course, you may resubmit a previous solution unless that program was involved in an academic misconduct case. If misconduct was found, you must write a new version of that program.

We enforce this policy vigorously by running similarity detection software a few times per quarter over all submitted student programs, including programs from past quarters. Students who violate the policy are offered reduced scores and sometimes sent to a University committee. **This can lead to marks on permanent academic records**. Generally, several dozen students each quarter are given reduced scores for violating these policies. Please be careful, and please contact the instructor if you are unsure whether a particular behavior falls within our policy.