Computer Science & Engineering 143

Computer Programming II

Instructor: Jeremy Lipschutz

Email: jeremyl2@cs.washington.edu

Office Hours: Mondays 12:30-1:30PM, Wednesdays 2:00-3:00PM

Lecture Time: MWF 10:50-11:50AM

Course Administrator: Pim Lustig (email cse143@uw.edu for registration issues)

Textbook
Building Java Programs, 5th edition, Reges & Stepp, recommended. The fifth edition has new material that is relevant to the CSE143 class.

Course Overview
This course is a continuation of CSE142. While CSE142 focused on control issues (loops, conditionals, methods, parameter passing, etc), CSE143 focuses on data issues. Topics include: ADTs (abstract data types), stacks, queues, linked lists, binary trees, recursion, interfaces, inheritance and encapsulation. The course also introduces the notion of complexity and performance tradeoffs in examining classic algorithms such as sorting and searching and classic data structures such as lists, sets and maps. The course will include a mixture of data structure implementation as well as using off-the-shelf components from the Java Collections Framework. The prerequisite is CSE142 or equivalent.

Discussion Sections
You will be expected to participate in two weekly 60-minute discussion sections. We will be using Zoom software to set up a chat room for each section. The TA who runs your discussion section will grade your homework assignments. In section we will answer questions, go over common errors in homework solutions and discuss sample problems in deeper detail than we can in lecture.

Course Registration
To add the class or switch sections, email cse143@uw.edu.

Course Web Page
Information about the course will be kept at https://cs.uw.edu/143. Links to course handouts will be kept on this page along with useful links to other class resources.

Religious Accommodations
See Religious Accommodations Policy
(https://registrar.washington.edu/staffandfaculty/religiousaccommodations-policy).

Computer Access/Software
You can use any Java environment you want although the recommended software is the Java Development Kit (JDK) version 11 or higher and the jGRASP editor. More information can be found on the class web page under the "Java Software" link

Required course work
There will be four categories of required course work:

Take-home assessments (~Weekly, 7 total)

Longer programming assignments that will assess your mastery of the skills and concepts covered in class that week. While Take-home assessments will emphasize content covered in the week preceding the assessment's release, they will often integrate content from earlier in the quarter as well. Assessments will generally be due on Thursdays and must be completed independently, though you may discuss your approach with classmates.

Checkpoints (Weekly, 7 total)

Short assignments consisting of a few problems to give you practice with the content covered in class that week. Checkpoints will generally be due on Sundays and may be worked on collaboratively with classmates.

Simulated Midterm (1 total)

Large, cumulative assignments consisting of a series of problems requiring you to engage with all course content covered to that point. The simulated midterm will include three parts: (1) completing a timed set of problems to the best of your ability; (2) reviewing your work against provided solutions and annotating areas for improvement; and (3) meeting 1-on-1 with your TA to discuss your results and your progress in the course. Culminating assessments must be completed independently.

Reflections (w/other assignments 7 total)

Brief, written assignments designed to help you engage in metacognition and reflect on your current progress and understanding. Each reflection will be directly tied to another assignment, usually a take-home assessment, and will be due along with that assignment.

Revision and Resubmission

Learning from mistakes is an important part of mastering any skill, especially for novices. To enable this, you are allowed to revise and resubmit your work on take-home assessments to demonstrate improved mastery after your initial submission. Resubmissions are subject to the following rules:

- You may not resubmit a take-home assessment for which you did not make an initial submission, or for which your initial submission did not pass the verification tests.
- You may not resubmit a take-home assessment until you have received feedback on your previous submission of that assessment. (Generally one week after the due date.)
- Resubmissions must be accompanied by a short write-up describing the changes made. This will both support you in being deliberate about the changes you make and ease grading of resubmissions by making the changes clear.
- A maximum of one take-home assessment can be resubmitted each week.
- All resubmissions must be made no later than Saturday August 21, 11:59PM
- A take-home assessment that has been found to involve academic misconduct may not be resubmitted (see below).

Resubmissions will be graded, and **the new grades will fully replace the previous grades**. The new grade will be based entirely on the resubmitted work, meaning that your grades *may go down* when resubmitting. In addition, while every effort is made to identify all areas that could be improved when grading, feedback is not guaranteed to be exhaustive. Be sure to consult all available resources and materials to ensure your work meets all guidelines.

Only take-home assessments may be resubmitted.

Late Work

You **MUST** make an initial submission for each take-home assessment by the published deadline to receive a grade other than **U** on that assessment (see grading scale below). Even if your work is incomplete, you are

highly encouraged to submit whatever work you have so that you can receive feedback. You will then be able to act on that feedback and resubmit (see the resubmission policy above).

Checkpoints and culminating assessments will generally not be accepted after their published deadlines.

Extenuating Circumstances

We recognize that our students come from varied backgrounds and can have widely-varying circumstances. We also know that the unusual circumstances of this quarter, and the current state of the world, may bring unique challenges for everyone. If you have any unforeseen or extenuating circumstance that arise during the course, please do not hesitate to contact the instructor in office hours, via email, or through a private Ed Discussion post to discuss your situation. Extenuating circumstances can include work-school balance, familial responsibilities, military duties, unexpected travel, or anything else beyond your control that may negatively impact your performance in the class. Please make every effort to contact us as soon as possible once you are aware of the circumstances. **The sooner we are made aware of the situation, the more easily it can be resolved and the more likely we will be able to offer flexibility.**

Grading scale

Work in CSE 143 will be graded using an ESNU scale:

E (Exemplary)
Work that meets all requirements and displays full mastery of *all* learning goals and material.

S (Satisfactory)
Work that meets all requirements and displays at least partial mastery of all learning goals as well as full mastery of core learning goals.

N (Not yet)
Work that does not meet some requirements and/or displays developing or incomplete mastery of at least some learning goals and material.

U (Unassessable)
Work that is missing, does not demonstrate meaningful effort, does not provide enough evidence to determine a level of mastery, or violates major course rules (such as using forbidden features or violating the academic conduct policy).

Take-home assessments will be graded using the full ESNU scale. Checkpoints, reflections, culminating assessments will be graded using only SN; no grades of E or U will be awarded on these assignments.

It is important to note that, under this system, it is the *work* that is assessed as a proxy for the student. This is an imperfect system, but is necessary to manage a course of the size and scale of CSE 143. It is in your best interest to ensure that your work accurately reflects your mastery by being careful and diligent in following instructions, meeting deadlines, and understanding requirements.

Take-home assessment grading
Take-home assessments will be graded on four dimensions:

Behavior
Does the input and output functionality of the submission conform to the specification?

Abstraction and Design
Are data structures, methods, and classes chosen, designed, and implemented according to best practices?

Use of Concepts
Are all language features and course concepts used appropriately, effectively, and efficiently?

Documentation and Readability
Is the code well-written and well-documented, and does it conform to all code quality and other related guidelines?

Work will be assigned a grade on each dimension using the ESNU scale, resulting in *four* ESNU grades per assessment. The linked table summarizes the expectations for each grade in each dimension.

Final grade assignment

Final grades will be based on the quantity of work completed at each level of mastery, based on the following minimums:

at least 3.5

- **S** or better on all dimensions on all take-home assessments, including at least 5 total **E**'s on each dimension
- **S** on at least 6 checkpoints
- **S** on the culminating assessment
- **S** on at least 6 reflections

at least 2.5

- At least 26 total **S**'s or better and no **U**'s on take-home assessments
- **S** on at least 5 checkpoints
- **S** on the culminating assessment
- **S** on at least 3 reflections

at least 1.5

- At least 16 total **S**'s or better on take-home assessments
- **S** on at least 3 checkpoints
- No culminating assessment required
- No reflections required

at least 0.7

- At least 14 total **S**'s or better on take-home assessments
- **S** grades on at least 2 checkpoints
- No culminating assessment required
- No reflections required

To be guaranteed a particular minimum grade, *all* requirements for that grade must be met. Failing to meet any of the requirements for a particular minimum grade does not preclude the possibility of receiving that grade, but we do not provide any estimates or guarantees beyond those listed here.

Exact final grades, including all grades not listed above, will be determined at the end of the quarter by the course staff based on each student's overall body of work. Estimates of students' final grades beyond the requirements listed above will not be provided.

Policy on Collaboration

You are to complete programming assignments individually. You may discuss the assignment in general terms with other students including a discussion of how to approach the problem, but the code you write must be your own. The intent is to allow you to get some help when you are stuck, but this help should be limited and should never involve details of how to code a solution. **You must abide by the following**:

- You may not work as a partner with another student on an assignment.
- You may not show another student your solution to an assignment.
- You may not have another person (current student, former student, tutor, friend, anyone) "walk you through" how to solve an assignment.
- You may not post your homework solution code online to ask others for help. This includes public message boards, forums, file sharing sites and services, or any other online system
- You are not to examine online solutions that you might find on the web.

Under our policy, a student who gives inappropriate help is equally guilty with one who receives it. Instead of providing such help, refer other students to class resources (lecture examples, the textbook, the IPL, or emailing 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 by not leaving it in public places, emailing it others, posting it on the web, etc.

If you are taking the course a second time, you are allowed to submit a previous solution that you authored unless that program was involved in a case of academic misconduct. For any assignment where academic misconduct was involved, you must write a new version of the program. We enforce this policy by running similarity-detection software over all submitted student programs, including programs from past quarters.