Computer Science & Engineering 143
Computer Programming II

Instructor: Stuart Reges
Email: [redacted]
Phone: [redacted]
Office: Gates Center (CSE2), room 305
Office hours: Tuesdays 1:30-3:30

Course Administrator
Pim Lustig
Email: [redacted]
(email for registration issues)

Lecture Times
A: MWF 12:30-1:20, Kane 120
B: MWF 2:30-3:20, Kane 120

Textbook
Building Java Programs, 5th edition, Reges & Stepp, required.

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.

Lecture Policy
In the lecture room students should keep talking to a minimum and are limited in their use of electronic equipment. Students who want to use cell phones or laptops will be required to sit in the last four rows of the classroom. If it is important to you to use your laptop during lecture, email Stuart to describe your situation and request an exception. TAs will periodically enforce this policy during lecture.

Discussion Sections
You will be expected to participate in two weekly 50-minute discussion sections. 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 more detail than we can in lecture.

Grading
You will be expected to complete a variety of programming assignments for this course and to take two exams. The resulting scores will be combined according to the following weightings:

40%	weekly homework assignments (generally graded on a 20-point scale)
20%	midterm (Friday, 2/11/22)
40%	final exam (date and time not yet determined)

Using the weightings above, each student's scores will be turned into an overall score ranging from 0 to 100 percent. These will be turned into grades as follows:

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

The exams will be closed-book and closed-note. If you need to miss an exam, you must contact Stuart *prior* to the exam to get permission. Even if you are sick, you should be able to call your instructor's office phone and leave a message that you need to be contacted. Students wishing to take an exam at the DRS testing facility must schedule their exam at least three weeks in advance of the exam or they may not be accommodated.

Course Registration
To add the class or switch sections, email [redacted]

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/religious-accommodations-policy/).

Indigenous Land Acknowledgement

I acknowledge that by the labor theory of property the Coast Salish people can claim historical ownership of almost none of the land currently occupied by the University of Washington..

Computer Access/Software

The school operates an Introductory Programming Lab (IPL) that is located on the third floor of Mary Gates Hall. TAs will be available at the lab to help students with problems. You can use any Java environment you want although the recommended software for this course is the Java Development Kit (JDK) version 8 or higher and the jGRASP editor. More information can be found on the class web page under the "Java software" link.

Late Policy

Each assignment will list its due date. Most will be due on Thursdays at 11 pm. Each student in the class will have a total of eight "free" late days (a late day is 24 hours of lateness). There are no partial days, so assignments are either on time, 1 day late, 2 days late, etc. Because of this generous late policy, students will not be granted extensions for assignments unless they have some highly extenuating circumstances. Once a student has used up all of his or her late days, each successive late day will result in a loss of 1 point. No assignment will be accepted more than 4 days after its due-date. If you are experiencing a problem that makes it difficult for you to complete an assignment on time, you should contact Stuart by email as early as possible to request an extension.

We will grade only one version of any given program. If you make multiple submissions for an assignment, we will grade the last version submitted. If you submit a version that you later decide you do not want to have graded, you must warn your TA not to grade that version and to wait for a later submission from you.

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, TA office hours, or emailing a TA or instructor). 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 have to 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.