

# CSE 143: Computer Programming II

## Syllabus, Summer 2015

### Information

Instructor:	
Name:	Riley Porter
E-mail:	rileymp2@cs.uw.edu
Office:	CSE 214
Office Hours:	Wed: 12:00pm – 2:00pm

Course Website:
<a href="http://cs.uw.edu/143">http://cs.uw.edu/143</a>

Lecture Time:
Bagley 131 on MWF 10:50 AM - 11:50 AM

<b>Textbook:</b> Reges/Stepp, <i>Building Java Programs (3rd Edition)</i>
---

### Course Overview

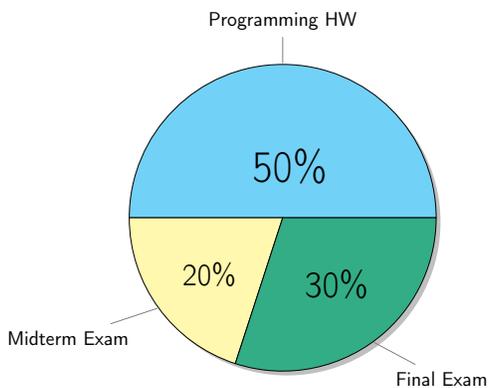
This course is a continuation of CSE 142.

Prerequisite: CSE 142 or equivalent.

While CSE 142 focused on control issues (loops, conditionals, methods, parameter passing, etc.), CSE 143 focuses on data issues. Topics include: abstract data types (ADTs), lists, stacks, queues, linked lists, binary trees, recursion, interfaces, inheritance, and encapsulation. The course also introduces the notion of complexity and performance trade-offs 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 implementing data structure and using components from the Java Collections Framework.

### Assessments

Every homework we give you has a very important purpose to your understanding of the material. Here's a handy pie chart that explains how your grade will be calculated:



This maps to the 4.0 scale roughly as follows:

90%: at least 3.5  
75%: at least 2.0

85%: at least 3.0  
70%: at least 1.5

We might make these cutoffs more lenient.

80%: at least 2.5  
60%: at least 0.7

**Programming HW** consists of weekly, individual assignments. Programs will be graded on “external correctness” (behavior) and “internal correctness” (style and design). Disputes about homework grading must be made using the grading complaint form within 2 weeks of receiving the grade. We *do not* drop any homeworks.

**Exams** will be closed-book and closed-note. You may not use the textbook, notes or any other written material during exams. No electronic devices may be used, including calculators.

**Make-up Exams** will not be given except in case of a serious emergency. If you expect to be accommodated, you must contact Riley before the exam (even due to injury or sickness). You must provide evidence that you are physically unable to take the exam (e.g., a clear doctor’s note mentioning the date and reason). No special accommodations will be made for students who arrive late to exams, regardless of reason. For non-medical excuses, you must contact Riley within the first two weeks of class.

## Late Days

Each student receives 5 “late days” for use on homework assignments. A late day allows you to submit a program up to 24 hours late without penalty. For example, you could use 2 late days and submit a program due Tuesday 11:30pm on Thursday by 11:30pm with no penalty. Once a student has used up all the late days, each successive day that an assignment is late will result in a loss of 2 points on that assignment. Regardless of how many late days you have, **you MAY NOT submit a program more than THREE days after it is due** or after the last day of class. Students will not be given extensions unless they have extenuating circumstances as decided by the instructors.

## Getting Help

Please don't be afraid to ask for help if you don't understand something.

At office hours (or the IPL, see below), you can ask for clarification on a lecture (or for a *repetition* of the lecture!). You can ask for help with a frustrating part of the homework. You can also ask questions about what other relevant material to explore in computer science. You can even show up just to tell us you're frustrated and vent.

Here's some first steps on how to get help:

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

## What/Where/When is 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 every day of the week, and you can find a link to the exact schedule on the course website.

## Academic Integrity

Programming HW 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 should be limited and should never involve details of how to code a solution. It should also not be in front of a computer.

Here's some do's and don'ts:

- DO NOT work as a partner with another student on an assignment.
- DO NOT show (or send!) you solution to an assignment (or look at someone else's solution) for any reason at any time. EVER.
- DO NOT sit with someone (current or former students, tutors, friends, TAs, paid consultants, people on the Internet, or anyone else) and “walk through” an assignment, or describe in detail how to solve it.
- DO post questions about specification on Catalyst.
- DO NOT post any piece of solution code anywhere online (on Catalyst, on message boards, forums, file sharing sites and services, or any other online system).

Please note that *both parties* involved in a cheating case are treated equally. That is, helping someone else cheat is *just as bad as reading someone else's solution*. There are many resources for students who are struggling, and the right thing to do in this situation is to point them to these resources.

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 several times per quarter over submitted student programs. Students who violate the policy are offered reduced scores and sent to a University committee. This leads to marks on permanent academic records. If you're not sure if you're following the policy, ask.