Welcome to CSE 121!

Elba Garza

Winter 2023
Agenda

• About me
• About this course
  • Learning objectives
  • Other similar courses
  • Course components
• Our learning model

• Tools and resources
  • Course Website
  • Ed
• Assessment and grading
• Collaboration
Agenda

• About me
  • About this course
    • Learning objectives
    • Other similar courses
    • Course components
  • Our learning model

• Tools and resources
  • Course Website
  • Ed
  • Assessment and grading
  • Collaboration
Hi, I’m Elba! (she/her)

• Newly minted Assistant Teaching Professor
• PhD in CS (Computer Architecture)
• Favorites:
  • Ice cream flavor: lavender
  • Movie: In the Mood for Love
  • Number: 27
Meet (most of) your 28 TAs
Agenda

• About me
• About this course
  • Learning objectives
  • Other similar courses
  • Course components
• Our learning model

• Tools and resources
  • Course Website
  • Ed
• Assessment and grading
• Collaboration
Learning Objectives

or, “What will I learn in this class?”

• **Functionality/Behavior:** Write functionally correct Java programs that meet a provided specification and/or solve a specified problem

• **Functional Decomposition:** Break down problems into subproblems that are modular and reusable, and define methods to represent those subproblems

• **Control Structures:** Select and apply control structures (e.g. methods, loops, conditionals) to manage the flow of control and information in programs

• **Data Abstraction:** Select and apply basic data abstractions (e.g. variables, parameters, arrays, classes) to manage and manipulate data in programs

• **Code Quality:** Define programs that are well-written, readable, maintainable, and conform to established standards
 Other Similar Courses

<table>
<thead>
<tr>
<th>Course</th>
<th>Good choice if…</th>
</tr>
</thead>
</table>
| CSE 121 | • You’ve never programmed before AND  
          • You are, or want to be, in a major such as CS, CE, ECE, Info, etc. that requires Java programming |
| CSE 122 | • You’ve done some programming (roughly one course worth) in any programming language AND  
          • You are, or want to be, in a major such as CS, CE, ECE, Info, etc. that requires Java programming |
| CSE 123 | • You’ve taken CSE 123 AND  
          • You are, or want to be, in a major such as CS, CE, ECE, Info, etc. that requires Java programming |
| CSE 160 | • You’ve never programmed before AND  
          • You’re interested in data science and analysis OR  
          • You’d rather learn Python than Java* OR  
          • You are, or want to be, in a major such as Physics, Bio, Stat, etc. where analyzing data through programming is useful |

Other courses of interest: CSE 154, CSE 163

See Guided Self-Placement and Introductory Courses for more info
Help Us Improve!

• CSE 121 is brand new!
• We worked hard to build a course we think will be effective and supportive and help you succeed
• We probably didn’t get it all right

• We appreciate your patience and understanding if we need to make adjustments during the quarter
• Please give us lots of feedback!
  • Post on Ed and/or use the Anonymous Feedback Tool
Course Components

Lessons (aka Lectures)
• WF, 11:30 or 2:30
• Held live in KNE 210/GUG 220; recordings released afterward
• First introductions to course concepts
• Mix of presentation of content and practice activities/problems
• Pre-work for most sessions

Sections
• TuTh, various times
• Led by TAs
• Held live in person; not recorded
  • Materials will be released online afterwards
• Additional review, discussion, and practice
• Mostly practice problems

Attendance is not taken, but you are responsible for all material (including announcements).
Agenda

• About me
• About this course
  • Learning objectives
  • Other similar courses
  • Course components
• Our learning model

• Tools and resources
  • Course Website
  • Ed
• Assessment and grading
• Collaboration
Digression: Macarons
Digression: Macarons
Digression: Macarons

Actually…
Course Culture and Support

• Currently 566 students enrolled!
  • Wide range of backgrounds, interests, and goals
  • Everyone is new to programming
• Support and help each other!
  • Form study groups
  • If you have a question, others almost certainly do too
• Lots of ways to get support from us
  • Ed message board, Introductory Programming Lab (IPL), section
Course Culture and Support

• Policies designed with flexibility in mind
  • Resubmissions/Retakes, lecture recordings

• But life and the world still happen…

• Please reach out ASAP if you’re struggling or have circumstances that require extra support
Learning in CSE 121 (or anywhere)

- **Exposure**
  - Lessons, videos, textbook
  - Encounter concepts for the first time. See examples and ask questions. Nowhere near mastery!

- **Independent/Group Practice**
  - Checkpoints, section problems, additional practice
  - Practice on your own or with classmates. Continue to learn by doing. Get close to mastery.

- **Guided Practice**
  - Lesson activities, sections, labs
  - Practice with support from course staff. Learn by doing: make mistakes and learn from them. Start to develop mastery.

- **Assessment**
  - Take-home assessments
  - Build on the scaffolding from 1-3. Still learning by doing: you're not done! Demonstrate your mastery (even if it's still developing).
Agenda

• About me
• About this course
  • Learning objectives
  • Other similar courses
  • Course components
• Our learning model

• Tools and resources
  • Course Website
  • Ed
• Assessment and grading
• Collaboration
Course Website

cs.uw.edu/121

- Primary source of course information (not Canvas)
- Calendar will contain links to (almost) all resources
Please review the syllabus ASAP.
Ed

• Our online learning platform
• Lessons, sections, quizzes all here
• Intro and walkthrough in Section 0
Agenda

• About me
• About this course
  • Learning objectives
  • Other similar courses
  • Course components
• Our learning model

• Tools and resources
  • Course Website
  • Ed
• Assessment and grading
• Collaboration
Assessment and Grading

• Our goal in the course is for you to **gain proficiency of the concepts and skills** we teach

• We assess your proficiency by asking you to apply the concepts and skills on tasks or problems

• By necessity, we are assessing your **work** as a proxy for your proficiency
Assignments

• Your learning in this course will be assessed in four ways:
  • Programming Assignments (~biweekly, 4 total)
    • Structured programming assignments to assess your proficiency of programming concepts
  • Creative Projects (~biweekly, 4 total)
    • Smaller, more open-ended assignments to give you space to explore
  • Quizzes (3 total, in section)
    • Series of problems covering all material up to that point
  • Final Exam (Tuesday, March 14th)
    • Final, culminating assessment of all your skills and knowledge
Resubmission/Retakes

Learning takes time, and doesn’t always happen on the first try!

• One previous Programming Assignment or Creative Project can be resubmitted each week
  • Must be accompanied by a write-up describing changes
  • Grade on resubmission will replace original grade
  • Each assignment should only be resubmitted once!
• Each Quiz can be retaken once
  • Retakes scheduled at certain times– details forthcoming
  • Best per-problem policy, so you can focus on what you missed

• See the syllabus for more details
Grading

*Grades should reflect your proficiency in the course objectives*

- All assignments will be graded **E (Excellent), S (Satisfactory), or N (Not yet)**
  - Under certain circumstances, a grade of U (Unassessable) may be assigned
  - In some cases, not all grades will be given
- Final grades will be assigned based on the **amount of work at each level**

- See the [syllabus](#) for more details
Grading

1. **Base grade**: Identify the highest minimum grade for which the student meets all requirements.

2. **Additional S's and E's**: Count the number of each of the following earned beyond the requirements for the base grade identified in the last step:

3. **Adjustment**: Multiply each counts by a multiplier and add the result to the base grade. Each count may use a different multiplier. Multipliers will be determined at the end of the quarter.
1. **Base grade**: Identify the highest minimum grade for which the student meets all requirements.

2. **Additional S's and E's**: Count the number of each of the following earned beyond the requirements for the base grade identified in the last step:

3. **Adjustment**: Multiply each count by a multiplier and add the result to the base grade. Each count may use a different multiplier. Multipliers will be determined at the end of the quarter.
1. **Base grade**: Identify the highest minimum grade for which the student meets all requirements.

2. **Additional S’s and E’s**: Count the number of each of the following earned beyond the requirements for the base grade identified in the last step:

3. **Adjustment**: Multiply each count by a multiplier and add the result to the base grade. Each count may use a different multiplier. Multipliers will be determined at the end of the quarter.
Agenda

• About me
• About this course
  • Learning objectives
  • Other similar courses
  • Course components
• Our learning model

• Tools and resources
  • Course Website
  • Ed
• Assessment and grading
  • Collaboration
Collaboration Policy

*Learning is hard, but it’s easier when you learn from each other*

- You are encouraged to form study groups, work together on practice and review, and discuss your ideas and approaches *at a high level*
- If you discuss your ideas with others, you must *cite them*
- All work you submit for grading *must be predominantly and substantially your own*
- Work that violates policy may be withdrawn

- See the [syllabus](#) for more details