CPSC S112 Introduction to Programming

Summer 2025 – Session B

Course Description

CS112 is an introductory programming course, using the Java programming language. There is no prerequisite for the course -- in particular, no prior programming experience is required.

By the end of the course, you should be able to write useful Java programs, but the focus of the course is programming concepts that can be applied to other languages as well. Throughout the course you will write cool programs demonstrating data types (arrays, strings, numbers, etc.), control structures (conditionals, loops, etc.), program analysis (assertions, etc), basic algorithms (sorting, searching, etc.), object-oriented programming (classes, methods, objects, inheritance, polymorphism, etc.), and basic libraries (graphics, digital audio, text processing, etc.).

Course Format

This course has three live in-person lectures (MWF) each week followed and we will offer smaller discussion sections (TTH). The lectures present the majority of the course content, while discussions reinforce the concepts with walk-throughs and programming exercises. Attendance is expected in the discussion sections and lecture participation will be judged by in-class polls.

Teaching Staff - TBD

	Name	**Email**
Primary instructor	Cody Murphey (he/him)	cody.murphey@yale.edu
Course Assistant		
Course Assistant		

Note that Ed Discussions will be the fastest way to get questions answered outside of office hours!

Office Hours Schedule - TBD

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

Location: TBD

Prerequisites

None.

Required Course Materials

All software used in the course is free. There is not a required text book, but we recommend the following book if you would like to follow along with topics in text.

- Stuart Reges and Marty Stepp. Building Java Programs: A Back to Basics Approach.
  URL: http://www.buildingjavaprograms.com

An additional online book from which we will draw examples and exercises:

- Robert Sedgewick and Kevin Wayne. Introduction to Programming in Java: An Interdisciplinary Approach.
  URL: http://introcs.cs.princeton.edu/java/home/

Assessments and Grading

Learning to program requires a lot of practice. A large part of the course revolves around weekly programming assignments that will build upon skills from prior weeks. These will be time consuming, but they should also be fun. There will be 6-8 assignments, after which you will transition to working on a final group project.

Grade breakdown:

70% programming assignments
20% final project
10% attendance/participation

Late Submission Policy

To allow for the exigencies of computer failures and personal crises, each student has **2 discretionary late days** for homework assignments. These late days can be used for *any reason* and there is no need to request permission to use them.

If all late days have been used up, then assignments may still be submitted up to 2 days late, but they will incur a 10% late penalty per day (5 minutes after the deadline is still considered 1 day late).

To reiterate, whether you have used all late days or not, assignments are not accepted more than 2 days late. The submission will close at 11:59pm 2 days after the due date.

Course Communication

Course announcements and assignments will be handled through Ed Discussion. You are responsible for staying up to date! We will also use Ed Discussions for discussion and Q&A.

Access Ed Discussions using the link in the sidebar on the left.

Rather than using email for questions, you should use this space to discuss the course with classmates and ask questions of the instructors. You may not post answers or code for any assignment, or exam, in whole or in part. You may ask for assistance or provide assistance with homework problems (with respect to understanding what you are required to do), specific language syntax, compilation errors, program environments, topics discussed in class or in the textbook, and other course related matters. You are asked to treat each other with respect. Please do not post or respond with derogatory remarks, or these remarks will be removed at the instructor's discretion.

Accessibility

Our institution values diversity and inclusion; we are committed to a climate of mutual respect and full participation. Our goal is to create learning environments that are usable, equitable, inclusive and welcoming. If there are aspects of the instruction or design of this course that result in barriers to your inclusion or accurate assessment or achievement, please notify the instructor as soon as possible. Students are also welcome to contact Student Accessibility Services to discuss a range of options to removing barriers in the course, including accommodations.

Academic Integrity

The homework assignments in this course are intended to give you practice at working through problems independently. Therefore, unless otherwise specified, the homework assignments are your individual responsibility and are not group assignments. Some assignments will specifically tell you that they allow pair-programming. This allows you to work side-by-side with a partner and submit one solution for both of you. In these cases, each pair must still work independently of other pairs and each student within a pair is responsible for understanding the full program being submitted.

Plagiarism is a violation of University rules and will not be tolerated. You must neither copy work from others (at Yale or elsewhere) nor allow your own work to be copied. This also applies to AI generated work. In addition to grade penalties, additional consequences for breaking this policy

may be imposed by the Yale College Executive Committee. Note that Gradescope will automatically check your submissions for code similarity with your peers and past submissions to similar assignments.

You *may:*

- Ask others or search online for help with general issues with programming languages, APIs, IDEs, tools, and high-level course concepts that are not specific to the assignment.
- Ask clarifying questions about the requirements of an assignment to TAs or on the course discussion board.
- Discuss more specific issues on an assignment with a TA or instructor.

You *may not:*

- Discuss your individual solution with your peers.
- Receive a printed or electronic copy of anyone else's work for the course from this term or any other term. This includes asking or paying someone else to complete the assignment for you.
- Give anyone else a printed or electronic copy of your work for the course for this term or any other term. This includes posting your work publicly on sites such as Github.
- The two items above also include looking over another student's submission on their screen.
- Seek out solutions to similar assignments online.
- Ask ChatGPT or other AI tools to generate your code or written responses.

If an assignment specifically allows group submissions, you may not switch groups after starting work on the assignment unless you obtain permission from the instructor.

If you have any questions about this policy or are unsure if you may have crossed a line, discuss it with the instructor as soon as possible.

List of Topics

The following is an outline of topics for the course:

1. Introduction
    - What is an algorithm?
    - Java program structure
    - Java compilation and execution steps
2. Elements of Programming

- Static methods and method decomposition (Top-down design, bottom-up implementation)
- Data types; Primitive data types; Operators and expressions; Data type conversions; Assignment operators
- Basic for loops; Loops and coding style; Variable scoping; Nested loops; ASCII figures as loop examples
- Parameterized methods
- Java graphics; Examples of parameterized drawings and loops; Animations
- Methods with returns; Summary of static method definition and invocation
- Text I/O: Input using Scanner; Exception handling; File input/output; Text output using printf/format

3. Basic Program Flow of Control and Basic Arrays
    - Logical conditions (Boolean expressions); Boolean type
    - Nested if/else
    - Conditional shortcuts (switch, conditional operators)
    - Strings processing with conditional
    - String as an array of chars; Text encryption/decryption
    - Array of Strings; Using array as counters (histogram)
4. Indefinite Loops using while, do/while
5. General Program Flow of Control: Indefinite Loops

1. Program analysis; Loop pattern: sentinel input/fencepost
2. Loop examples of various patterns

6. General Arrays
    - Array algorithms, Array reference semantics
    - Arrays and digital audio
    - Multidimensional arrays
7. GUI interaction
    - mouse and keyboard inputs
8. Object-oriented Programming (OOP)
    - OOP design and analysis

- Encapsulation using public/private
- OOP relationships and basic UML
- Inheritance; Method overriding
- Event-driven programming (Critters)
- Polymorphism
- Interface

9. Collections/Data structures
    - Generic types
    - ArrayList; HashMap; HashSet
10. GUI Widget Frameworks