CPSC 112: Introduction to Programming

Jay Lim

Summer, 2022

Class Room: TBA
Session B: July 4th - August 5th
Class Hours: M/W/F 1-3:15pm

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, although it helps to be "computer literate."

By the end of the course, you should be able to write useful Java programs, but the focus of the course is programming concepts. Besides learning how to write cool programs, you will focus on learning data types (arrays, strings, numbers, lists, queues, stacks, etc.), control structures (conditionals, loops, exception handling, recursion, etc.), program analysis (assertions, etc), basic algorithms (sorting, searching, etc.), object-oriented programming (classes, methods, objects, inheritance, polymorphism, interface, etc.), and basic libraries (graphics, digital audio, text processing, etc.).

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 in the text.

- Stuart Reges and Marty Stepp. Building Java Programs: A Back to Basics Approach.
- Robert Sedgewick and Kevin Wayne. Introduction to Programming in Java: An Interdisciplinary Approach.

Prerequisites

None.

Assessments and Grading

This course will have between 7-8 weekly Java programming assignments and a final group project.

Grade breakdown:

1. 75% programming assignments
2. 20% final project
3. 5% attendance/participation

Late Submission Policy

To allow for the exigencies of computer failures and personal crises, each student has 5 discretionary late days for homework assignments, but any one assignment may only be up to 1 days late so that we can discuss solutions, return grades in a timely fashion, and move on to the next assignment. These late days can be used for any reason and there is no need to get a Dean's excuse or special permission to use them.

If all 5 late days have been used up, then assignments may still be submitted up to 1 days late, but they will incur a 10% late penalty.

Academic Honesty Policy

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. 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. In addition to grade penalties, additional consequences (Links to an external site.) 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:

1. 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.
2. Ask clarifying questions about the requirements of an assignment to TAs or on the course discussion board.
3. Discuss more specific issues on an assignment with a TA or instructor.

You may not:

1. Discuss your individual solution with your peers.

2. 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.
3. 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.
4. Seek out solutions to similar assignments online.

Course Communication

Course announcements and assignments will be handled through this Canvas site. You are responsible for checking Canvas and staying up to date. Additionally we will use Ed Discussions (in place of Piazza) for discussion and Q&A.

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.

Course Outline

The following is a non-exhaustive list of topics in this course (subject to change).

1. Introduction
    - Introduction
    - Java program structure
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
    - Program analysis; Loop pattern: sentinel input/fencepost
    - Loop examples of various patterns
6. General Arrays
    - Array algorithms, Array reference semantics
    - Arrays and digital audio; Multidimensional arrays
    - PageRank
7. Object-oriented Programming (OOP)
    - User-defined type: encapsulation of data+code
    - OOP design and analysis
    - The OOP Encapsulation Principle
    - OOP examples: Random vs Math.Random; StdDraw vs DrawingPanel; Complex numbers and fractal graphics
8. Class Relationship; Inheritance/Polymorphism
    - Class relationship: Composition, association, inheritance
    - Inheritance and object construction; Method override
    - Inheritance hierarchy (Critters hierarchy as an example)
    - Event-driven programming (Critters with coordination)
    - Polymorphism
    - Interface
9. Programming with generic types: Sorting, searching
10. GUI Frameworks