SJSU SAN JOSÉ STATE UNIVERSITY

College of Science · Computer Science

Introduction to Programming Workshop

CS 46AW

Fall 2026 Section 01 In Person 1 Unit(s) 08/19/2026 to 12/07/2026 Modified 08/26/2026

Contact Information

Instructor: Prof. Navrati Saxena

All course-related correspondence should go to Chase Potter (chase.potter@sjsu.edu), who is the Lab Instructor.

Course Information

Lab

Friday, 11:00 AM to 1:50 PM, SCI 311

Course Description and Requisites

Designed to help all students excel in Introduction to Programming. Students work in groups on challenging problems to help them understand programming concepts more deeply and lay the groundwork for success in future courses.

Repeatable for a total of 2 units.

Prerequisite(s): CS 46A (in an object-oriented language) or CS 46AX, or current enrollment in CS 46AX.
Grading: Credit/No Credit.

Classroom Protocols

Class Modality

This is an in-person class.

Tips for Succeeding

Why CS46AW? To learn your first programming language, you need to overcome several barriers:

1. You need to understand the organization and logical steps required to accomplish whatever task you are supposed to solve with your program. You have learned many "algorithms" over your life: something like the process to multiply two long numbers against each other is an algorithm. But, in programming, sometimes we are expected to come up with an algorithm to solve a problem, rather than just following an algorithm. That is, rather than being asked to follow a given algorithm to solve an instance of a problem, you are asked to come up with an algorithm that can solve all instances of some problem. Consider: if you didn't know the algorithm for long multiplication, inventing that method would be much more difficult than just following it.
2. You need to learn the syntax of the programming language (Java here) in order to be able to tell the computer how to follow that logic. Like any language, small changes in a program can greatly change what it accomplishes, but it tends to be more extreme: in a Java program, one misplaced character might change a perfectly working program into one that is faulty, or one that the computer can't even recognize as a program.
3. Because these first two tasks are so difficult, many tools have been developed to help you program. To help with the fact that the syntax can be so finicky, your Integrated Development Environment (IDE) can highlight the parts of your code that are not legal Java syntax. It can even make suggestions to you for how to fix your code. And, once the program is in running order, they can help you to find errors in your logic. These tools have grown in usefulness and sophistication over the years, and will help to make your time coding much more fruitful...eventually. Unfortunately, like many tools, it takes some familiarity with them before they start to really pay off. And, while you are already struggling with all of the new material from the two issues above, adding new tools to the mix can just make programming seem overwhelming at first.

Because of this, the start-up learning curve for programming can be quite steep. Consider this: some students program their first line of code in college, and less than 4 years later, they are able to get jobs as professional programmers. That is quite a quick ramp-up from novice to professional. I suspect there aren't too many music majors who take up their first instrument in college, expecting to be professional musicians 4 years later.

So, why do students get discouraged? Because, at first, it is difficult. They don't get enough practice. They think that they can't do it. And, sometimes, they compare themselves to other students in the class and feel like they don't measure up. Oftentimes, that is a very unfair measurement: many students in CS46A have programming experience. Even if it was in a different language, frequently, the issues from the first point above are language independent. So, even if they programmed in a different language, they have already seen much of the logic, and now only have to learn the syntax. And tools...except they have likely already seen some similar tools too. And, some of them have done this in Java: maybe they were introduced to Java in the CS Principles AP, which doesn't give credit for CS46A. Maybe they took a CS A AP course, which would give credit for CS46A, but they were unable to take the exam, or didn't score well enough on the exam to get credit. In any of those cases, it is not reasonable to compare your knowledge, six weeks into your first programming class, with the knowledge of students who have seen it before, last year.

We have developed CS46AW specifically to give students with less programming experience extra opportunities to practice computational thinking, and extra chances to do lots of simple programs. Like most things in life, we get better at programming by practicing. For CS46AX, much of that practicing will

be at home, on your own, but in CS46AW, you will get the chance to work on more problems, with peers and supervision, to help keep you from getting stuck. Students with programming experience are certainly allowed to take CS46AX/AW instead of CS46A, but our expectation is that this extra practice will be most beneficial to students who haven't programmed before.

Effective Study Habits

- Engage with the instructor's examples during class.
- Make use of additional references and resources.
- Seeing somebody else code can help to teach you something about coding, but it will not make you a good coder. Only writing your own code can do that. Similarly, watching somebody else play piano/tennis might teach you something, but it will not make you a good piano/tennis player. Students assume that, if they can follow someone else's code, they understand the topic, but if you don't practice writing it yourself, it isn't the same.
- Get all the help you can from your instructor or the tutoring service

Major Causes For Failure

- Lack of motivation, unwilling to commit the time and personal effort necessary to master course details.
- Students don't try to get help from instructors and tutorial services.
- Students get answers from others, confusing "getting the answer" with "understanding the material".
- Early difficulties lead to students getting discouraged, thinking that they are "naturally bad" at programming. A little secret: at first, almost all of us were bad at programming.

Program Information

Diversity Statement - At SJSU, it is important to create a safe learning environment where we can explore, learn, and grow together. We strive to build a diverse, equitable, inclusive culture that values, encourages, and supports students from all backgrounds and experiences.

Course Materials

Students are expected to come to lab with a laptop computer with BlueJ installed. We will allocate time to install BlueJ during the first/second labs. BlueJ can be downloaded from https://www.bluej.org/. (https://www.bluej.org/)

Course Requirements and Assignments

There will be no homework or exams in this lab.

Students are required to participate actively throughout the semester, and participation will be assessed at every class meeting. Insufficient participation will result in either being dropped or receiving a grade of NC (no credit). This is a one-unit Lab that meets 3 hours per week.

✓ Grading Information

This workshop is graded Credit/No Credit. The grading policy for the workshop will be based on classwork given during each workshop meeting. To receive a “CR” grade for this class, you must earn at least 70% of the participation points.

The classwork is based on topics that you are currently studying, have studied, or will study shortly. In order to receive credit for the classwork you must be actively participating in the coursework and either complete the assignment or spend most of the lab trying. You may use your books and notes and work with others in a group to complete assignments.

University Policies

Per University Policy S16-9 (PDF) (http://www.sjsu.edu/senate/docs/S16-9.pdf), relevant university policy concerning all courses, such as student responsibilities, academic integrity, accommodations, dropping and adding, consent for recording of class, etc. and available student services (e.g. learning assistance, counseling, and other resources) are listed on the Syllabus Information (https://www.sjsu.edu/curriculum/courses/syllabus-info.php) web page. Make sure to visit this page to review and be aware of these university policies and resources.

Course Schedule

Date	Topic	Details
8/21/26	Class Introduction	Review the Syllabus, play RoboRally
8/28/26	Everyday Algorithms + Java Syntax	Declaring variables, using comments, printing results
9/04/26	Scope + Object Oriented Programming + Casting	writing classes, this.variable vs variable, return values / method calling
9/11/26	Conditionals	booleans, boolean logic, if/else if/else
9/18/26	APIs	How to read them, how to make BlueJ generate them
9/25/26	Loops	for, while, do while, practice looping over strings
10/02/26, 10/09/26	Loops with Arrays	how to declare arrays, practice looping over arrays, ArrayLists

10/16/26, 10/23/26	2D Loops and Arrays	Monte carlo estimation, Connect 4 or larger TicTacToe, Insertion Sort
10/30/26	Static Methods/Variables, Inheritance	
11/6/26 - 12/4/26	Exercises	FileIO exercises, Inheritance exercises, etc