**CS& 141: Computer Science I (Java)**

Table of Contents

[Course Details 1](#_Toc186900923)

[**Prerequisites** 1](#_Toc186900924)

[**Course Outcomes** 2](#_Toc186900925)

[**Topic Coverage** 2](#_Toc186900926)

[**Course Calendar** 3](#_Toc186900927)

[Required Materials 6](#_Toc186900928)

[**Textbook** 6](#_Toc186900929)

[**Software** 6](#_Toc186900930)

[**Computer & Technical Requirements** 6](#_Toc186900931)

[Grading and Assessment 7](#_Toc186900932)

[**Purpose of Activities / Course Alignment** 7](#_Toc186900933)

[**Mandatory Decimal Grading** 8](#_Toc186900934)

[**Late Policy** 8](#_Toc186900935)

[**Academic Integrity** 9](#_Toc186900936)

[Classroom Contract 10](#_Toc186900937)

[**Diversity, Equity, Inclusion, and Accessibility (DEIA) in our Classroom** 10](#_Toc186900938)

[**How to be Successful** 11](#_Toc186900939)

[**Getting Unstuck** 12](#_Toc186900940)

[Letter of Recommendation Requests 13](#_Toc186900941)

[Student Services 14](#_Toc186900942)

[**Campus Closures / Cancelled Class** 14](#_Toc186900943)

[**Access and Accommodations** 14](#_Toc186900944)

[**Tutoring Services** 14](#_Toc186900945)

[**Counseling Center** 15](#_Toc186900946)

[**Additional Campus Resources** 15](#_Toc186900947)

[Privacy Policies 16](#_Toc186900948)

[Accessibility Policies 16](#_Toc186900949)

**Winter 2025 Quarter – Hybrid course**

# **Course Details**

|  |  |
| --- | --- |
| **Instructor** | Dr. Crystal Hess - [chess@shoreline.edu](mailto:chess@shoreline.edu)   * I prefer messages to be sent via Canvas. * Typical response time is <24 hours,  this excludes weekends and holidays. |
| **Office Hours** | Mondays 1 PM - 2:30 PM on [zoom](https://zoom.us/my/chess)  Tuesdays and Thursdays 11:30 AM - 12:30 PM in 1308 lab  Or by appointment |
| **Class Hours** | Online Material +  T/Th 12:30pm - 2:30pm in computer lab 1308 |

**CS& 141 - Computer Science I (5 credits)**

This course will explore common computational problem-solving techniques useful to computer scientists, but also to anyone who has large data sets, repetitive processes or other needs for computation. Topics include: fundamental programming-in-the-small abilities and concepts, including procedural programming (methods, parameters, return, values), basic control structures (sequence, if/else, for loop, while loop), file processing, arrays, and an introduction to defining objects.

Due to the amount of material and quick pace for consumption, prior programming experience is expected.

## **Prerequisites**

ENGL& 101 (may be taken concurrently) and MATH& 141 or MATH 111 with grades of 2.0 or better. CS 121 with a grade of 2.0 or better or other programming experience.

## **Course Outcomes**

1. Develop solutions to small-scale problems, such as determining the best candidate for an admissions process, analyzing text for keywords, or playing a game of word guess.
2. Design and implement solutions to partially ambiguous problems, such as analyzing a set of data for patterns and information or building upon an existing code base.
3. Identify and use programming language constructs and coding patterns, such as iteration, methods, classes, arrays, fencepost problems, and token-based processing.
4. Break down problems into logical pieces using repetition, methods, and classes.
5. Test and debug solutions until they meet specifications.
6. Examine and compare reasoning, design, and effectiveness of alternative solutions.

## **Topic Coverage**

*Estimated* Topic Coverage

|  |  |
| --- | --- |
| Week 1 | Chapter 1: Introduction to Java  Chapter 2: Primitive Data and Definite Loops |
| Week 2 | Chapter 2: Primitive Data and Definite Loops  Chapter 3: Parameters and Objects |
| Week 3 | Chapter 4: Conditional Execution |
| Week 4 | Chapter 4: Conditional Execution  Chapter 5: Program Logic and Indefinite Loops |
| Week 5 | Chapter 5: Program Logic and Indefinite Loops |
| Week 6 | Chapters 1-5, Midterm Exam  Chapter 6: File Processing |
| Week 7 | Chapter 6: File Processing |
| Week 8 | Chapter 7: Arrays |
| Week 9 | Chapter 8: Classes  Chapter 10: ArrayLists |
| Week 10 | Chapter 9: Inheritance and Interfaces  Chapters 1-10, Pair Work |
| Finals Week | Final Exam |

**The course is a HYBRID course** meaning that we will have in-person as well as online course material. Students are expected to attend in-person activities as this is NOT a fully online course.

## **Course Calendar**

\*subject to modification if needed

|  |  |  |
| --- | --- | --- |
| WEEK 1 | Lecture | Homework |
| Tuesday | Notes: 1a Introduction to Java Programming  Lecture: Ch1a | 1a Practice - java programs  1a Reflection - Egg2  Study: Chapter 2  Preview Quiz - Chapter 2 |
| Thursday | Notes: 2a Primitive Data and Definite Loops  Lecture: Ch1b, Ch2a | 2a Practice - data and for loops  2a Reflection - waveNumbers40  Program #0 |
| Sunday |  | Start: Program #1 |
| WEEK 2 | Lecture | Homework |
| Tuesday | Notes: 2b Definite Loops, Constants  Lecture: Ch2b | 2b Practice - complex loops and constants  2b Reflection - nestedNumbers2  Study: Chapter 3  Preview Quiz - Chapter 3 |
| Thursday | Notes: 3a Parameters, Return Types, and Objects  Lecture: Ch2c, Ch3a | 3a Practice - parameters  3a Reflection - printPowersOf2 |
| Sunday |  | Study: Chapter 4  Preview Quiz - Chapter 4 |
| WEEK 3 | Lecture | Homework |
| Tuesday | Notes: 4a User Input and Conditionals  Lecture: Ch4a | 4a Practice - if statements  4a Reflection - fractionSum |
| Thursday | Notes: 4b Common Algorithms and printf  Lecture: Ch4b | 4b Practice - more practice  4b Reflection - wordCount |
| Sunday |  | Get ahead? Study: Midterm  Expressions, Parameters, If/Else |
| WEEK 4 | Lecture | Homework |
| Tuesday | Pair Programming: Candy Machine (started and completed in class) | Study: Chapter 5  Preview Quiz - Chapter 5 |
| Thursday | Notes: 5a while loops and fence post problems  Lecture: Ch5a | 5a Practice - while loops  5a Reflection - digitSum |
| Sunday |  | Start: Program #2  Get ahead? Study: Midterm |
| WEEK 5 | Lecture | Homework |
| Tuesday | Notes: 5b Random numbers and Assertions  Lecture: Ch5b | 5b Practice - putting it all together  5b Reflection - threeHeads |
| Thursday | Pair Programming: Guessing Game (started and completed in class) |  |
| Sunday |  | Study: Midterm |
| WEEK 6 | Lecture | Homework |
| Tuesday |  | Study: Chapter 6  Preview Quiz - Chapter 6 |
| Thursday | Notes: 6a Token-based Processing and File Output  Lecture: Ch6a | 6a Practice - token-based processing  6a Reflection - negativeSum |
| Sunday |  | Get ahead?  Study: Chapter 7  Preview Quiz - Chapter 7 |
| WEEK 7 | Lecture | Homework |
| Tuesday | Notes: 6b Line-based Processing  Lecture: Ch6b | 6b Practice - line-based processing  6b Reflection - printDuplicates  Study: Chapter 7  Preview Quiz - Chapter 7 |
| Thursday | Notes: 7a Arrays  Lecture: Ch7a | 7a Practice - Arrays  7a Reflection - range |
| Sunday |  | Start: Program #3  Get ahead?  Study: Chapter 8  Preview Quiz - Chapter 8 |
| WEEK 8 | Lecture | Homework |
| Tuesday | Notes: 7b More on Arrays  Lecture: Ch7b | 7b Practice - More Arrays  7b Reflection - collapse  Study: Chapter 8  Preview Quiz - Chapter 8 |
| Thursday | Notes: 8a OOP  Lecture: Ch8a | 8a Practice - OOP  8a Reflection - quadrantPoint |
| Sunday |  | Start: Program #4  Get ahead?  Study: Chapter 10  Preview Quiz - Chapter 10 |
| WEEK 9 | Lecture | Homework |
| Tuesday | Pair Programming: OOP (started and completed in class) | 8b Practice - More OOP  8b Reflection - addTimeSpan  Study: Chapter 10  Preview Quiz - Chapter 10 |
| Thursday | Notes: 10a ArrayList  Lecture: Ch10a | 10a Practice - ArrayList  10a Reflection - stutter |
| Sunday |  | Get ahead?  Study: Chapter 9  Preview Quiz - Chapter 9  Study: Final |
| WEEK 10 | Lecture | Homework |
| Tuesday | Pair Programming: GroceryList (started and completed in class) | Chapter 9 Material  Preview Quiz - Chapter 9  🎬Video Lecture: 9a Inheritance  Graphics setup for next class: ACM Graphics |
| Thursday | Pair Programming: Wack-a-mole (started and completed in class) | 9a Practice - Inheritance  9a PracticeIt Reflection - MinMaxAccount |
| Sunday |  | Study: Final |

# **Required Materials**

## **Textbook**

* OPTIONAL: Building Java Programs: A Back to Basics Approach, 5th Edition (older versions work) by Reges and Stepp (same book used for CS143)

## **Software**

* *Java Development Kit (JDK) (freely available online)*
* *JGrasp IDE (freely available online)*

## **Computer & Technical Requirements**

* It is expected that when students enroll for this course they will be able to use a computer, be able to upload and download files, and successfully navigate browsers and websites.
* Students will receive homework throughout the course that must be completed on a computer with Internet access. The school library computers can be used for these purposes.
* **Need help?** Canvas questions can be directed to Shoreline e-Learning at [elearning@shoreline.edu](mailto:elearning@shoreline.edu), (206) 546-6966, visit Building 1200, or [24hour Canvas Support via chat or telephone](http://www.shoreline.edu/virtual-campus/elearning/canvas.aspx). Canvas and basic computing support is also available on campus at the [Shoreline Business Technology & eLearning Center](http://www.shoreline.edu/apply-and-aid/learning-support-centers/business-computer-software.aspx) in Building 1300, Room 1304.
* More [information on general technical requirements](http://www.shoreline.edu/virtual-campus/elearning/tech.aspx) can be found here.

# **Grading and Assessment**

In this course, you will be assessed through **Learning Practice** (daily practice assignments that are graded mostly for completion), **Java Programs** (longer pair assignments and individual homework), **Exams** (assessing your ability to apply information within time bounds).

I strive to grade daily and weekly assignments within 5 days of the due date. The larger assignments take longer to grade and are generally graded within 7-10 days.

|  |  |
| --- | --- |
| **Activity** | **Possible Points** |
| Learning Practice | ~40 pts |
| Pair Programming | ~10 pts |
| Individual Programs | ~25 pts |
| Exams | ~40 pts |
| **TOTAL** | **~105 pts** |

## **Purpose of Activities / Course Alignment**

* **Learning Practice** (including Preview Quizzes, Practice Problems, Practice Reflections) gives you the opportunity to practice reading, writing, and reflecting on small segments of code. These assignments are mostly graded for completion. (Course Objectives 3, 4, and 6)
* **Pair Programming** creates a space for you to use vocabulary, negotiate use of different programming constructs, and test/debug with another person. These assignments must be completed with a partner. (Course Objectives 1, 2, 3, 4, and 5)
* **Long Programs** builds in difficulty throughout the course, providing an opportunity to build software solutions to tackle varying problems. (Course Objectives 1, 2, 3, 4, and 5)
* **Exams** provide an assessment of your understanding of programming vocabulary and ability to read and write code quickly. (Course Objectives 3 and 4)

## **Mandatory Decimal Grading**

**Your grade will be calculated as the cumulative points that you earn over the quarter.** Expect that at the start of the quarter your grade will appear low as you build up points. This also means that your grade (total points and final GPA) will *never* go down.

**Final GPA decimal grading** will be used in accordance with the following table. There is not an option for pass/fail grading in this course as it is transfer-equivalent.



## **Late Policy**

* Students are expected to turn in all assignments by the due date and time.
* Assignments not submitted on-time may receive zero points.
* Alternate arrangements *may* be approved for flexible due dates in special circumstances when the instructor is contacted **prior** to the due date. *No day-of extensions will be given. No assignments will be accepted beyond one week past an assignment’s due date.*

More information available here: [Shoreline's full Grades Policy (#6260)](https://www.shoreline.edu/about-shoreline/policies-procedures/documents/6260GradesPolicy.pdf)

## **Academic Integrity**

Academic integrity is a commitment, even in the face of adversity, to actively engage in the learning process by using appropriate resources, asking for help, and doing your best to learn and grow your skillset toward the course outcomes. This means that you should showcase your own learning throughout the course. Any student found guilty of cheating and/or plagiarism will receive a zero for the assignment. If it happens a second time, you may fail the course.

* You **should** honor whether assignments are to be completed individually or in collaboration with others.
* For individual assignments, you **should not** share solutions; you also **should not** look at someone else’s solution before submitting your own.
* For individual assignments, you **should not** have another person "walk you through" the assignment, describe in extreme detail how to solve it, nor sit with you as you write it.
* You **should not** post your homework solution online to ask others for help.
* You **should not** use generative AI to complete long programs.
* You **should not** present a solution you did not write as your own. For online and/or AI related resources, you may utilize small segments of code **if** you attribute the code found/used and the use of those resources is allowed for that assignment (e.g., external resources are NOT allowed for exams)

Please note that **both parties involved in a cheating case are equally guilty**. That is, helping someone else cheat is just as bad cheating yourself. There are many resources for students who are struggling, and the right thing to do in this situation is to point to these resources or ask the instructor for help.

# **Classroom Contract**

You are expected to uphold the classroom contract as follows:

* **Be Prepared**: You are expected to complete pre-work and homework before class starts so that you are ready to be engaged in conversations and activities. We learn to code by coding—so put your finger muscles to work!
* **Be Present**: You are expected to be in class, on time, and actively present in the learning process.
* **Participate**: You are expected to be actively engaged in your learning in class and online. This means both asking questions and helping others. It also means utilizing the tools that enhance your ability to learn while avoiding the temptations and tools that hinder learning. When in doubt, don’t hesitate to reach out to the instructor to strategize how you will learn this material best!
* **Create space for learning:** You are expected to create and cultivate a space where learning is conducive for all learners. This means that you participate in the class without dominating the learning space.

## **Diversity, Equity, Inclusion, and Accessibility (DEIA) in our Classroom**

It is my hope that our classroom is filled with a **diversity** of experience, backgrounds, and perspectives which we honor and celebrate. Having different lived experiences and different ways of thinking can make our interactions richer, our thinking more comprehensive, and—as a result—our products better.

Because every person has different needs and different ways of engaging, each student may need different resources and support on their learning journey. Acknowledging and honoring these differences means that we strive to provide an **equitable** learning experience (not necessarily an equal, standardized, experience) where each individual student gets the resources and support which they need.

Together, we must intentionally **include** our peers in our learning process by seeking their opinions, valuing divergent thinking, and finding ways to help each other feel a sense of belonging and success along the way. Sometimes this means stepping up to lead and sometimes this means stepping back to listen.

To make learning **accessible**, I strive to provide multiple ways of student engagement, a variety of representation in the material, and multiple ways of expressing learning. I also strive to honor accommodation needs (documented or undocumented) due to visible/invisible differences as well as temporary/ relapsing/remitting conditions, or long-term life situations.

We all play a role in the ongoing effort to create a diverse, equitable, inclusive, and accessible learning environment. I hope you will join me in making our learning space a place where all learners find joy and success in learning Computer Science.

Campus Life Resources

|  |  |
| --- | --- |
| * **Gender**   + [How do I update the college with my preferred name?](http://www.shoreline.edu/apply-and-aid/name-change.aspx)   + [Where are the all gender restrooms on campus?](https://www.shoreline.edu/map/all-gender-bathrooms.aspx) * **Financial**   + [Where can I find information on scholarships or financial aid?](https://www.shoreline.edu/apply-and-aid/financial-aid/sources-on-campus.aspx)   + [Where can I get an emergency short-term personal loan?](https://www.shoreline.edu/multicultural/short-term-emergency-loan.aspx)   + [Where is the Food Pantry / Benefits Hub?](https://www.shoreline.edu/apply-and-aid/funding-and-aid/financial-aid/benefits-hub/default.aspx) | * **Mental Health**   + [Where can I talk to someone about mental health or counseling?](https://www.shoreline.edu/counseling-center/) * **Tell me about the...**   + [Multicultural Center](https://www.shoreline.edu/multicultural/)   + [Gender Equity Center](https://www.shoreline.edu/gender-equity-center/default.aspx)   + [Veteran and Military Services](https://www.shoreline.edu/oss/veterans/)   + [Parent/Child Center](https://www.shoreline.edu/parentchildcenter/)   + [Living on campus in Student Housing](https://www.shoreline.edu/housing/)   + [Career Center](https://www.shoreline.edu/job-career-services/career-center/default.aspx) |

## **How to be Successful**

* **Come to class**: Participating in class lectures and practice problems will give you structured time to absorb new concepts, practice them, and ask questions to the Instructor and your peers.
* **Ask for help early**: In programming, everything stacks, meaning what you learn today, you WILL use tomorrow. Therefore, if you start to fall behind, it is imperative that you ask for help as soon as possible.
* **Practice**: It is not enough in programming to just read about or listen to lectures about concepts. It is important to solve problems both in-class and on your own. You should work through practice problems (like on PracticeIt) until you are able to solve them without the aid of another person.
* **Utilize the textbook, slides, and videos provided**: While lecture will cover most of the concepts, the textbook, slides, and videos provided will provide more detail and reference outside of class.

## **Getting Unstuck**

Getting stuck is part of the programming process. However, one of the most useful skills you may learn from this course is "how to get unstuck."

Here are some suggestions:

* **Make systematic changes.** Look carefully at the work you have just done. Try changing one thing and see what effect it has.
* **Talk to a rubber duck.** No really, this is actually [a thing](https://en.wikipedia.org/wiki/Rubber_duck_debugging). Sometimes if you take a moment to tell someone (even someone non-techie) what you're trying to do, you'll end up having an Ah-Ha moment of clarity!
* **Ask a classmate.** Try to share ideas about how to figure out the problem rather than telling your classmate the answer. You'll learn as much by helping others find their mistakes as you will by finding your own.
* **Take a break.** Sometimes you just need to take a break! Looking at code with a fresh set of eyes solves problems quicker than beating your head against the computer :)
* **Ask the Instructor.** I'm here to help you. It's literally my job.

# **Letter of Recommendation Requests**

Letters of recommendation are often needed for applications to transfer universities, jobs, and internships. Instructors take pride in the letters that they write for students and can only craft strong letters for students whom they know well. Part of your job as a college student is to become the kind of student that a professor can speak highly of—hardworking, capable, and intellectually inquisitive.

**Before requesting a letter of recommendation, ask yourself:**

* Have I discussed my academic or career goals with this instructor?
* Have I demonstrated an excellent work ethic or produced quality work in this course?
* Have I demonstrated responsibility for my learning and active participation in class (good attendance, thoughtful communication with the instructor and my peers)?
* Have I shown or communicated a passion for the subject or concepts that were taught?
* Will this instructor be able to incorporate personal and specific details about my academic growth or trajectory?

It is not necessary that the person writing your recommendation be able to speak to every bullet point above, but they should be able to address at least one.

Sometimes you are asked if you would like to **waive your rights** to read the letter. One advantage to waiving your rights is that the people reading the letter will know it was written candidly, which could make the letter more influential.  The disadvantage is that you won’t get to see what was written. Thus, it is important to ask your potential recommender if they can provide a *positive* letter.  If they can’t say “yes” or suggest you ask another person, then ask someone else.

**Requests may be declined for a variety of reasons**, including, but not limited to: insufficient time to write the letter, not knowing the student well enough to provide specific character observations, or too much time has elapsed since working with the student.

**Requests for recommendations should come well in advance of the deadline for submission** (i.e., at least 14 days in advance of the deadline)

# **Student Services**

## **Campus Closures / Cancelled Class**

There are two types of suspended operations possible: campus is closed or classes are cancelled.  In the event of campus closure or cancelled class, students will be notified via Canvas of their expectations in relation to school work.

You can read Shoreline’s [Suspended Operations Procedures](https://www.shoreline.edu/about-shoreline/policies-procedures/documents/6030SuspendedOperationsProcedure.pdf)

## **Access and Accommodations**

Shoreline Community College is committed to providing educational programs without regard to disabling conditions as defined by Section 504 of the Rehabilitation Act of 1973. Reasonable accommodations will be made and no otherwise qualified individual with disabling conditions shall, on the basis of disability, be excluded from participation in, be denied the benefits of, or otherwise be subjected to discrimination under any program, activity or service administered by the college.

For more information: [Shoreline Student Accessibility Services](http://www.shoreline.edu/oss/students-with-disabilities/default.aspx)

## **Tutoring Services**

Tutoring Services provides students with free one-on-one tutoring support for any Shoreline Community College classes. Students can receive 2 hours of free tutoring a week per class they are registered for in a quarter.

For more information and to apply for tutoring assistance or to apply to become a tutor, please visit the office in 4228 (Library), email [pttutors@shoreline.edu](mailto:pttutors@shoreline.edu), call 206-546-4776, or check out the webpage: [www.shoreline.edu/tutoring](http://www.shoreline.edu/tutoring)

## **Counseling Center**

The Counseling Center provides free, confidential and professional counseling services, resources, and referral to support the academic and personal success, health, and well-being of our students and campus community.

Students often visit the Counseling Center to discuss a wide variety of topics: depression, anxiety, relationship concerns, and stress management; indecision about major or career path; and academic concerns such as failing grades, struggling with a subject, or managing a learning disability. The Center also supports students who may be feeling suicidal or in crisis.

* FOSS – 5245, 206-546-4594, [www.shoreline.edu/counseling-center](http://www.shoreline.edu/counseling-center)

Need support when they are not available? For 24/7 emergency counseling, referral, or assistance please contact:

* King County: 24-Hour Crisis Line | 866-427-4747
* Snohomish County: 24-Hour Crisis Line | 800-584-3578
* Live Chat: [crisischat.org](http://crisischat.org/)
* Crisis Text Line: Text 741741
* 911 (for immediate health-related emergency)

## **Additional Campus Resources**

Check out the [Current Students page](https://www.shoreline.edu/currentstudents/) for more information about Academic Support, Student Services, Campus Life, and much more.

# **Privacy Policies**

To learn about the practices regarding personal information that may be collected from users in this course, check the privacy policies below:

* [PracticeIt](https://practiceit.cs.washington.edu/privacy-policy) Privacy Policy
* [Canvas](https://www.canvaslms.com/policies/privacy)
* [Panopto](https://www.panopto.com/privacy/)
* [Shoreline Privacy Policies](https://www.shoreline.edu/currentstudents/privacyandnondiscrimination.aspx)
* [Student Policies](https://www.shoreline.edu/currentstudents/student-policies.aspx)
* [Microsoft 365](http://www.microsoft.com/online/legal/v2/?docid=22&langid=en-us)
* [Google Privacy Policy](https://www.google.com/intl/en/policies/privacy/)

# **Accessibility Policies**

Below are links to accessibility policies for sites used within this course:

* [Canvas](https://www.canvaslms.com/accessibility)
* [Panopto Accessibility Features](https://support.panopto.com/articles/Documentation/accessibility-features)
* [Shoreline Community College](http://intranet.shoreline.edu/policies/documents/5000/5114.pdf) [(Audio Version)](http://intranet.shoreline.edu/policies/documents/5000/5114.pdf)
* [Microsoft Office 365](https://support.office.com/en-US/article/Accessibility-in-Office-365-ACA7ACCF-58A0-4467-BE5C-24A7E7933A9D?ui=en-US&rs=en-US&ad=US)
* [Google (covers YouTube)](https://www.google.com/accessibility/products-features.html)