

Scott Bracken  
Room A127

Phone: 360-563-4106  
Email: [scott.bracken@sno.wednet.edu](mailto:scott.bracken@sno.wednet.edu)

Table of Contents

Curricular Requirements..... 2

Note to Parents or Guardians .....3

Course Overview ..... 3

College Credit (College in the High School - Everett Community College)..... 3

Technology Requirement..... 3

High Expectations ..... 4

Textbook ..... 4

Additional Resources ..... 4

Grading..... 5

Cheating Policy..... 5

Curriculum Overview and Goals .....6

Computer Science A Course-At-A-Glance .....7

---

## Curricular Requirements

|             |                                                                                                                  |
|-------------|------------------------------------------------------------------------------------------------------------------|
| <b>CR1</b>  | The course teaches students to design and implement computer-based solutions to problems                         |
| <b>CR2A</b> | The course teaches students to use and implement commonly used algorithms                                        |
| <b>CR2B</b> | The course teaches students to use commonly used data structures.                                                |
| <b>CR3</b>  | The course teaches students to select appropriate algorithms and data structures to solve problems.              |
| <b>CR4</b>  | The course teaches students to code fluently in an object-oriented paradigm using the programming language Java. |
| <b>CR5</b>  | The course teaches students to use elements of the standard Java library from the AP Java subset                 |
| <b>CR6</b>  | The course includes a structured lab component comprised of a minimum of 20 hours of hands-on lab experiences    |
| <b>CR7</b>  | The course teaches students to recognize the ethical and social implications of computer use.                    |

## Parents or Guardians

Please read this carefully. It is important that you understand what your student may be up against this year in this class. It is going to present challenges. There will likely be some stressful moments as they learn new concepts and strive to keep up with the workload. Please encourage your student to work hard each day, turn assignments in on time, and to get help when they feel lost. Just know, that I am always glad to help before and after school.

## Course Overview

This is a college-level computer science course designed to help you pass the AP Computer Science A exam. The major topics of this course are computer systems, primitive data types, conditional and flow control structures, arrays and array lists, introductory object-oriented programming concepts and structures, and recursion and sorting concepts. We will investigate and analyze course topics using in-class exercises, practice problems, multi-class labs and programming projects. We will provide a balanced focus on both the conceptual basis for the constructs and the implementation of those structures. We will also be exploring ways in which computer science and the problem-solving skills it requires are used in the real-world.

## Approach to earn College Credit – Advanced Placement

The alternative approach to earn college credit is to take the AP CS A exam in the first week of May. Materials will be provided to you, in class, before the enrollment period begins. The cost for taking the AP exam is \$98. Financial support is available to eligible students.

Don't worry. Students have done very well on the AP CS-A exam. Work hard and prepare and you too will do well.

## Technology Requirement

Students will be doing most of their work in CSAwesome, CodeAcademy and Practice-It!. All are accessible via the Internet. If you do not have working Internet connection at home, please let me know. Our district may be able to help by providing a hotspot for you. Students will also be using an integrated development environment (IDE) targeting a current version of the Java Runtime Engine.

## High Expectations

AP-Computer Science(CS) A will be a challenging course for most students. Unless you have already written code in Java before, you will be learning a new language, using your mathematics skills, and learning how to think in more logical, sequential ways. For most students, this class will require more effort than their other classes. Here's the good news! The rewards for knowing CS could be tremendous. Doing well in this class may launch you into an exciting career as a software engineer. There are millions of high-paying jobs in CS and if you like this class, you will probably love a career in CS.

## Advice to Students and Parents

This course is a COLLEGE-LEVEL course. There will be a lot of homework and a lot of coding practice. You will spend quite a bit of time debugging your code and beating your head against the wall because you can't figure something out. The key is to have a lot of GRIT. Writing code can be a very frustrating task. Yet, it will be a rewarding task. Some of my best memories come from developing software back in the 1990's. I hope that you will have great memories of some of your accomplishments that you have in this class this year. You can have that if you work hard and do not procrastinate. You will need every moment of the time provided to complete the work. Students who put assignments off until the last minute often get desperate and start copying other student's code, only to get caught and ultimately fail.

## Textbook

We will be using CSAwesome, CodeAcademy, CodingBat.com and CodeStepbyStep. They are online either directly or through HelloID. Much of our learning will not come from a textbook. However, we do provide a textbook (Reges and Stepp. Building Java Programs. 4th ed. Pearson, 2016) that you can use as a resource. This is the same textbook used at the University of Washington for their Introduction to Java programming course. The textbook can be a great tool for looking for further clarification of the concepts we will be learning in Code.org.

## Class Rules

1. Respect for students and adults. You may not agree with someone but that is not a reason to be disrespectful.
2. Assignments are due on the posted due date. Late work is accepted if the teacher is notified and an agreement is made on when it will be turned in. **Assignments must be made up prior to the unit test.** Missing assignments will earn a "0" until turned in and will be final after the test for the unit.
3. If you are absent on the day a quiz or test is given, it must be made up either before or after school, or Panther Period, on the day your return.
4. Follow all SHS rules in student handbook. The handbook can be located online at the SHS website.
5. **Cell phones and other personal connected electronic devices will be turned off and put away while in class and during instructional time.**

## Grading

Your grade will be determined as follows:

65% - Unit Tests, Quizzes and Projects

35% - Daily Participation and Homework

Grading is based on the following scale:

|    |         |    |        |    |        |      |        |
|----|---------|----|--------|----|--------|------|--------|
| A  | 93-100% | B  | 83-87% | C  | 73-77% | D    | 60-67% |
| A- | 90-93%  | B- | 80-83% | C- | 70-73% | Fail | <60%   |
| B+ | 87-90%  | C+ | 77-80% | D+ | 67-70% |      |        |

## Late Work

Learning to complete projects on time is an important skill that you will need when you become a software engineer. People who code are notorious for putting in long hours to meet a deadline. I think you should have the same mindset. Here is what I plan to do to help you with learning to meet deadlines.

You can earn up to:

- 100% for on-time work.
- 70% for late work within 1 week.
- 50% for late work within 2 weeks.
- After 2 weeks, late work earns no credit.

## Cheating Policy

Computer science is by its nature a collaborative endeavor. It is essential to learn the skill of effectively communicating with your peers so that you can help each other solve difficult problems. However, you also need to be an independent learner. As such, cheating is strictly forbidden. For the purposes of this class the following actions constitute cheating:

- Submitting another student's code (even a small section) as your own.
- Submitting code generated solely from internet sources.
- Providing another student code from your project or lab assignment.
- Sitting at another student's computer and entering code for them.

---

Here is what you can do:

- You may help a friend by walking through their code with them, but they have no right to steal your code. Keep this in mind.
  - You can talk about strategies for how to write their code.
  - You can help your friend develop a flowchart or flow diagram to help them think through the process for solving the problem.
-

## Curriculum Overview and Goals

Software development and design is a core component of writing apps and software that users interact with every day. This curriculum seeks to develop software engineering skills and expose students interested in pursuing a computer science education to industry best practices while learning the Java programming language.

| Unit                                                                                           | Description                                                                                                                                                                                                             |
|------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Unit 1</b><br><b>Object-Oriented Programming</b><br><i>twenty 45-minute lessons</i>         | Students learn fundamental Java concepts, discover their identity as software engineers, and use debugging strategies and code reviews to improve their programming skills.                                             |
| <b>Unit 2</b><br><b>Class Structure and Design</b><br><i>fifteen 45-minute lessons</i>         | Students develop an understanding of data encapsulation and object-oriented design decisions.                                                                                                                           |
| <b>Unit 3</b><br><b>Arrays and Algorithms</b><br><i>fifteen 45-minute lessons</i>              | Students use one-dimensional (1D) arrays to store lists of primitive values and object references.                                                                                                                      |
| <b>Unit 4</b><br><b>Conditions and Logic</b><br><i>fifteen 45-minute lessons</i>               | Students improve decision-making in their programs using relational and logical operators to evaluate primitive values and object references and use the Mathclass to perform calculations and generate random numbers. |
| <b>Unit 5</b><br><b>Two-Dimensional Arrays</b><br><i>fifteen 45-minute lessons</i>             | Students use two-dimensional (2D) arrays to expand storage and organization capabilities in their programs.                                                                                                             |
| <b>Unit 6</b><br><b>ArrayLists and String Methods</b><br><i>fifteen 45-minute lessons</i>      | Students use ArrayLists to store program data and work with methods in the String class to manipulate and evaluate Stringobjects.                                                                                       |
| <b>Unit 7</b><br><b>Method Decomposition and Recursion</b><br><i>fifteen 45-minute lessons</i> | Students practice software design and development using the skills and programming constructs they have learned and explore recursion.                                                                                  |
| <b>Unit 8</b><br><b>Searching and Sorting</b><br><i>fifteen 45-minute lessons</i>              | Students implement searching and sorting algorithms to improve the efficiency and capabilities of their programs.                                                                                                       |
| <b>Unit 9</b><br><b>AP Exam Review and Practice</b><br><i>fifteen 45-minute lessons</i>        | Students prepare for the AP CSA Exam by practicing strategies for the multiple-choice and free response questions.                                                                                                      |

| Unit 1                      |                              | Unit 4 |                             | Unit 7 |                                  |                        |              |
|-----------------------------|------------------------------|--------|-----------------------------|--------|----------------------------------|------------------------|--------------|
| wk 1                        | Welcome to CSA               | wk 1   | The Theater                 | wk 1   | Project Planning                 |                        |              |
|                             | Java Lab                     |        | Object Aliases and Equality |        | Method Parameters                |                        |              |
|                             | The Neighborhood             |        | Nested if Statements        |        | Overloading Methods              |                        |              |
|                             | Navigating and Painting      |        | Logical Operators           |        | Private Methods                  |                        |              |
|                             | One-Way Selection Statements |        | De Morgan's Laws            |        | Overriding Methods               |                        |              |
|                             | PainterPlus                  |        | Static Methods              |        | Intellectual Property            |                        |              |
|                             | Writing Methods              |        | The Math Class              |        | Recursion                        |                        |              |
|                             | 2                            |        | Code Reviews                |        | 2                                | Recursion vs Iteration |              |
|                             | Loops                        |        | Random                      |        | Recursive Algorithms             |                        |              |
|                             | Two-Way Selection Statements |        | Multi-Selection Statements  |        | Maximizing System Reliability    |                        |              |
| 3                           | Debugging Strategies         | 3      | Abstract Data Art Project   | 3      | Creative Coding with The Theater |                        |              |
|                             | Decomposition and Design     |        | FRQ Practice                |        | FRQ Practice                     |                        |              |
|                             | PatternPainter               |        | Unit 4 Assessment           |        | Unit 7 Assessment                |                        |              |
|                             | Unit 5                       |        | Unit 8                      |        |                                  |                        |              |
|                             | BackgroundPainter            |        | Two-Dimensional (2D) Arrays |        | Project Planning                 |                        |              |
| 4                           | Open Source Code             | wk 1   | Row-Major Traversal         | wk 1   | Searching                        |                        |              |
|                             | Asphalt Art Project          |        | Column-Major Traversal      |        | Binary Search                    |                        |              |
|                             | FRQ Practice                 |        | Reverse a 2D Array          |        | Recursive Binary Search          |                        |              |
|                             | Unit 1 Assessment            |        | Enhanced For Loops          |        | Selection Sort                   |                        |              |
|                             | Unit 2                       |        | Images in The Theater       |        | Insertion Sort                   |                        |              |
| wk 1                        | Variables                    | 2      | Impacts of Programs         | 2      | Merge Sort                       |                        |              |
|                             | Operators and Expressions    |        | Modifying Images            |        | Multiple Lists                   |                        |              |
|                             | Attributes                   |        | Finding Duplicates          |        | Searching and Sorting            |                        |              |
|                             | Constructors                 |        | Shifting a 2D Array         |        | Privacy and Security             |                        |              |
|                             | User Input                   |        | Personal Narrative Project  |        | Creating Coding with the Console |                        |              |
|                             | Class Hierarchies            |        | 3                           |        | FRQ Practice                     | 3                      | FRQ Practice |
|                             | Accessor Methods             |        | Unit 5 Assessment           |        | Unit 8 Assessment                |                        |              |
|                             | 2                            |        | Mutator Methods             |        | Unit 6                           |                        | Unit 9       |
|                             | Printing Objects             |        | Projects and Documentation  |        | The AP CSA Exam                  |                        |              |
|                             | Scope and this               |        | Integer and Double Objects  |        | MCQ Pre-Assessment               |                        |              |
| 3                           | Store Management Project     | wk 1   | ArrayLists                  | wk 1   | MCQ Study Plan                   |                        |              |
|                             | FRQ Practice                 |        | Manipulating Elements       |        | MCQ Practice                     |                        |              |
|                             | Unit 2 Assessment            |        | Lists of Objects            |        | FRQ Pre-Assessment               |                        |              |
| Unit 3                      |                              | 2      | Removing Elements           | 2      | FRQ Study Plan                   |                        |              |
| One-Dimensional (1D) Arrays | String Methods               |        | FRQ Practice                |        |                                  |                        |              |
| Array Elements              | Comparing Strings            |        | Mock MCQ Exam               |        |                                  |                        |              |
| For Loops                   | String Algorithms            |        | Mock FRQ Exam               |        |                                  |                        |              |
| Array Algorithms            | ArrayList Algorithms         |        |                             |        |                                  |                        |              |
| wk 1                        | Enhanced For Loops           |        |                             |        |                                  |                        |              |

|   |                                |   |                                     |
|---|--------------------------------|---|-------------------------------------|
|   | Static Variables and Constants |   | Natural Language Processing Project |
|   | Polymorphism                   | 3 | FRQ Practice                        |
| 2 | More Array Algorithms          |   | Unit 6 Assessment                   |
|   | Reversing Arrays               |   |                                     |
|   | Finding Duplicates             |   |                                     |
|   | Data for Social Good Project   |   |                                     |
| 3 | FRQ Practice                   |   |                                     |
|   | Unit 3 Assessment              |   |                                     |

Would you please sign on the line below to indicate that you have read the syllabus?

---

(Signature)

Also, would you please tell me any concerns that you have as well as suggestions that you feel may be helpful to deal with them?

---

---

---

---

---

Thank you,

Scott Bracken  
Snohomish High School  
Computer Science Instructor