CodeHS

AP Computer Science A Course Syllabus

Introduction

AP Computer Science A introduces students to computer science through programming. Fundamental topics in this course include the design of solutions to problems, the use of data structures to organize large sets of data, the development and implementation of algorithms to process data and discover new information, the analysis of potential solutions, and the ethical and social implications of computing systems. The course emphasizes object-oriented programming and design using the Java programming language.

The CodeHS AP Computer Science A course is a year-long course designed to help students master the basics of Java and equip them to successfully pass the College Board AP Computer Science A Exam at the end of the school year. All learning materials and resources teachers and students need for a successful year-long AP Computer Science A course can be found on the CodeHS website.

Course Overview and Goals

Prerequisites

It is recommended that a student in the AP Computer Science A course has successfully completed a first-year high school algebra course with a strong foundation of basic linear functions, composition of functions, and problem-solving strategies that require multiple approaches and collaborative efforts. In addition, students should be able to use a Cartesian (x, y) coordinate system to represent points on a plane. It is important that students and their advisers understand that any significant computer science course builds upon a foundation of mathematical reasoning that should be acquired before attempting such a course.
This course is meant to be a first time introduction to computer science, and does not require students to come in with any computer programming experience. However, we recommend that students take our Introduction to Computer Science prior to our AP courses (more info at codehs.com/library). Students who have completed our Intro to CS course will be able to apply knowledge of concepts covered in the Intro course to the more advanced setting of the AP courses.

Learning Environment

The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, example programs to explore, and written programming exercises, adding up to over 100 hours of hands-on programming practice in total. Several units have free response questions that have students consider the applications of programming and incorporate examples from their own lives.

Programming Environment
Students write and run Java programs in the browser using the CodeHS editor.

College Board Personal Progress Checks
College Board has launched the AP Classroom, a new resource for teachers with summative and formative assessments. At the end of each unit, we recommend that teachers give students the corresponding Personal Progress Check to understand student strengths and weaknesses.

More information: Browse the content of this course at https://codehs.com/course/6165

Course Breakdown

Semester One

Unit 1: Primitive Types (2.5 weeks or 8 hours)

This unit introduces students to the Java programming language and the use of classes, providing students with a firm foundation of concepts that will be leveraged and built upon in all future units. Students will focus on writing the main method and will start to call preexisting methods to produce output. Students will start to learn about three built-in data types and learn how to create variables, store values, and interact with those variables using basic operations. The ability to write expressions is essential to representing the variability of the real world in a program and will be used in all future units. Primitive data is one of two categories of variables covered in this course. The other category, reference data, will be covered in Unit 2.

Topics Covered	● Welcome to CSA ● Why programming? Why Java? ● Variables and Data Types ● Expressions and Assignments Statements ● Compound Assignment Operators ● User Input ● Casting and Ranges of Variables

Unit 2: Using Objects (3 weeks or 13 hours)

In the first unit, students used primitive types to represent real-world data and determined how to use them in arithmetic expressions to solve problems. This unit introduces a new type of data: reference data. Reference data allows real-world objects to be represented in varying degrees specific to a programmer's purpose. This unit builds on students' ability to write expressions by introducing them to `Math` class methods to write expressions for generating random numbers and other more complex operations. In addition, strings and the existing methods within the

`String` class are an important topic within this unit. Knowing how to declare variables or call methods on objects is necessary throughout the course but will be very important in Units 5 and 9 when teaching students how to write their own classes and about inheritance relationships.

Topics Covered	• Objects: Instances of Classes • Creating and Storing Objects (Instantiation) • Calling a Void Method • Calling a Void Method with Parameters • Calling a Non-void Method • `String` Objects: Concatenation, Literals, and More • `String` Methods • Wrapper Classes: `Integer` and `Double` • Using the `Math` class

Unit 3: Boolean Expressions and `if` Statements (3 weeks or 13 hours)

Algorithms are composed of three building blocks: sequencing, selection, and iteration. This unit focuses on selection, which is represented in a program by using conditional statements. Conditional statements give the program the ability to decide and respond appropriately and are a critical aspect of any nontrivial computer program. In addition to learning the syntax and proper use of conditional statements, students will build on the introduction of Boolean variables by writing Boolean expressions with relational and logical operators. The third building block of all algorithms is iteration, which you will cover in Unit 4. Selection and iteration work together to solve problems.

Topics Covered	• Boolean Expressions • `if` Statements and Control Flow • `if-else` Statements • `else if` Statements • Compound Boolean Expressions • Equivalent Boolean Expressions • Comparing Objects

Unit 4: Iteration (4 weeks or 16 hours)

This unit focuses on iteration using while and for loops. As you saw in Unit 3, Boolean expressions are useful when a program needs to perform different operations under different conditions. Boolean expressions are also one of the main components in iteration. This unit introduces several standard algorithms that use iteration. Knowledge of standard algorithms makes solving similar problems easier, as algorithms can be modified or combined to suit new situations. Iteration is used when traversing data structures such as arrays, ArrayLists, and 2D arrays. In addition, it is a necessary component of several standard algorithms, including searching and sorting, which will be covered in later units.

Topics Covered	• `while` loops • `for` loops • Developing Algorithms Using Strings

	● Nested Iteration ● Informal Code Analysis
Associated Lab: Consumer Review Lab	

Unit 5: Writing Classes (3 weeks or 13 hours)
This unit will pull together information from all previous units to create new, user-defined reference data types in the form of classes. The ability to accurately model real-world entities in a computer program is a large part of what makes computer science so powerful. This unit focuses on identifying appropriate behaviors and attributes of real-world entities and organizing these into classes. Students will build on what they learn in this unit to represent relationships between classes through hierarchies, which appear in Unit 9. The creation of computer programs can have extensive impacts on societies, economies, and cultures. The legal and ethical concerns that come with programs and the responsibilities of programmers are also addressed in this unit.

Topics Covered	● Anatomy of a Class ● Constructors ● Documentation with Comments ● Accessor Methods ● Mutator Methods ● Writing Methods ● Static Variables and Methods ● Scope and Access ● `this` Keyword ● Ethical and Social Implications of Computing Systems

Semester Two

Unit 6: Array (2 weeks or 8 hours)
This unit focuses on data structures, which are used to represent collections of related data using a single variable rather than multiple variables. Using a data structure along with iterative statements with appropriate bounds will allow for similar treatment to be applied more easily to all values in the collection. Just as there are useful standard algorithms when dealing with primitive data, there are standard algorithms to use with data structures. In this unit, we apply standard algorithms to arrays; however, these same algorithms are used with ArrayLists and 2D arrays as well. Additional standard algorithms, such as standard searching and sorting algorithms, will be covered in the next unit.

Topics Covered	● Array Creation and Access ● Traversing Arrays ● Enhanced for Loop for Arrays ● Developing Algorithms Using Arrays
Associated Lab: Magpie Lab	

Unit 7: `ArrayList` (2.5 weeks or 10 hours)

As students learned in Unit 6, data structures are helpful when storing multiple related data values. Arrays have a static size, which causes limitations related to the number of elements stored, and it can be challenging to reorder elements stored in arrays. The ArrayList object has a dynamic size, and the class contains methods for insertion and deletion of elements, making reordering and shifting items easier. Deciding which data structure to select becomes increasingly important as the size of the data set grows, such as when using a large real-world data set. In this unit, students will also learn about privacy concerns related to storing large amounts of personal data and about what can happen if such information is compromised.

Topics Covered	● Introduction to `ArrayList` ● `ArrayList` Methods ● Traversing ArrayLists ● Developing Algorithms Using ArrayLists ● Searching ● Sorting ● Ethical Issues Around Data Collection
Associated Lab: Pokemon Simulation, Blackjack, Mad Libs, Data Lab ** This lab is not currently available on CodeHS. Visit the AP Classroom for options to complete this lab.	

Unit 8: 2D Array (3 weeks or 13 hours)

In Unit 6, students learned how 1D arrays store large amounts of related data. These same concepts will be implemented with two-dimensional (2D) arrays in this unit. A 2D array is most suitable to represent a table. Each table element is accessed using the variable name and row and column indices. Unlike 1D arrays, 2D arrays require nested iterative statements to traverse and access all elements. The easiest way to accomplish this is in row-major order, but it is important to cover additional traversal patterns, such as back and forth or column-major.

Topics Covered	● 2D Arrays ● Traversing 2D Arrays
Associated Labs: Picture Lab, Steganography Lab, Battleship	

Unit 9: Inheritance (3 weeks or 13 hours)

Creating objects, calling methods on the objects created, and being able to define a new data type by creating a class are essential understandings before moving into this unit. One of the strongest advantages of Java is the ability to categorize classes into hierarchies through *inheritance*. Certain existing classes can be extended to include new behaviors and attributes without altering existing code. These newly created classes are called *subclasses*. In this unit, students will learn how to recognize common attributes and behaviors that can be used in a *superclass* and will then create a hierarchy by writing subclasses to extend a superclass. Recognizing and utilizing existing hierarchies will help students create more readable and maintainable programs.

Topics Covered	<ul><li>Creating Subclasses and Superclasses</li><li>Writing Constructors for Subclasses</li><li>Overriding Methods</li><li>`super` Keyword</li><li>Creating References Using Inheritance Hierarchies</li><li>Polymorphism</li><li>`Object` Superclass</li></ul>
Associated Lab: Celebrity Lab	

Unit 10: Recursion (1 week or 4 hours)
Sometimes a problem can be solved by solving smaller or simpler versions of the same problem rather than attempting an iterative solution. This is called recursion, and it is a powerful math and computer science idea. In this unit, students will revisit how control is passed when methods are called, which is necessary knowledge when working with recursion. Tracing skills introduced in Unit 2 are helpful for determining the purpose or output of a recursive method. In this unit, students will learn how to write simple recursive methods and determine the purpose or output of a recursive method by tracing.

Topics Covered	<ul><li>Recursion</li><li>Recursion Searching and Sorting</li></ul>

Unit 11: Exam Review (2-4 weeks)
Prepare for the AP Exam by practicing the multiple-choice and free-response questions. Students should assess areas or topics that they need to practice.

Topics Covered	<ul><li>Students know what to expect on the AP Exam</li><li>Practice solving AP Exam type multiple choice questions</li><li>Practice solving AP Exam type free response questions</li></ul>

Unit 12: Final Project (2-4 weeks)
Students apply what they learned towards a final project on a topic of their choice.

Topics Covered	<ul><li>Allow students to think creatively about the applications of the concepts covered in the course</li><li>Scoping a project</li><li>Designing an application from scratch</li><li>Incremental development</li><li>Testing</li><li>Debugging</li></ul>