



Course Description: AP Computer Science A

Prerequisite: High school courses in English and algebra, and familiarity with functions and the concepts found in the uses of function notation

Description: The AP Computer Science A course introduces students to fundamental programming concepts using the Java language. It covers topics such as primitive and reference data types, Boolean logic, control structures like loops and conditionals, and object-oriented programming, including class design, inheritance, and recursion. Students will learn to write and trace code, use built-in Java libraries, and apply algorithms to solve real-world problems. The course emphasizes understanding how data structures like arrays and ArrayLists function and the legal and ethical considerations of programming.

Core skills:

- Designing, writing, and using Java methods
- Understanding and applying primitive and reference data types
- Implementing Boolean expressions, conditionals, and loops
- Designing and implementing classes and objects
- Utilizing arrays, ArrayLists, and 2D arrays
- Understanding inheritance and class hierarchies
- Writing recursive methods
- Applying standard algorithms for searching and sorting
- Recognizing ethical issues in programming
- Testing program code and correcting errors
- Documenting and explaining how program code works

Unit list:

Unit 1: Primitive Types

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. The use of preexisting methods for input is not prescribed in the course; however, input is a necessary part of any computer science course so teachers will need to determine how they will address this in their classrooms. 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.

High Bluff Academy is accredited by the Western Association of Schools and Colleges (WASC). The above course is approved by the University of California system (A-G) and the National Collegiate Athletic Association (NCAA).



Unit 2: Using Objects

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.

Unit 3: Boolean Expressions and if Statements

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.

Unit 4: Iteration

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.

Unit 5: Writing Classes

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.

High Bluff Academy is accredited by the Western Association of Schools and Colleges (WASC). The above course is approved by the University of California system (A-G) and the National Collegiate Athletic Association (NCAA).



Unit 6: Array

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.

Unit 7: ArrayList

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.

Unit 8: 2D Array

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.

Unit 9: Inheritance

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.

Unit 10: Recursion

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

High Bluff Academy is accredited by the Western Association of Schools and Colleges (WASC). The above course is approved by the University of California system (A-G) and the National Collegiate Athletic Association (NCAA).



High Bluff Academy

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.

Curriculum:

- CodeHS AP Computer Science A (Nitro) videos, MCQ, and auto-graded coding assignments
- College Board AP Daily videos, assignments, and tests

High Bluff Academy is accredited by the Western Association of Schools and Colleges (WASC). The above course is approved by the University of California system (A-G) and the National Collegiate Athletic Association (NCAA).