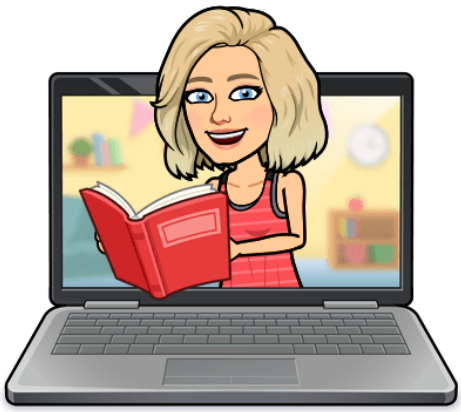
AP Computer Science A Syllabus 2021-22

Mrs. Goldschmidt
Harvard High School



Course Overview

Why is computer science significant? Computer science is an integral part of our lives, shaping virtually everything from the objects around us to the ways in which we communicate, travel, work, and play. And the computer revolution has just begun – CS is now a key enabler for discovery and innovation in most other fields of endeavor, making it an incredibly relevant course of study. Computer scientists invent the future by developing architectures and techniques for more advanced computing, and by developing the applications that operate within those frameworks. Computer science is a powerful force for making a positive difference in the world. Computing has made possible undertakings such as landing the Curiosity rover on Mars, managing patient care to avoid undesired drug interactions, revolutionizing K-12 teaching and learning through the use of mobile devices, and even the creation of a computer that can win at Jeopardy. It's a field of unbounded potential – get ready to change the world! (http://mibytes.eecs.umich.edu/what-is-computer-science/)

What is the course like? AP Computer Science is a college level course that teaches students how to program in the Java language. At the course's end, students are advised to take the AP Computer Science A exam. If they pass they may receive college credit for one semester of computer science.

Why Java? By any measure Java is one of the top computer languages in the world. A person knowledgeable in java can generally pick up other top rated languages such as C or C++ without too much difficulty. Computer science and the related area information technology (IT) form the single largest job market with one of the highest pay rates of the science, technology, engineering, and math (STEM) disciplines. While a student may never work as a computer professional, he or she is very likely to work with one. Previous exposure to Java would be helpful.

Who Should Take This Class? Traditionally, students with an interest in careers related to business, engineering, computer science, information technology, bioinformatics, genetics, physics, chemistry, or math should take AP computer science. However, in today's would, almost every occupation or endeavor uses some form of computer software including the fine arts. Some knowledge about how software is designed, created and maintained will provide useful background.

Course Outcomes

By the end of this course, students will be able to:

- Design and implement computer-based solutions to problems.
- Use and implement commonly used algorithms and data structures.
- Develop and select appropriate algorithms and data structures to solve new problems.
- Write solutions fluently an object-oriented paradigm
- Write, run, test and debug solutions in the Java programming language
- Read and understand programs consisting of several classes and interacting objects
- Read and understand a description of the design and development process
- Understand the ethical and social implications of computer use.

Materials

- Charged Chromebook - homework will be both written and typed; you will be using your chromebook every day in this class.
- Writing Utensil - notes will be given out as hard copies and you are expected to follow along and annotate them.
- Three Ring Binder or Folder - notes, homework, etc will always have a three hole punch for you to keep organized in your binder.

Grading

- 60% Tests - AP Style Multiple Choice and Free Response Problems (handwritten code)
- 10% Unit Projects - AP Level Programming Assignments (typed code)
- 30% Daily Work - Homework, lesson activities, programming assignments, etc.

Reassessments

Reassessments are a way to demonstrate understanding at any time after a unit test and before the end of the semester. In order to reassess you must first submit test corrections. Your test corrections (written or typed) must meet the following criteria:

- Identify what your mistakes are and explained – This can include understanding of the question, operational errors, use of the wrong method, or something else. “I didn’t know how to answer the question” is not an acceptable answer. Your gaps/errors must be explained.
- Indicate how to solve the problem – Explain in a few words the process you will use to solve the problem.
- Work out the test problem(s) showing detail and arriving at a correct solution.

You cannot get help from the teacher on the same day you reassess. Show me you can retain understanding! Test corrections must be received and reviewed prior to reassessing.

You may reassess at any point during one of my availability hours that are posted on the board.

Student Responsibility

You must take responsibility for your own learning. Students will be given an agenda at the beginning of each new unit, and a copy will be posted in the Google Classroom. This agenda includes the focus of each day's lesson, as well as assignments to be completed. Students should expect homework to be assigned almost daily.

Makeup Work

If a student is absent, he/she will be assigned a grade of zero (0) for all worked missed. It is the student's responsibility to obtain and complete the work missed for each absence.

Check the class agenda to see what assignments were missed and what worksheets, if any, need to be picked up. Check the file for any worksheets and papers that may have been passed back in your absence.

Any assignments that were due on the date(s) of your absence are due immediately upon your return to class.

For any assignments that were given on the date(s) of your absence, you will be given 1 day per each day of your absence, plus an additional day to turn in those assignments.

If you are absent on days prior to the test, excluding review day, you are still responsible for taking the test. Any new material that was presented in your absence will not be counted against you on the test. However, you will be given an additional opportunity to demonstrate mastery of those concepts in the form of test questions.

If you are absent on review day or test day, you will take the test on the day of your return, during class time. You will be responsible for making alternative arrangements to catch up on material covered in class, via tutoring sessions, scheduled with me.

AP Exam Information

The AP Computer Science A Exam is a two-part test.

Section	Question Type	Num of Questions	Time Allowed	Percent of Final Grade
I	Multiple Choice	40	90 minutes	50%
II	Free Response	4	90 minutes	50%

Free Response Question Breakdown:

- Question 1: Methods and Control Structures
- Question 2: Class
- Question 3: Array/ArrayList
- Question 4: 2D Array

All students are expected and strongly encouraged to take the AP exam. Students scoring 3 or higher on the AP exam may qualify for a college course exception, dependent upon the college or university. Even if you do not score "well" on the exam, research shows that those students who take an AP exam, in addition to the course, are more likely to remain in college.

Help Sessions

If you find that you are confused on any topic, please do not hesitate to ask for help. Do not wait; it will only hurt you in the long run. Although there may not be enough time in class to get assistance, there are other means available to you.

Finally….

A strong association exists between regular class attendance and successful completion of the class. While it is understood that absences cannot be completely avoided, students are strongly encouraged to be present in school whenever possible, really and willing to learn.

So, put your best foot forward and let's have a great year!!

Course Sequencing

Unit 1: Primitive Types

Lesson 1: Output in Java

- Write your first lines of code in Java
- Use the System.out.print() and System.out.println() methods to print text and understand the difference between these commands
- Encounter the String data type, and know that strings are specified using double quotes

Lesson 2: User Input and Variables

- Know how to declare a variable and use it to store values
- Gain some understanding of what happens when a variable is declared
- Use string concatenation to create strings from multiple parts

Lesson 3: Data Types

- Write programs with variables of the int, double and char data types.
- Understand the importance of matching inputs to the correct data type.

Lesson 4: Number Calculations
- Perform calculations with int and double values
- Understand how integer division works in Java, and how to get a more precise value for a division if desired

Lesson 5: Modular Division
- Understand the definition of modular division and be able to manually calculate the value of modular expressions (e.g. 7%3)
- Use modular division in Java to solve problems

Lesson 6: Numeric Casts
- Perform casts between int and double data types in both directions
- Understand the difference between a widening conversion and a narrowing conversion

Unit 2 - Using Objects

Lesson 1: Creating and Storing Objects
- Explain the relationship between a class and an object.
- Identify, using its signature, the correct constructor being called.
- Create objects by calling constructors with and without parameters
- Define variables of the correct type to represent reference data

Lesson 2: Calling a Void Method
- Call non-static void methods without parameters.

Lesson 3: Calling a Void Method with Parameters
- Call non-static void methods with parameters.

Lesson 4: Calling a Non-Void Method
- Call non-static non-void methods with or without parameters.

Lesson 5: String Objects
- Learn the difference between primitive and class data types, and how Java stores these in computer memory
- Understand the effect that this has when declaring new Strings
- Use the keyword null to indicate an empty class data variable
- Understand that the + operator causes strings to be concatenated and integers to be added, and therefore data type matters
- Understand the purpose of escape characters in strings and write programs using several different escape characters

Lesson 6: String Methods
- Understand that a string is made up of characters which are stored with indices starting at 0 and increasing sequentially up to length-1
- Learn the purpose and behavior of several different functions from the String class and use these in code

Lesson 7: Wrapper Classes

- Understand why wrapper classes for primitive data types are useful
- Encounter code which makes use of the Integer and Double wrapper classes
- Understand autoboxing and unboxing, and when Java applies them

Lesson 8: Math Class

- Use commands from the Math class in Java
- Understand that extra functionality can be added to programs by importing new commands

Unit 3 - Boolean Expressions and If Statements

Lesson 1: Simple ifs

- Understand that programs can have commands which only run if certain conditions are true (e.g. based on the value of a variable)
- Write code using if statements in Java
- Understand the meaning of the operators ==, <, >, <=, >= and !=

Lesson 2: Else

- Write programs which use else and else if statements
- Understand how Java interprets a structure of multiple conditional statements

Lesson 3: Booleans and Truth Tables

- Understand the meaning of the boolean operators &&, ||, and !
- Write programs which use boolean operators to create compound boolean statements
- Use truth tables to determine when boolean expressions are true or false

Lesson 4: Short Circuit Evaluation

- Use short-circuit evaluation to save time when evaluating boolean statements
- Understand that Java uses short-circuit evaluation when evaluating boolean statements and how to take advantage of this

Lesson 5: DeMorgan's Law

- Use De Morgan’s law to write equivalent boolean statements
- Learn the inverses of the statements x == y, x < y and x > y

Lesson 6: Comparing Objects

- Understand the difference between the way Java compares primitives and objects

Unit 4 - Iteration

Lesson 1: While Loops

- Write code using while loops to repeat blocks of commands
- Understand that a while loop repeats as long as a boolean condition is true

Lesson 2: Algorithms with While Loops

- Identify and modify standard algorithms
- Develop algorithms using while loops

Lesson 3: for Loops
- Learn the syntax of a for loop in Java
- Write code which uses for loops to repeat commands

Lesson 4: Algorithms for Strings
- Identify and modify standard String Algorithms
- Develop algorithms on Strings

Lesson 5: Nested Loops
- Write and trace code which uses nested loops

Unit 5 - Writing Classes

Lesson 1: Anatomy of a Class
- Designate access and visibility constraints to classes, data, constructors, and methods.
- Designate private visibility of instance variables to encapsulate the attributes of an object.

Lesson 2: Constructors
- Define instance variables for the attributes to be initialized through the constructors of a class.
- Understand that objects are mutable.

Lesson 3: Accessor and Mutator Methods
- Define behaviors of an object through non-void methods without parameters written in a class.
- Define behaviors of an object through void methods with or without parameters written in a class

Lesson 4: Static Variables and Methods
- Define behaviors of a class through static methods.
- Define the static variables that belong to the class.
- Explain where variables can be used in the program code.

Lesson 5: this Keyword
- Compare two objects for data equality
- Evaluate object reference expressions that use the keyword this.

Unit 6 – Array

Lesson 1: One-Dimensional Arrays
- Learn what an array data type is, and how to declare and initialize an array
- Learn how to extract and edit data which is contained in an array
- Understand how array elements are indexed

Lesson 2: Algorithms – Searching
- Understand how to use a for loop to search for a value in an array and examine an implementation of this algorithm
- Examine other algorithms that use the same for loop structure to traverse an array

Lesson 3: Algorithms on Arrays
- Analyze examples of common algorithms which are used on arrays
- Be able to recognize, debug, and create the common array algorithms

Lesson 4: The for-each Loop and Arrays of Objects

- Write code which implements a for-each loop in Java,
- Understand the advantages and limitations of using a for-each loop versus a standard for loop to traverse an array
- Understand how to extract and edit arrays that contain objects.
- Write and trace code that creates and modifies arrays of objects.

Unit 7 - ArrayList

Lesson 1: ArrayList

- Understand that an ArrayList is used to hold multiple data as with an array, but that it can only hold class type data and not primitives
- Write code which declares and adds data to an ArrayList
- Learn some of the methods which can be called by objects of the ArrayList class

Lesson 2: Traversing ArrayLists

- Understand the advantages and limitations of using a for-each loop versus a standard for loop to traverse an ArrayList
- Understand how to safely modify the length of an ArrayList while traversing it.

Lesson 3: Methods with ArrayList

- Trace and write methods that use ArrayList as parameters.
- Trace and write methods that use ArrayList as return types.

Lesson 4: Linear and Binary Search

- Review the steps in the Linear Search algorithm for arrays and use with ArrayList
- Understand the Binary Search algorithm for arrays and ArrayList

Lesson 5: Selection Sort

- Understand how the selection sort algorithm orders an array of data
- Write code to implement the selection sort algorithm in Java
- Understand the advantages and disadvantages of using the selection sort algorithm to order the elements of an array

Lesson 6: Insertion Sort

- Understand how the insertion sort algorithm orders an array of data
- Understand the advantages and disadvantages of using the insertion sort algorithm to order the elements of an array

Unit 8 - 2D Arrays

Lesson 1: 2D Array

- Understand how the two-dimensional array structure in Java may be represented by a table with rows and columns

Lesson 2: Traversing 2D Array

- Learn how to declare and initialize and access data from a two-dimensional array in Java

Unit 9 - Inheritance

Lesson 1: Inheritance

- Learn how to create a subclass of an existing class
- Understand that public methods are inherited by the subclass of a class and can be accessed from within the class by using the keyword super

Lesson 2: Overridden Methods

- Understand that methods inherited by a subclass can be overwritten in the definition for that subclass (polymorphism)

Unit 10 - Recursion

Lesson 1: Intro to Recursion

- Understand that a method may call itself and that this is called recursion
- Trace code which uses recursion and predict its results (including recursion algorithms with return values)

Lesson 2: Binary Search

- Understand how the binary search algorithm finds a value in an ordered array of data
- Write code to implement the binary search algorithm in Java
- Understand the advantages and disadvantages of using the binary search algorithm to find data in an array

Lesson 3: Merge Sort

- Understand how the merge sort algorithm orders an array of data
- Understand the advantages and disadvantages of using the merge sort algorithm to order the elements of an array