**AP Computer Science**

**Canyon Crest Academy**

**Instructor:** Tony Mauro **Location:** F101

[Tony.mauro@sduhsd.net](mailto:Tony.mauro@sduhsd.net) **Tutor Time:** TBD – after school most days

**Course Description:**

AP® Computer Science A is both a college-prep course for potential computer science majors and a foundation course for students planning to study in other technical fields such as engineering, physics, chemistry, and geology. The course emphasizes programming methodology, procedural abstraction, and in-depth study of algorithms, data structures, and data abstractions, as well as a detailed examination of on the new AP CS Labs – Magpie, Elevens, and PictureLab. The course is divided into units of study which contain slides, exercises, programming projects, lab quizzes, and an end of unit exam. Instruction also includes preparation for the AP Computer Science A Exam.

**Expectations:**

You will be expected to acquire, to the best of your ability, the skills, knowledge, and techniques as they relate to this course. Additionally, such skills as technical writing, listening, thinking, and the use of technology will be improved through instruction, review and practice. Every student is expected to take the AP Exam.

**Lesson and Assignment Materials:**

All lesson materials including instructional slides and example programs will be available on a weekly basis from the Google Classroom site.

**Course Resources**

* The AP® Computer Science https[://apstudent.collegeboard.org/apcourse/ap-computer-science-a](https://apstudent.collegeboard.org/apcourse/ap-computer-science-a)
* Online Code Practice <http://codingbat.com>
* *Think Java - How to Think Like a Computer Scientist*, Allen B. Downey, <http://greenteapress.com/thinkapjava>

**Lab Time:**

* Stay in your seats and work on your assignments and projects. If you have a question and need assistance raise your hand and the teacher will assist you as soon as possible.
* If the teacher is busy write your name on the white board and the teacher will see in the order as your name is listed.

**Attendance:***Tardies:*All classes are consistent with the school's policy regarding tardies.

* 1 tardy:           Verbal Warning
* 2 tardies:        Conference with student and phone call home
* 3 tardies:        Referral

You are responsible for making up work missed due to excused absences. You must make arrangements with The Teacher before a planned absence or immediately upon returning to class. The Teacher will not remind you to make up missed assignments.

Regular on time attendance and active positive participation are essential in order to understand and appreciate the topics covered in this class. It is also a critical life-long work skill. Irregular attendance and/or excessive tardiness will seriously affect your achievement in this class and consequently your grade.

**Late Assignment Policies:**Any assignment not turned in during your class period on the due date will be considered late.

* Students must complete and turn in a Late Assignment Grade Request Form in order for that assignment to be graded.
* A grade of 0(zero) will be given for assignments not turned in or attempted.
* Partial credit may be given for Late Assignments turned in within a week of the due date. (50% for any Assignment & no more than 70% for any Assessment)
* Full credit may be given for Late Assignments due to an **Excused Absence** within a week of the due date.

**Academic Honesty:**Students must complete their own work for all assignments. If a student is found to have copied or turned in work that is not theirs a grade of Zero will be given.

**Grading:**

Scale: { A(100-90), B(89-80), C(79-70), D(69-60), F(59-0) }

Incomplete grades will not be given. Students are expected to follow the missing work policy listed below.  
A grade of 0(zero) will be given for assignments not turned in or attempted.

Partial credit may be given for assignments turned in within a week of the due date.

Grades should be available online through Aries. Grades are evaluated in the following categories and weights; **Participation -**(10%), **Projects/Labs -** (20%), **Multiple Choice, FRQ Assessments –** (40%), **Exams -**(40%).

**Class Rules:**

* No Food or Drinks
  + Bottled water is okay
* Get to class On-Time
  + In your seat logging in
* Respect Everyone & Everything
  + Do not be mean or insulting
  + Do not alter or break the equipment

**Leaving the Classroom:**

* If you need to use the bathroom just let the teacher know **before** you leave.
* If you have a pass to leave early or you are playing a sport, it is your responsibility to let the teacher before you leave. Please do not expect the teacher to remind you that you have an appointment.

**Turning In Assignments:**

* Everything you turn in needs to be in a folder labeled with your name Example: (tmauro\_Lab0a-c)
* Organize your assignments and projects in folders on your H: Drive and/or DropBox so that you will be able to find them easily.
* Instructional Labs are graded randomly in class.
* At the end of each week specified assignments will be collected.

**Acceptable use Policy:**

It is a privilege for the students to be able to use the school computers and network resources. Please read the policies written in the document linked below.

<http://www.sduhsd.net/documents/Policies/6000%20Instruction/6168.pdf>

Welcome to the class! Look forward to working with you

**Units of Study**

**------------------------ Week 1-2 ------------------------**

**Unit 0A – Computer Science, Computer Lab, and Objects Introduction**   
**AP Topics –** Test classes and libraries in isolation; Identify and correct errors : compile-time, run-time, logic; Categorize error: compile-time, run-time, logic; Employ techniques such as using a debugger, adding extra output statements, or hand-tracing code; Understand and modify existing code; Inheritance; Object-oriented development; Top-down development; Encapsulation & information hiding;   
  
**Student Objectives -** Students will learn what Computer Science is, how a computer lab works, how to use the computer, how the network is setup, and how to use the labs and the network in an acceptable/ethical manner. Students will learn the basic syntax for Java and how to debug a program, the difference between a compile error and a syntax error, how to identify and correct errors, how to add to and remove from existing code. Students gain experience working with a large program, modifying existing code of a large program, and expanding and extending existing code.   
  
**Unit 0B - Computers   
AP Topics –** Primary and secondary memory; processors; peripherals; language translators/compilers; virtual machines; operating systems; networks; single-user systems; networks; system reliability; privacy; legal issues and intellectual property; social and ethical ramifications of computer use.

**Student Objectives –** Students will learn all of the fundamental components of a computer, how a computer works, hardware, software, compilers, programming languages, basic computer operations, integrity, and responsible use of the computer.

**Units A and B – Output and Variables   
AP Topics –** Primitive types vs. Objects; Constant declarations; Variable declarations; Console output; Java library classes; Simple data types(int, boolean, double); Classes; Representations of numbers in different bases; Limitations of finite representations;

**Student Objectives –** Students will learn what a variable is, how to define a variable, how to assign values to a variable, the difference between a primitive type and a reference, and how to print/println values to the console window.

**Units C and 1 – Input and Methods   
AP Topics –** Variable declarations; Console output; Java library classes; Simple data types(int, boolean, double); Classes; Method declarations, Class declarations; Parameter declarations.

**Student Objectives –** Students will learn how to perform basic input operations, write methods, define and pass parameters, and use graphics to make shapes and pictures.

**Units 2 and 3 – Classes, OOP, Math Operations**

**AP Topics –** Object Oriented development; Top-down development; Encapsulation and information hiding; Procedural abstraction; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations.

**Student Objectives –** Students will learn how to declare a class, class methods, and parameters, the difference between constructors, accessors, and modifiers, learn how to read and understand a problem description, purpose, and goals. Students will learn to solve problems using mathematical operators(+,-,/,\*,%), mathematical formulas, and Math class methods.

**------------------------ Week 2-3 ------------------------**

**Unit 4 - Strings and OOP**

**AP Topics –** Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations; AP CS A Magpie lab introduction and extension.

**Student Objectives –** Students will learn how to instantiate a String, more about references, how to create a reference to a String, perform String input and output, how to use String methods(length, substring, indexOf, charAt), how to write return methods(toString), and how to create more sophisticated classes.

**Units 5-7 – Conditionals – If, If else, If else if, Switch Case**

**AP Topics –** Conditional; Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations AP CS A Magpie lab introduction and extension.

**Student Objectives –** Students learn how to use if, if else, if else if, and switch case to test conditions and add decision making to their programs, and Boolean conditions and variables. Students learn how to use relational operators(>,<,>=,<=,!).

**------------------------ Week 3-5 ------------------------**

**Units 8-9 – Iteration – For Loop and While Loop**

**AP Topics –** Iteration; Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations.

**Student Objectives –** Students learn how to use for loops, use while loops, add iterative processes to their programs, and use Boolean conditions and variables. Students learn the different parts of a loop and when to use a particular type of loop. Students will learn when to use Integer.MAX\_VALUE and Integer.MIN\_VALUE.

**Unit 10 - Boolean Logic and Boolean Laws**

**AP Topics –** Boolean; Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations.

**Student Objectives –** Students learn boolean laws, truth tables, logical operators(&&, ||, !, ^), how to use do while loops, how to use boolean logic to solve problems, and how to use Random and Math.random() to generate random numbers.

**Unit 11 - Iteration – Nested Loops**

**AP Topics –** Iteration; Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations;

**Student Objectives –** Students learn how to use nested loops, add iterative processes to their programs, and use Boolean conditions and variables. Students learn how to use nested for and nested while loops.

**Units 14-15 - One dimensional arrays [!!Critical Topic!!]**

**AP Topics –** One-dimensional arrays; Traversals; Insertions; Deletions; Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations.

**Student Objectives –**  Students will learn how to instantiate a one-dimensional array, add items to a one-dimensional array, delete items from a one-dimensional array, and use a one-dimensional array to solve problems. Students will learn the differences between arrays of primitives and arrays of references.

**------------------------ Week 5-7 ------------------------**

**Unit 16 - ArrayList [!!Critical Topic!!]**

**AP Topics –** One-dimensional arrays; Traversals; Insertions; Deletions; Object Oriented development; Searching; Sorting; Test classes and libraries in isolation; Identify boundary cases and generate appropriate test data; Perform integration testing; Choose appropriate data representation and algorithms; extensive work with AP CS A Elevens Lab to assist in teaching lists of references.

**Student Objectives –** Students will learn how to add to, delete from, sort, search, and perform all types of manipulations on an ArrayList. Students will learn about the java.util.List interface. Students gain experience working with a large program, modifying existing code of a large program, and expanding and extending existing code.

**Unit 17 – References / Parameters   
AP Topics –** Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations; AP CS A Elevens Lab modification and extension.

**Student Objectives –** Students will learn more about references and parameter passing. Students will learn the differences between passing primitives and references as parameters.

**Unit 18 - OOP / Array of References / List of References [!!Critical Topic!!]  
AP Topics –** Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations; One-dimensional arrays; Traversals; Insertions; Deletion; Object Oriented development; Read and understand a problem description, purpose, and goals; Class design;

**Student Objectives –** Students gain experience working with existing code and modifying, expanding, and extending existing code. Students will learn more about storing references in arrays. Students will learn the difference between arrays of primitives and arrays of references.

**------------------------ Week 8-9 ------------------------**

**Unit 19 - Interfaces  
AP Topics –** Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations; Interface declarations; New AP CS A Labs – Elevens and Magpie

**Student Objectives –** Students will learn how to design and implement a class; apply data abstraction and encapsulation; and implement an interface and learn why interfaces are useful. Students will learn how interfaces are used to build hierarchies. Students gain experience working with a large program, modifying existing code of a large program, and expanding and extending existing code.

**Unit 20 – Inheritance [!!Critical Topic!!]**

**AP Topics –** Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations; Interface declarations; Read and understand class specifications and relationships among the classes("is-a", "has-a" relationships); Understand and implement a class hierarchy; Identify reusable components from existing code using classes and class libraries; Choose appropriate data representation and algorithms.; Extend a class using inheritance; work with *AP CS A Labs – Magpie and Elevens.*

**Student Objectives –** Students will learn how to extend a given class using inheritance, design and implement a class hierarchy, write a multi-tiered game with graphics and animation. Students will learn how to build a new class from an existing class using extends and super calls. Students will learn how to use static variables. Students gain experience working with a large program, modifying existing code of a large program, and expanding and extending existing code.

**------------------------ Week 10-12 ------------------------**

**Unit 21 - Abstract Classes – BlackJack - Elevens**

**AP Topics –** Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations; Interface declarations; Read and understand class specifications and relationships among the classes("is-a", "has-a" relationships); Understand and implement a class hierarchy; Identify reusable components from existing code using classes and class libraries; Choose appropriate data representation and algorithms; Extend a class using inheritance. Use a List of classes to build a group of objects.

**Student Objectives –** Students will learn how to design and implement an abstract class, extend an abstract class to make sub classes, and implement an interface. Students will learn to compare and contrast a class, an interface, and an abstract class. Students will learn when to use an interface, when to use an abstract class, and when to use static variables.

**Unit 22 – Matrices [!!Critical Topic!!]**

**AP Topics –** One-dimensional arrays; Two-dimensional arrays; Traversals; Insertions; Deletions; Object Oriented development; Class design; Method declarations; Parameter declarations; Class declarations; Top-down development; Encapsulation & information hiding; Introduction and Extended work with AP CS A Lab PictureLab.

**Student Objectives –** Students will learn how to instantiate a one-dimensional and two-dimensional array, add items to a one-dimensional and two-dimensional array, and delete items from a one-dimensional and two-dimensional array.

**Unit 23 - Recursion**

**AP Topics –** Recursion; Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Class declarations; work with *AP CS A Labs – Elevens and PictureLab.*

**Student Objectives –** Students will learn how to use recursion to solve problems, the benefits of using recursion, when to use recursion, and the negative effects of using recursion. Students gain experience working with a large program, modifying existing code of a large program, and expanding and extending existing code.

**------------------------ Week 13-14 ------------------------**

**Unit 24 - Advanced Sorting and Searching / Comparable**

**AP Topics –** One-dimensional arrays; Traversals; Insertions; Deletions; Object Oriented development; Searching; Sequential Search; Binary Search; Sorting; Selection Sort; Insertion Sort; Merge Sort; Test classes and libraries in isolation; Identify boundary cases and generate appropriate test data; Perform integration testing; Choose appropriate data representation and algorithms; Analysis of algorithms; Informal comparisons of running times; Exact calculation of statement execution counts.

**Student Objectives –** Students will learn to identify all sorting and searching algorithms, code all sorting and searching algorithms, and to select the appropriate sorting and searching algorithm for the appropriate situation. Students will learn where to use a particular sort/search and the benefits of using a particular type of sort/search.

**AP Review Time**

*AP Topics : Arrays, ArrayList, Inheritance, AP CS A Labs – Magpie, Elevens, and PictureLab  
Guided Practice : Past year’s free response and multiple choice questions*

*Guided Practice : Slides, AP CS A Labs – Magpie, Elevens, and PictureLab*

*Readings : Past year’s free response and multiple choice questions*

*Readings : Review book units*

**Units 12-13 – Chopping Strings and File Input ( Optional Topics )**

**AP Topics –** Object Oriented development; Read and understand a problem description, purpose, and goals; Class design; Method declarations; Parameter declarations; Class declarations.

**Student Objectives –**  Students learn how to use Scanner to chop up Strings, to read data from data files, to instantiate Objects using the data extracted from files. Students learn more about constructor overloading and using a single class for multiple purposes.