AP Computer Science Principles Syllabus 2024-25 Course Information

Course Title: AP Computer Principles A

No. of Credits: 5.0

Course Length: Full year

Grade Levels: 9 – 12

Instructor Information

Instructor: Sokha Tee

Classroom: 40

Office Hours: 8:30 A.M – 4:00 P.M

E-mail: stee@compton.k12.ca.us

Phone: (310) 800-7767 Xt. 68290

Resource sites: Google Classroom

www.sokhatee.com

Welcome to Computer Science.

It's awesome, **it's** everywhere, **it's** for everyone. Today, every technology and field are under the umbrella of Computer Science. Everyone in this world is absolutely interacting with and affected by this technology every day, no matter who you are. You can learn computer science to harness the Power of this technology, and apply it to whatever you are interested in. Computer science is affecting every industry and every field. It's really affecting every aspect of our modern life. Computer science is all about **Innovation**, this technology allows you to create things that exist in the world that being used every day. For example, Apps, social media, AI etc. Computer science is basically all the innovations that are impactful on our everyday lives.

Course Objectives:

AP Computer Science Principles is intended to simulate an introductory college computing course. This course demonstrates the relevance of computer science by highlighting the importance of computing in society. Students will utilize Big 5 ideas that are central to computer science to sharpen their computational skills by analyzing, visualizing, planning and drawing conclusions from trends in large data sets. Students are asked to think creatively to solve problems and analyze computer data to forecast a future trend. This class learns how computer science technology affects today's society. Scholars will study machines language and programing language; they will learn how to convert binary number system to decimal system. Also, Students will develop computational thinking skills necessary for success in many disciplines such as software or platform, and other on-line apps to write algorithms in Python program language. The course endeavors to teach students to be creative and to use the creative process to solve computational problems. Students will construct and implement solutions to complex problems like what computer scientists and computer engineers do. Finally, students will be learning how to create apps and get students prepared for the College Board Exam.

THE FIVE BIG IDEAS

BIG IDEA 1 CREATIVE DEVELOPMENT
BIG IDEA 2 DATA
BIG IDEA 3 ALGORITHMS AND PROGRAMMING

BIG IDEA 4 COMPUTING SYSTEMS AND NETWORKS
BIG IDEA 5 IMPACT OF COMPUTING

Computational Thinking Practice 1: Computational Solution Design

- Computational Thinking Practice 2: Algorithms and Program Development
- Computational Thinking Practice 3: Abstraction in Program Development
- Computational Thinking Practice 4: Code Analysis
- Computational Thinking Practice 5: Computing Innovations
- Computational Thinking Practice 6: Responsible Computing

Essential Questions

- How can computing, and the use of computational tools foster creative expression?
- How does abstraction help us in writing programs, creating computational artifacts, and solving problems?
- How can computational models and simulations help generate new understanding and knowledge?
- How can computation be employed to facilitate exploration and discovery when working with data?
- What opportunities do large data sets provide for solving problems and creating knowledge?
- How are algorithms implemented and executed on computers and computational devices?
- Why are some languages better than others when used to implement algorithms?
- How are programs developed to help people, organizations, or society solve problems?
- How are programs used for creative expression, to satisfy personal curiosity, or to create new knowledge?
- How does abstraction make the development of computer programs possible?
- Which mathematical and logical concepts are fundamental to computer programming?
- What is the Internet? How is it built? How does it function?
- How is cybersecurity impacting the ever-increasing number of Internet users?
- How does computing enhance human communication, interaction, and cognition?
- How does computing enable innovation?
- What are some potential beneficial and harmful effects of computing?
- What are creativities in programming process?

<u>Computational Thinking Practices</u>

P1: Connecting Computing

P2: Creating Computational Artifacts
P3: Abstract
P4: Analyzing Problems and Artifacts
P5: Communicating
P6: Collaborating

Ethics and Technology

In addition to the previously mentioned areas of focus, this class will emphasize the importance of ethical practices when working with technology. This is an important area, which deserves special attention, and will be woven throughout the framework of the course. The intention is to address each of the issues numerous times, touching on at least on issue per lesson, and to use case studies and examples and discussion points. The ethical area that the class will look at are:

- Responsible and Ethical Use
- Privacy
- Economic and Legal Implications
- Safety and Harassment
- Intellectual Property Rights

Guiding Practices / Ideas

The learning in AP Computer Science Principles is guided by six Computational Thinking Practices and seven Big Ideas. (from AP Computer Science Principles: Course and Exam Description Effective Fall 2016, College Board).

Grading

- Assignments: 30%
- Quizzes: 30%
- Tests/Projects/Performance Tasks: 40%

Required Supplies

- Three-ring binder for handouts, notes, and assignments
- Google Drive account or USB flash drive
- Reliable access to a computer outside of class

Course Expectations

- Students will be required to work on Basic Python programming, both class-works and home works are done on Tech-smart plat form. All students should access to Tutorial, Projects, Quiz and Test through Tech Smart platform. Student is required to work on Game Design Unity/Unreal, Scratch MIT's platform, Adobe Suites.

- Effort:

- Learn:

- Do not download anything onto the computers unless you are explicitly told to do so.

- **NO FOOD OR DRINK AROUND THE COMPUTERS**!
- Bathroom - Only 2 students may leave the classroom at a time. No Exceptions! Students will use the sign out sheet and hall pass when leaving the classroom for any reason. Minimize time out of the classroom.
- Cell Phones – Class is not the time or place to be using your cell phone for personal reasons. All cell phones are to be placed in the cell phone holder for the duration of class.
- Watching videos, playing games, or other inappropriate use of computers during class time will not be tolerated.
- Plagiarism of any kind, including attempting to pass off someone else's code as your own will result in an automatic zero for the assignment and a referral. See Academic Integrity on the last page.
- Attendance is mandatory. Students who miss class for any reason are expected to make up missed work on their own time.

AP Test

The AP Computer Science Principles course has three assessments consisting of two performance tasks and an end-of course multiple-choice AP Exam. These assessments are summative and the scoring results from each will be used to calculate a final AP score using the 1 – 5 scale. Each assessment will count for a certain percentage of the total AP Sore: Explore Performance Task (8 hours) 16%; Create Performance Task (12 hours) 24%, End-of-Course Exam (2 hours) 60%. All students will complete all three parts of the AP Exam.

Explore Performance Task: Computing innovations impact our lives in ways that require considerable study and reflection for us to fully understand them. In this performance task, students will explore a computing innovation of their choice. The close examination of a computing innovation will deepen the students' understanding of a computer science principle.

Create Performance Task: Programming is a collaborative and creative process that brings ideas to life through the development of software. Programs can help solve problems, enable innovations, or express personal interests. In this performance task, students will be developing a program of their choice. The students' development process should include iteratively designing, implementing, and testing their program. Students are strongly encouraged to work with another student in the class.

Methodology

The class will be primarily based on the College Board endorsed curriculum developed by Code.org which is designed specifically for use in AP Computer Science Principles. Other resources, such as Beauty and Joy of Computing (BJC) curriculum developed at the University of California and Harvard's CS50, will be used to supplement the course along with Code.org. The course is broken down into 6 units each containing multiple topics, reading assignments,

research assignments, writing assignments, and programming problems. Below is the calendar we will follow.

Unit (weeks)	Title	Topics
		Semester 1
1	**Computers and the Internet**	Computer Architecture, Number Systems, Internet Simulator, Addressing, Routers, Packets, DNS, HTTP, Websites, Search Engine, SEO,
2	**Digital Information**	Bits and Bytes, Files size, Compression, Color, Data. Data Visualization, Data Story, Cleaning Data, DNS, Google Data and Meta data
3	**Intro to Programming**	Programming Languages (Python). Statements & Variables, Libraries, Values (datatype), Operation & Symbols. Algorithms, Commands, Conditional, String, Class, Instance, Concatenation.
4	**Explore Performance Task**	A minimum of 8 hours of class time will be dedicated to completing the Explore Performance Task (16% of AP Score)
		Semester 2
5	**Big Data and Privacy**	Big Data, Privacy, Encryption, Asymmetric Encryption, Cybercrime, Data Security. Passwords, Binary System.
6	**Building Apps**	Buttons, Events, Screens, Variables, Input, Strings, Control Structures, Booleans, Loops, Arrays, Return Values.
7	**Create Performance Task**	A minimum of 12 hours of class time will be dedicated to completing the Create Performance Task (24% of AP Score)
8	**Expanding Apps**	Objects, Data Storage, Reading, Google Map, Cloud Storage

9	**Collaboration Group Project**	Everyone works in a group of four people or more to collaborate more ideas to prepare for College Board Final Exam.
10	**Create Apps**	Code.org

Academic Integrity

This course takes academic integrity quite seriously. Be assured that tools exist that make it trivially simple to detect cases of academic dishonesty, and as such this course's philosophy on academic integrity is best stated as "be reasonable." The course recognizes that interactions with classmates and others can facilitate mastery of the course's material. However, there remains a line between enlisting the help of another and submitting the work of another. This policy endeavors to characterize both sides of that line.

The essence of all work that you submit to this course must be your own. Collaboration on problems is not permitted (unless explicitly stated otherwise) except to the extent that you may ask classmates and others for help so long as that help does not reduce to another doing your work for you. Generally speaking, when asking for help, you may show your code or writing to others, but you may not view theirs.

Collaboration on any quizzes and tests is not permitted at all. Collaboration on the Create Performance Task is permitted to the extent prescribed by its specifications.

Below are examples that inexhaustibly characterize acts that the course considers reasonable and not reasonable. If in doubt as to whether some act is reasonable, do not commit it until you solicit and receive approval in writing from your instructor. If a violation of this policy is suspected and confirmed, your instructor reserves the right to impose an appropriate penalty.

Reasonable

- Communicating with classmates about problems in English (or some other spoken language). • Discussing the course's material with others in order to understand it better.
- Helping a classmate identify a bug in his or her code, such as by viewing, compiling, or running his or her code, even on your own computer.
- Incorporating snippets of code that you find online or elsewhere into your own code, provided that those snippets are not themselves solutions to assigned problems and that you cite the snippets' origins (as via comments in your code). • Sending or showing code that you've written to someone, possibly a classmate, so that he or she might help you identify and fix a bug.
- Sharing snippets of your own solutions to problems online so that others might help you identify and fix a bug or other issue.

- Turning to the web or elsewhere for instruction beyond the course's own, for references, and for solutions to technical difficulties, but not for outright solutions to problems or your own final project.
- • Whiteboarding solutions to problems with others using diagrams or pseudocode but not actual code.
- Working with (and even paying) a tutor to help you with the course, provided the tutor does not do your work for you.

Not Reasonable

- Accessing a solution to some problem prior to (re-)submitting your own.
- Asking a classmate to see his or her solution to a problem before (re-)submitting your own.
- Failing to cite (as with comments) the origins of code, writing, or techniques that you discover outside of the course's own lessons and integrate into your own work, even while respecting this policy's other constraints.
- Giving or showing a classmate a solution to a problem when it is he or she, and not you, who is struggling to solve it.
- Looking at another individual's work during a quiz or test. • Paying or offering to pay an individual for work that you may submit as (part of) your own.
- Providing or making available solutions to problems to individuals who might take this course in the future.
- Searching for, soliciting, or viewing a quiz's questions or answers prior to taking the quiz.
- Searching for or soliciting outright solutions to problems online or elsewhere.
- Splitting a problem's workload with another individual and combining your work (unless explicitly authorized by the problem itself). • Submitting (after possibly modifying) the work of another individual beyond allowed snippets.
- Submitting the same or similar work to this course that you have submitted or will submit to another.
- Using resources during a quiz beyond those explicitly allowed in the quiz's instructions.
- Viewing another's solution to a problem and basing your own solution on it.