La Cañada High School

Proposed Course Outline – AP Computer Science Principles

I. **Course Title – AP Computer Science Principles**

II. **Grade Level(s) – Grades 10-12**

III. **Length/Credit – 1 Year - 10.0 units Laboratory Science**

IV. **Preparations –** Successful completion of LC Math 1 or LC Math 1 Advanced, with a strong foundation in basic linear functions and 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 in a plane and employ mathematical and computational reasoning across the course.

V. **Course Description**

The AP Computer Science Principles course is designed to be equivalent to a first-semester introductory college computing course. In this course, students will develop computational thinking skills vital for success across all disciplines, such as using computational tools to analyze and study data and working with large data sets to analyze, visualize, and draw conclusions from trends. The course engages students in the creative aspects of the field by allowing them to develop computational artifacts based on their interests. Students will also develop effective communication and collaboration skills by working individually and collaboratively to solve problems, and will discuss and write about the impacts these solutions could have on their community, society, and the world. The major areas of study in the course are organized around seven big ideas, which encompass ideas foundational to studying computer science and are followed by a series of essential questions which students engage with across the course. Essential to the composition of the course are a set of standards in two specific domains: computational thinking and enduring understanding. These are detailed below.

VI. **Standards/ESLRs**

Computational Thinking

P1: Connecting Computing

Developments in computing have far-reaching effects on society and have led to significant innovations. The developments have implications for individuals, society, commercial markets, and innovation. Students in this course study these effects, and they learn to draw connections between different computing concepts.

P2: Creating Computational Artifacts

Computing is a creative discipline in which creation takes many forms, such as remixing digital music, generating animations, developing websites, and writing programs.

Students in this course engage in the creative aspects of computing by designing and developing interesting computational artifacts as well as by applying computing techniques to creatively solve problems.

P3: Abstracting

Computational thinking requires understanding and applying abstraction at multiple levels, such as privacy in social networking applications, logic gates and bits, and the human genome project. Students in this course use abstraction to develop models and simulations of natural and artificial phenomena, use them to make predictions about the world, and analyze their efficacy and validity.

P4: Analyzing Problems and Artifacts

The results and artifacts of computation and the computational techniques and strategies that generate them can be understood intrinsically both for what they are as well as for what they produce. They can also be analyzed and evaluated by applying aesthetic, mathematical, pragmatic, and other criteria.

P5: Communicating

Students in this course describe computation and the impact of technology and computation, explain and justify the design and appropriateness of their computational choices, and analyze and describe both computational artifacts and the results or behaviors of such artifacts. Communication includes written and oral descriptions supported by graphs, visualizations, and computational analysis.

P6: Collaborating

Innovation can occur when people work together or independently. People working collaboratively can often achieve more than individuals working alone. Learning to collaborate effectively includes drawing on diverse perspectives, skills, and the backgrounds of peers to address complex and open-ended problems.

Enduring Understanding Standards (EU 1-7)

Big Idea 1: Creativity

EU 1.1 Creative development can be an essential process for creating computational artifacts.

EU 1.2 Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.

EU 1.3 Computing can extend traditional forms of human expression and experience.

Big Idea 2: Abstraction

EU 2.1 A variety of abstractions built on binary sequences can be used to represent all digital data.

EU 2.2 Multiple levels of abstraction are used to write programs or create other computational artifacts.

EU 2.3 Models and simulations use abstraction to generate new understanding and knowledge.

Big Idea 3: Data and Information

EU 3.1 People use computer programs to process information to gain insight and knowledge.

EU 3.2 Computing facilitates exploration and the discovery of connections in information.

EU 3.3 There are trade-offs when representing information as digital data.

Big Idea 4: Algorithms

EU 4.1 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages.

EU 4.2 Algorithms can solve many, but not all, computational problems.

Big Idea 5: Programming

EU 5.1 Programs can be developed for creative expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations,or society).

EU 5.2 People write programs to execute algorithms.

EU 5.3 Programming is facilitated by appropriate abstractions.

EU 5.4 Programs are developed, maintained, and used by people for different purposes.

EU 5.5 Programming uses mathematical and logical concepts.

Big Idea 6: The Internet

EU 6.1 The Internet is a network of autonomous systems.

EU 6.2 Characteristics of the Internet influence the systems built on it.

EU 6.3 Cybersecurity is an important concern for the Internet and the systems built on it.

Big Idea 7: Global Impact

EU 7.1 Computing enhances communication, interaction, and cognition.

EU 7.2 Computing enables innovation in nearly every field.

EU 7.3 Computing has global effects — both beneficial and harmful — on people and society.

EU 7.4 Computing innovations influence and are influenced by the economic, social, and cultural contexts in which they are designed and used.

EU 7.5 An investigative process is aided by effective organization and selection of resources. Appropriate technologies and tools facilitate the accessing of information and enable the ability to evaluate the credibility of sources.

VII. Brief Course Outline

1. **Big Idea 1: Creativity**
   Computing is a creative activity. Creativity and computing are prominent forces in innovation; the innovations enabled by computing have had and will continue to have far-reaching impact. At the same time, computing facilitates exploration and the creation of computational artifacts and new knowledge that help people solve personal, societal, and global problems. This course emphasizes the creative aspects of computing. Students in this course use the tools and techniques of computer science to create interesting and relevant artifacts with characteristics that are enhanced by computation.

2. **Big Idea 2: Abstraction**
   Abstraction reduces information and detail to facilitate focus on relevant concepts. Everyone uses abstraction on a daily basis to effectively manage complexity. In computer science, abstraction is a central problem-solving technique. It is a process, a strategy, and the result of reducing detail to focus on concepts relevant to understanding and solving problems. This course requires students to use abstractions to model the world and communicate with people as well as with machines. Students in this course learn to work with multiple levels of abstraction while engaging with computational problems and systems; use models and simulations that simplify complex topics in graphical, textual, and tabular formats; and use snapshots of models and simulation outputs to understand how data changes, identify patterns, and recognize abstractions.

3. **Big Idea 3: Data and Information**
   Data and information facilitate the creation of knowledge. Computing enables and empowers new methods of information processing, driving monumental change across many disciplines — from art to business to science. Managing and interpreting an overwhelming amount of raw data is part of the foundation of our information society and economy. People use computers and computation to translate, process, and visualize raw data and to create information. Computation and computer science facilitate and enable new understanding of data and information that contributes knowledge to the world. Students in this course work with data using a variety of computational tools and techniques to better understand the many ways in which data is transformed into information and knowledge.

4. **Big Idea 4: Algorithms**
   Algorithms are used to develop and express solutions to computational problems.

Algorithms are fundamental to even the most basic everyday task. Algorithms realized in software have affected the world in profound and lasting ways. Secure data transmission and quick access to large amounts of relevant information are made possible through the implementation of algorithms. The development, use, and analysis of algorithms are some of the most fundamental aspects of computing. Students in this course work with algorithms in many ways: they develop and express original algorithms, they implement algorithms in a language, and they analyze algorithms analytically and empirically.

5. **Big Idea 5: Programming**
   Programming enables problem solving, human expression, and creation of knowledge. Programming and the creation of software has changed our lives. Programming results in the creation of software, and it facilitates the creation of computational artifacts, including music, images, and visualizations. In this course, programming enables exploration and is the object of study. This course introduces students to the concepts and techniques related to writing programs, developing software, and using software effectively. The particular programming language is selected based on appropriateness for a specific project or problem. The course acquaints students with fundamental concepts of programming that can be applied across a variety of projects and languages. As students learn language specifics for a given programming language

6. **Big Idea 6: The Internet**
   The Internet pervades modern computing. The Internet and the systems built on it have had a profound impact on society. Computer networks support communication and collaboration. The principles of systems and networks that helped enable the Internet are also critical in the implementation of computational solutions. Students in this course gain insight into how the Internet operates, study characteristics of the Internet and systems built on it, and analyze important concerns such as cybersecurity.

7. **Big Idea 7: Global Impact**
   Computing has global impact. Computation has changed the way people think, work, live, and play. Our methods for communicating, collaborating, problem solving, and doing business have changed and are changing due to computing innovations, which are innovations that include a computer or program code as an integral part of their function. Many innovations in other fields are fostered by advances in computing. Computational approaches lead to new understandings, new discoveries, and new disciplines. Students in this course become familiar with many ways in which computing enables innovation, and they analyze the potential benefits and harmful effects of computing in a number of contexts.

VIII. Methods of Assessment

Evaluation:

Students are assessed with learning opportunities that fall into one of four categories listed below. The learning opportunities provide students with chances to engage in assignments that will prepare them to successfully complete the Advanced Placement Computer Science Principles exam and performance tasks. The categories are specified below.

Grades:

All work will be assigned a point value, although not all work will receive a letter grade. Grades are based on total points accumulated during each grading period. I have structured the class in such a way to approximate your grade breaking down into the following percentages:

Examinations/Performance Tasks:	**50%**	**A-/A-= 89.5-100 %**
Homework/Classwork:	**10%**	**B-/B/B+ - 79.5-89.4**
Class Participation/Engagement	**25%**	**C-/C/C+ = 69.5- 79.4%**
Final Exam:	**15%**	**D-/D/D+ + 59.5 – 69.4 %**
		F = Below 59.5 %

IX. Materials/Textbook(s)

The College Board has endorsed innovative curricula that come with pre-approved syllabi, lesson plans, and other instructional supports, delivered by expert education organizations. These curricula include professional development to prepare teachers, whether experienced or new to computer science, to teach the AP Computer Science Principles course. A sample curriculum provider being considered by LCHS teachers is from CODEHS, which provides students the opportunity to explore several topics of computing in JavaScript. Information can be found at codehs.com/course/catalog.

X. Seeking "a-f" Approval – Yes/No – Yes, this course will be submitted to the University of California for approval for the 2020-21 academic year in the subject domain "D" for Science.

XI. Seeking AP Class Approval – Yes/No – This course seeks AP approval.