Livermore Valley Joint Unified School District

Course Title:	Advanced Placement (AP) Computer Science Principles A/B
Grade Level(s):	10-12
Length of Course:	Two semesters or equivalent term
Credit:	10 units
Prerequisite:	Completion of Exploring Computer Science or Introduction to Computer Programming
Co-requisite:	None

Course Overview:

AP Computer Science Principles offers a multidisciplinary approach to teaching the underlying principles of computation. Through the course, students will be introduced to the creative aspects of programming, abstractions, algorithms, large data sets, the Internet, cybersecurity concerns, and computing impacts. Students will also have the opportunity to use current technologies to create computational artifacts for both self-expression and problem solving. Together, these aspects of the course make up a rigorous and rich curriculum that aims to broaden participation in computer science. Through the AP Computer Science Principles course, students will be provided with valuable skills that will benefit them in many future areas of study and careers. Using Python® as a primary tool and incorporating multiple platforms and languages for computation, students will develop computational thinking, learn about career paths that utilize computing, and will be introduced to professional tools that foster creativity and collaboration. Through this course, students will complete the AP Computer Science Principles Digital Portfolio and will gain support in learning the standards on the AP exam. Students who successfully complete the course and AP exam may receive credit, advanced placement, or both for a one-semester introductory college computer science course.

Schools Offering:	Del Valle High School Granada High School Livermore High School Vineyard High School Phoenix High School
Meets University of California Entrance Requirements:	"g"- Elective, Mathematics-Computer Science *Counts as a student's 3rd or 4th recommended year of UC/CSU laboratory science.
Board Approval:	*Pending Board Approval*

Course Materials:	Computer Science Illuminated 7th Edition Dale, Nell & Lewis, John Jones and Bartlett Learning, 2020 ISBN-13: 978-1284155617 Industry standard languages and applications, which are open-source and free to install and use.
Supplemental Materials:	A small set of hardware tools will be used throughout the course

ADVANCED PLACEMENT (AP) COMPUTER SCIENCE PRINCIPLES

COURSE CONTENT:

Unit 1: Algorithms, Graphics, and Graphical User Interfaces

The goal of Unit 1 is to excite students about programming and to build their algorithmic thinking and ability to use abstraction. Student creativity is emphasized as they work with ScratchTM, AppInventor, and Python® programming languages to tell graphical stories, publish games and AndroidTM applications, and explore various development environments and programming techniques. Students will create original code and read and modify code provided from other sources. An Agile software development process is emphasized, and personal, professional, and collaborative skills take center stage. Students debate policy questions about the ownership and control of digital data and examine the implications for creative industries and consumers. In this unit, students begin their exploration of career paths tied to computing.

Summary of Key Assignments, Labs and/or Activities

Lesson 1.1: Algorithms and Agile Development

The goal of this lesson is to introduce students to programming at a level appropriate for novice programmers. With an introduction to pair programming and the Agile software development process, students create original programs in Scratch that incorporate audio and visual elements while tackling algorithmic problems. The lesson opens with an introduction to how computing is affecting our lives. Students explore tools for collaboration over the Internet and select from these tools in order to manage the projects that they create. The foundations for later algorithmic thinking are built by focusing on the most common roles that variables fulfill, with an introduction to the conventions of object-oriented programming.

Lesson 1.2: Mobile App Design

The goal of this lesson is for students to build their skills by analyzing existing code, particularly with an emphasis on the roles of variables. Students create an Android app of their own design. The lesson begins with an introduction to binary representations of numbers, letters, colors, images, etc., using a CS unplugged activity in which students create a physical representation of data storage.

Students work with and make minor modifications to two App Inventor programs, building their ability to analyze a complex program and incorporate event handlers into programs in meaningful ways. Students conclude by designing and creating their own Android app, using pair programming and practicing the Agile software design process.

Lesson 1.3: Algorithms in Python
The goal of this lesson is for students to understand all information as bits and to transfer their understanding of algorithms to a new language, Python, which is powerful enough to raise all the opportunities and issues targeted in the course. Students are introduced to functional, imperative, and declarative programming paradigms with Python, again learning to use variables in the most common roles. Before learning about variable types and the fundamental algorithmic structures in Python, students simulate program execution in a model assembly language. After building strength with basic Python algorithms, students create algorithms to compete a round-robin tournament of the Prisoner's Dilemma, using the collaborative programming platform GitHub in the process.

Lesson 1.4: Images and Object-Oriented Libraries
The goal of this lesson is for students to become independent learners of a programming language, able to refer to documentation to use object-oriented libraries commonly available. The lesson begins with an unplugged activity to teach object-oriented concepts. Students build additional strength with Python algorithms, manipulating image files by modifying pixel data and using code libraries to work at higher levels of abstraction. As part of that work, they learn to use a variety of documentation including application-programming interfaces (APIs). Students read, discuss, and debate intellectual property issues associated with digital data. In the culminating problem of the lesson, they collaborate to create an image processing function that highlights the power of automation.

Lesson 1.5: GUIs in Python
The goal of this lesson is for students to conceive of any class of objects as an abstraction. Students will create a graphical user interface (GUI) with considerations of audience and accessibility. The lesson begins with an unplugged activity that generalizes the user interface topic of this lesson to the field of human-computer interaction. Students practice using an application-programming interface (API) to acquire methods that affect an object's state. Students work with two APIs: the Tkinter Canvas for drawing and animation, and then the Tkinter toolbox of GUI widgets. Students are provided code for a simple GUI that implements a model-view-controller (MVC) pattern. Students will modify the elements of that pattern to suit their own needs. The lesson concludes with a problem in which students create a model-view-controller GUI using Scratch or Python. Strategies for documentation are reinforced, and Agile development is emphasized in the concluding problem.

Unit 2: The Internet

The goal of Unit 2 is for students to have a more concrete understanding of the Internet as a set of computers exchanging bits and the implications of these exchanges. Students use PHP and SQL to structure and access a database hosted on a remote server, learn how HTML and CSS direct the client computer to render a page, and experiment with JavaScriptTM to provide dynamic content. The focus of the unit is on the protocols that allow the Internet to function securely to deliver social media and eCommerce content. Students work briefly in each of several Web languages to understand how the languages work together to deliver this content. The history and workings of the Internet are explored, and issues of security, privacy, and democracy are considered. Practical cyber

security hygiene is included. Career paths in cyber security, web development, and information technology are highlighted.

Summary of Key Assignments, Labs and/or Activities

Lesson 2.1: The Internet and the Web

In this lesson the goal is to build student understanding of the Internet as a set of computers exchanging bits in the form of packets. Students will learn to identify the components of their digital footprint. To provide a hook, students compare the designs, strengths, and weaknesses of their favorite web pages. In this context students use an unplugged activity to understand (in broad brushstrokes) the content and flow of data when browsing the Web. They compare results from different search engines and learn to refine their search techniques. They review how to assess the trustworthiness of web-based media and consider the data flow that permits targeted advertisements. Students employ appropriate tools to explore the hierarchical nature of DNS and IP. Students identify ways that a web developer's decisions affect the user and ways that the user's decisions impact society. The tree structure of web documents is introduced alongside HTML and CSS. Paired key encryption and authentication are introduced with an unplugged activity.

Lesson 2.2: Shopping and Social on the Web

The goal for this lesson is for students to understand the role of client-side code, server-side code, and databases in delivering interactive web content. The hook is a problem in which CS students collaborate with art students to publish content on the Web. Students are provided with JavaScript and PHP code and can access an SQL database from a secure shell command line, as well as through PHP. Students compare languages encountered so far to generalize the concepts of sequencing instructions, selection of instructions by conditionals, iteration, and the common roles of variables. Students explore and compare career paths within computing.

Lesson 2.3: Security and Cryptography

The goal of this lesson is for students to personally invest in maintaining online security and to improve their personal cyber security hygiene. Students focus on cyber security from the perspectives of the user, the software developer, the business, the nation, and the citizen. In the team competition at the end of the lesson, students explore parallel strands in encryption and security. Encryption is used as a route to explore the efficiency of algorithms and how the time for an algorithm to execute can be dependent on its input.

Unit 3: Raining Reigning Data

The goal of Unit 3 is for students to see the availability of large-scale data collection and analysis in every area they can imagine. Students examine very large data sets tied to themselves as well as to areas of work and society. They learn a variety of data visualization techniques and work to recognize opportunities to apply algorithmic thinking and automation when considering questions that have answers embedded in data. The complexity of the data sets, visualizations, and analysis increases in the second lesson of the unit, challenging students to generalize concepts developed in the first lesson.

Summary of Key Assignments, Labs and/or Activities

Lesson 3.1: Visualizing Data

The goal of this lesson is for students to be able to create visualizations to analyze sets of large data and to meaningfully interpret the patterns they uncover. They draw conclusions about themselves from relevant data, including local weather, the economics of their community, and naming trends

with their name. At the beginning of the lesson, students weigh societal concerns around the collection and persistence of Big Data. The students learn how to use Python to make useful graphic representations of data, developing from familiar visualizations to more modern visual analyses like scaled-dot or colorized scatter plots of multidimensional data sets. Students are introduced to basic Excel® spreadsheet programming and cell manipulation. A Monte Carlo simulation is used to help students appreciate the meaning of evidence for association between two variables.

Lesson 3.2: Discovering Knowledge from Data
As in the previous lesson, the goal of this lesson is for students to be able to create a range of visualizations to analyze complex sets of large data and to meaningfully interpret the patterns they uncover. Students use statistics to deepen the meaning of knowledge gained by visualization. The hooks are again conclusions they can draw about themselves from relevant data, including various geographic perspectives on their life and facial recognition of their own features. The lesson uses Excel, as well as Python, to manipulate and visualize data. Students examine multidimensional data sets using scatter plot arrays and view geographic and social data using heat maps and directed graphs. Students experiment with object recognition and facial recognition. They are challenged to discover clustering and linear correlation patterns lurking in data sets distributed across student computers and school sites, such that data cleaning and warehousing are necessary. Finally, student teams choose a question and answer it using large data.

Unit 4: Intelligent Behavior

In Unit 4 the emergence of intelligent behavior is explored from two distinct approaches: from human crowd sourcing of data and from separate algorithmic agents working in parallel. The goal is to galvanize the connections among computing concepts and between computing and society. The first lesson explores the hardware layer of computing, working from discrete components to integrated circuits. The exponential advancement of electronics, low on the ladder of abstraction, is connected to advancements at the highest levels on the ladder of abstraction, where artificial intelligence and simulation and modeling are impacting all fields. In the concluding lesson, students identify problems and questions that can be addressed with computer simulation, incorporating agent-based modeling. Students are challenged to explore the assumptions and parameters built into several simulations and to attach meaning to the results. Having explored a few applications of intelligent behavior emerging from algorithmic components, students reflect on the current and future state of artificial intelligence.

Summary of Key Assignments, Labs and/or Activities
Lesson 4.1: Moore's Law and Modeling
In this lesson, students construct an understanding of how the explosion of technology over the last two decades has impacted every realm of study and employment. Students begin by researching the impact of computer modeling and simulation which have been made possible by the rapid increase in computational power due to the continued applicability of Moore's Law. They then manipulate discrete electronic components to create logic gates and create comparable results using integrated circuits to get a feel for what it means to double the number of transistors that can fit in a given area. Students explore simulation in NetLogo directly by manipulating a model of predation and a model of the spread of viruses in humans. The lesson concludes with an examination of the code of ethics for simulationists and reflection on the necessity of adhering to such a code.

Lesson 4.2 Intelligent Agents

In this lesson, students experiment with materials designed to illuminate the rise of intelligent and complex behavior from simple rules and seemingly unintelligent agents. Students begin by studying a model of Langton's ant, a simple Turing machine with some surprising emergent behavior. The students manipulate models of neurons and neural networks. Students design and conduct their own experiments on a model of their own choosing using Monte Carlo methods. Students explore the generation and observation of fractals and study a diffusion limited aggregation model for producing fractal behavior. In the final project of the course, students choose a tool or tools that they have learned about in the course and apply their knowledge to create a novel product of their own design. They present their product to their class along with reflections about how it is tied to everything they've learned about computer science.

California Career Technical Education Model Curriculum Standards

Information and Communications Technology Anchor Standards

1.0 Academics
Analyze and apply appropriate academic standards required for successful industry sector pathway completion leading to postsecondary education and employment. Refer to the Information and Communication Technologies academic alignment matrix for identification of standards.

2.0 Communications
Acquire and accurately use Information and Communication Technologies sector terminology and protocols at the career and college readiness level for communicating effectively in oral, written, and multimedia formats. (Direct alignment with LS 9-10, 11-12.6)

2.1 Recognize the elements of communication using a sender–receiver model.
2.2 Identify barriers to accurate and appropriate communication.
2.3 Interpret verbal and nonverbal communications and respond appropriately.
2.4 Demonstrate elements of written and electronic communication such as accurate spelling, grammar, and format.
2.5 Communicate information and ideas effectively to multiple audiences using a variety of media and formats.
2.6 Advocate and practice safe, legal, and responsible use of digital media information and communications technologies.
2.7 Use technical writing and communication skills to work effectively with diverse groups of people.
2.8 Understand the principles of a customer-oriented service approach to users.

3.0 Career Planning and Management
Integrate multiple sources of career information from diverse formats to make informed career decisions, solve problems, and manage personal career plans. (Direct alignment with SLS 11-12.2)

3.1 Identify personal interests, aptitudes, information, and skills necessary for informed career decision making.
3.2 Evaluate personal character traits such as trust, respect, and responsibility and understand the impact they can have on career success.
3.3 Explore how information and communication technologies are used in career planning and decision making.

3.4 Research the scope of career opportunities available and the requirements for education, training, certification, and licensure.

3.5 Integrate changing employment trends, societal needs, and economic conditions into career planning.

3.6 Recognize the role and function of professional organizations, industry associations, and organized labor in a productive society.

3.7 Recognize the importance of small business in the California and global economies.

3.8 Understand how digital media are used by potential employers and postsecondary agencies to evaluate candidates.

3.9 Develop a career plan that reflects career interests, pathways, and postsecondary options.

4.0 Technology

Use existing and emerging technology, to investigate, research, and produce products and services, including new information, as required in the Information and Communication Technologies sector workplace environment. (Direct alignment with WS 11-12.6)

4.1 Use electronic reference materials to gather information and produce products and services.

4.2 Employ technology based communications responsibly and effectively to explore complex systems and issues.

4.3 Use information and communication technologies to synthesize, summarize, compare, and contrast information from multiple sources.

4.4 Discern the quality and value of information collected using digital technologies, and recognize bias and intent of the associated sources.

4.5 Research past, present, and projected technological advances as they impact a particular pathway.

4.6 Assess the value of various information and communication technologies to interact with constituent populations as part of a search of the current literature or in relation to the information task.

5.0 Problem Solving and Critical Thinking

Conduct short, as well as more sustained, research to create alternative solutions to answer a question or solve a problem unique to the Information and Communication Technologies sector using critical and creative thinking, logical reasoning, analysis, inquiry, and problem-solving techniques. (Direct alignment with WS 11-12.7)

5.1 Identify and ask significant questions that clarify various points of view to solve problems. 5.2 Solve predictable and unpredictable work-related problems using various types of reasoning (inductive, deductive) as appropriate.

5.3 Use systems thinking to analyze how various components interact with each other to produce outcomes in a complex work environment.

5.4 Interpret information and draw conclusions, based on the best analysis, to make informed decisions.

5.5 Use a logical and structured approach to isolate and identify the source of problems and to resolve problems.

5.6 Know the available resources for identifying and resolving problems.

5.7 Work out problems iteratively and recursively.

5.8 Create and use algorithms and solve problems. 5.9 Deconstruct large problems into components to solve. 5.10 Use multiple layers of abstraction. 3 ICT | California Career Technical Education Model Curriculum Standards 5.11 Understand the concept of base

systems, including binary and hexadecimal. 5.12 Apply the concepts of Boolean logic to decision making and searching.

6.0 Health and Safety

Demonstrate health and safety procedures, regulations, and personal health practices and determine the meaning of symbols, key terms, and domain-specific words and phrases as related to the Information and Communication Technologies sector workplace environment. (Direct alignment with RSTS 9-10, 11-12.4)

6.1 Locate, and adhere to, Material Safety Data Sheet (MSDS) instructions.
6.2 Interpret policies, procedures, and regulations for the workplace environment, including employer and employee responsibilities.
6.3 Use health and safety practices for storing, cleaning, and maintaining tools, equipment, and supplies.
6.4 Practice personal safety when lifting, bending, or moving equipment and supplies.
6.5 Demonstrate how to prevent and respond to work-related accidents or injuries; this includes demonstrating an understanding of ergonomics.
6.6 Maintain a safe and healthful working environment.
6.7 Be informed of laws/acts pertaining to the Occupational Safety and Health Administration (OSHA).
6.8 Maintain a safe and healthful working environment.
6.9 Dispose of e-waste properly, understanding the health, environmental, and legal risks of improper disposal.
6.10 Act conscientiously regarding the use of natural resources (e.g., paper, ink, etc.)
6.11 Conserve energy while computing (e.g., turn off equipment at night, power-saving settings, etc.)

7.0 Responsibility and Flexibility

Initiate, and participate in, a range of collaborations demonstrating behaviors that reflect personal and professional responsibility, flexibility, and respect in the Information and Communication Technologies sector workplace environment and community settings. (Direct alignment with SLS 9-10, 11-12.1)

7.1 Recognize how financial management impacts the economy, workforce, and community.
7.2 Explain the importance of accountability and responsibility in fulfilling personal, community, and workplace roles.
7.3 Understand the need to adapt to changing and varied roles and responsibilities.
7.4 Practice time management and efficiency to fulfill responsibilities.
7.5 Apply high-quality techniques to product or presentation design and development.
7.6 Demonstrate knowledge and practice of responsible financial management.
7.7 Demonstrate the qualities and behaviors that constitute a positive and professional work demeanor, including appropriate attire for the profession.
7.8 Explore issues of global significance and document the impact on the Information and Communication Technologies sector.

8.0 Ethics and Legal Responsibilities

Practice professional, ethical, and legal behavior, responding thoughtfully to diverse perspectives and resolving contradictions when possible, consistent with applicable laws, regulations, and organizational norms. (Direct alignment with SLS 11-12.1d)

8.1 Access, analyze, and implement quality assurance standards of practice.
8.2 Identify local, district, state, and federal regulatory agencies, entities, laws, and regulations related to the Information and Communication Technologies industry sector.

8.3 Demonstrate ethical and legal practices consistent with Information and Communication Technologies sector workplace standards.
8.4 Explain the importance of personal integrity, confidentiality, and ethical behavior in the workplace.
8.5 Analyze organizational culture and practices within the workplace environment.
8.6 Adhere to copyright and intellectual property laws and regulations, and use and appropriately cite proprietary information.
8.7 Conform to rules and regulations regarding sharing of confidential information, as determined by Information and Communication Technologies sector laws and practices.
8.8 Identify legal and ethical issues that have proliferated with increased technology adoption, including hacking, scamming, and breach of privacy.

9.0 Leadership and Teamwork

Work with peers to promote divergent and creative perspectives, effective leadership, group dynamics, team and individual decision making, benefits of workforce diversity, and conflict resolution such as those practiced in the Future Business Leaders of America and SkillsUSA career technical student organization. (Direct alignment with SLS 11-12.1b)

9.1 Define leadership and identify the responsibilities, competencies, and behaviors of successful leaders.
9.2 Identify the characteristics of successful teams, including leadership, cooperation, collaboration, and effective decision-making skills as applied in groups, teams, and career technical student organization activities.
9.3 Understand the characteristics and benefits of teamwork, leadership, and citizenship in the school, community, and workplace setting.
9.4 Explain how professional associations and organizations and associated leadership development and competitive career development activities enhance academic preparation, promote career choices, and contribute to employment opportunities.
9.5 Understand that the modern world is an international community and requires an expanded global view. 5 ICT | California Career Technical Education Model Curriculum Standards
9.6 Respect individual and cultural differences and recognize the importance of diversity in the workplace.
9.7 Participate in interactive teamwork to solve real Information and Communication Technologies sector issues and problems.

10.0 Technical Knowledge and Skills

Apply essential technical knowledge and skills common to all pathways in the Information and Communication Technologies sector, following procedures when carrying out experiments or performing technical tasks. (Direct alignment with WS 11-12.6)

10.1 Interpret and explain terminology and practices specific to the Information and Communication Technologies sector.
10.2 Comply with the rules, regulations, and expectations of all aspects of the Information and Communication Technologies sector.
10.3 Construct projects and products specific to the Information and Communication Technologies sector requirements and expectations.
10.4 Collaborate with industry experts for specific technical knowledge and skills.
10.5 Understand the major software and hardware components of a computer and a network and how they relate to each other.
10.6 Understand data sizes of various types of information (text, pictures, sound, video, etc.) and data capacity of various forms of media.

10.7 Understand the SI (metric) prefixes commonly used in computing including, at least, kilo, mega, giga, and tera.
10.8 Understand security concepts including authorization, rights, and encryption.
10.9 Use common industry-standard software and their applications including word processing, spreadsheets, databases, and multimedia software.
10.10 Manage files in a hierarchical system.
10.11 Know multiple ways in which to transfer information and resources (e.g., text, data, sound, video, still images) between software programs and systems.
10.12 Know appropriate search procedures for different types of information, sources, and queries.
10.13 Evaluate the accuracy, relevance, and comprehensiveness of retrieved information.
10.14 Analyze the effectiveness of online information resources to support collaborative tasks, research, publications, communications, and increased productivity.

11.0 Demonstration and Application
Demonstrate and apply the knowledge and skills contained in the Information and Communication Technologies anchor standards, pathway standards, and performance indicators in classroom, laboratory, and workplace settings, and through career technical student organizations such as Future Business Leaders of America and SkillsUSA.

11.1 Utilize work-based/workplace learning experiences to demonstrate and expand upon knowledge and skills gained during classroom instruction and laboratory practices specific to the Information and Communication Technologies sector program of study.
11.2 Demonstrate proficiency in a career technical pathway that leads to certification, licensure, and/or continued learning at the postsecondary level.
11.3 Demonstrate entrepreneurship skills and knowledge of self-employment options and innovative ventures.
11.4 Employ entrepreneurial practices and behaviors appropriate to Information and Communication Technologies sector opportunities.
11.5 Create a portfolio, or similar collection of work, that offers evidence through assessment and evaluation of skills and knowledge competency as contained in the anchor standards, pathway standards, and performance indicators.

California Career Technical Education Model Curriculum Standards

C. Software and Systems Development Pathway

Students in the Software and Systems Development pathway prepare for careers related to computer science that involve the design, development, implementation, maintenance, and management of systems that rely on software programs to satisfy the operational needs of modern business organizations. Persons with expertise in systems development and programming are critical to support operations like electronic commerce, medical records management, retail sales and inventory management, digital entertainment, and use of energy. Sample occupations associated with this pathway: Computer Programmer Software Developer/Applications Information Security Analyst Web Developer E-Business/E-Commerce Specialist

C1.0 Identify and apply the systems development process.

C1.1 Identify the phases of the systems development life cycle, including analysis, design, programming, testing, implementation, maintenance, and improvement.
C1.2 Identify and describe models of systems development, systems development life cycle (SDLC), and agile computing.

C1.3 Identify and describe how specifications and requirements are developed for new and existing software applications.
C1.4 Work as a member of, and within the scope and boundaries of, a development project team.
C1.5 Track development project milestones using the concept of versions.
C1.6 Diagram processes using flowcharts and the Unified Modeling Language.

C2.0 Define and analyze systems and software requirements.

C2.1 Describe the major purposes and benefits of development, including automation, improving productivity, modeling and analysis, and entertainment.
C2.2 Recognize and prevent unintended consequences of development work: programming errors, security issues, health and environmental risks, and privacy concerns.
C2.3 Develop strategies that target the specific needs and desires of the customer.
C2.4 Analyze customers' needs for development.
C2.5 Determine and document the requirements and alternative solutions to fulfill the customers' needs.

C3.0 Create effective interfaces between humans and technology.

C3.1 Describe and apply the basic process of input, processing, and output.
C3.2 Design effective and intuitive interfaces using knowledge of cognitive, physical, and social interactions.
C3.3 Support methods of accessibility for all potential users, including users with disabilities and non-English-speaking users.

C4.0 Develop software using programming languages.

C4.1 Identify and describe the abstraction level of programming languages from low-level, hardware-based languages to high-level, interpreted, Web-based languages.
C4.2 Describe the interaction and integration of programming languages and protocols such as how client-side programming can work with server-side programming to use a query language to access a database.
C4.3 Identify and use different authoring tools and integrated development environments (IDEs).
C4.4 Identify and apply data types and encoding.
C4.5 Demonstrate awareness of various programming paradigms, including procedural, object oriented, event-driven, and multithreaded programming.
C4.6 Use proper programming language syntax.
C4.7 Use various data structures, arrays, objects, files, and databases.
C4.8 Use object oriented programming concepts, properties, methods, and inheritance.
C4.9 Create programs using control structures, procedures, functions, parameters, variables, error recovery, and recursion.
C4.10 Create and know the comparative advantages of various queue, sorting, and searching algorithms.
C4.11 Document development work for various audiences, such as comments for other programmers, and manuals for users.

C5.0 Test, debug, and improve software development work.

C5.1 Identify the characteristics of reliable, effective, and efficient products.
C5.2 Describe the ways in which specification changes and technological advances can require the modification of programs.
C5.3 Use strategies to optimize code for improved performance.

C5.4 Test software and projects.
C5.5 Evaluate results against initial requirements.
C5.6 Debug software as part of the quality assurance process.

C6.0 Integrate a variety of media into development projects.

C6.1 Identify the basic design elements necessary to produce effective print, video, audio, and interactive media.
C6.2 Describe the various encoding methods of media and trade-offs: vector graphics vs. bitmaps, and bit depth.
C6.3 Use media design and editing software: keyframe animation, drawing software, image editors, and three-dimensional design.
C6.4 Develop a presentation or other multimedia project: video, game, or interactive Web sites, from storyboard to production.
C6.5 Analyze the use of media to determine the appropriate file format and level of compression.
C6.6 Integrate media into a full project using appropriate tools.
C6.7 Create and/or capture professional-quality media, images, documents, audio, and video clips.

C7.0 Develop Web and online projects.

C7.1 Identify the hardware (server) and software required for Web hosting and other services.
C7.2 Describe the full process of online content delivery, registering domain names, setting up hosting, and setting up email addresses.
C7.3 Attract Web-site visitors through search engine optimization using various strategies like keywords and meta-tags.
C7.4 Enable e-commerce capabilities to sell products, create a shopping cart, and handle credit card transactions.
C7.5 Create an online project, Web-based business, and e-portfolio.
C7.6 Optimize fast delivery and retrieval of online content such as Web pages.

C8.0 Develop databases.

C8.1 Describe the critical function of databases in modern organizations.
C8.2 Identify and use the basic structures of databases, fields, records, tables, and views.
C8.3 Identify and explain the types of relationships between tables (one-to-one, one-to-many, many-to-many) and use methods to establish these relationships, including primary keys, foreign keys, and indexes. C8.4 Use data modeling techniques to create databases based upon business needs.
C8.5 Use queries to extract and manipulate data (select queries, action queries).
C8.6 Develop databases that are properly normalized using appropriate schemas.
C8.7 Export and import data to and from other applications and a database recognizing the limitations and challenges inherent in the process.
C8.8 Analyze and display data to assist with decision making using methods like cross tabulations, graphs, and charts.

C9.0 Develop software for a variety of devices, including robotics.

C9.1 Demonstrate awareness of the applications of device development work, including personalized computing, robotics, and smart appliances.
C9.2 Install equipment, assemble hardware, and perform tests using appropriate tools and technology.

C9.3 Use hardware to gain input, process information, and take action.
C9.4 Apply the concepts of embedded programming, including digital logic, machine-level representation of data, and memory-system organization.
C9.5 Program a microcontroller for a device or robot.

C10.0 Develop intelligent computing.

C10.1 Describe models of intelligent behavior and what distinguishes humans from machines. C10.2 Describe the major areas of intelligent computing, including perception, proximity, processing, and control.
C10.3 Know artificial intelligence methods such as neural networks, Bayesian inferences, fuzzy logic, and finite state machines.
C10.4 Implement artificial intelligent behavior through various methods: mathematical modeling, reinforcement learning, and probabilistic analysis.

Computer Science Teachers Association (CSTA) K-12 Computer Science Standards

Computational Thinking (CT)

The student will be able to:

1. Classify problems as tractable, intractable, or computationally unsolvable.
2. Explain the value of heuristic algorithms to approximate solutions for intractable problems.
3. Critically examine classical algorithms and implement an original algorithm.
4. Evaluate algorithms by their efficiency, correctness, and clarity.
5. Use data analysis to enhance understanding of complex natural and human systems.
6. Compare and contrast simple data structures and their uses (e.g., arrays and lists).
7. Discuss the interpretation of binary sequences in a variety of forms (e.g., instructions, numbers, text, sound, image).
8. Use models and simulations to help formulate, refine, and test scientific hypotheses.
9. Analyze data and identify patterns through modeling and simulation.
10. Decompose a problem by defining new functions and classes.

Collaboration (CL)

The student will be able to:

1. Use project collaboration tools, version control systems, and Integrated Development Environments (IDEs) while working on a collaborative software project.
2. Demonstrate the software life cycle process by participating on a software project team.
3. Evaluate programs written by others for readability and usability.

Computing Practice and Programming (CPP)

The student will be able to:

1. Use advanced tools to create digital artifacts (e.g., web design, animation, video, multimedia).
2. Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).
3. Classify programming languages based on their level and application domain
4. Explore principles of system design in scaling, efficiency, and security.
5. Deploy principles of security by implementing encryption and authentication strategies.
6. Anticipate future careers and the technologies that will exist.
7. Use data analysis to enhance understanding of complex natural and human systems.

8. Deploy various data collection techniques for different types of problems.

Community, Global, and Ethical Impacts (CI)

The student will be able to:

1. Demonstrate ethical use of modern communication media and devices.
2. Analyze the beneficial and harmful effects of computing innovations.
3. Summarize how financial markets, transactions, and predictions have been transformed by automation.
4. Summarize how computation has revolutionized the way people build real and virtual organizations and infrastructures.
5. Identify laws and regulations that impact the development and use of software.
6. Analyze the impact of government regulation on privacy and security.
7. Differentiate among open source, freeware, and proprietary software licenses and their applicability to different types of software.
8. Relate issues of equity, access, and power to the distribution of computing resources in a global society.

Instructional Methods and/or Strategies

The instructional strategy is three-fold: Instruct the students in essential computational thinking skills by giving them an activity to accomplish; practice newly learned skills with one or more projects; and then apply the skills and project experience to solve a real-world, often student designed problem. This activity-project-problem based methodology inspires students to learn according to their interests.

Assessment Methods and/or Tools

During activities, students will be assessed in a formative way; students will answer questions about the activity and specific skill they are learning. For the project-based assessment, students' projects will be assessed according to a rubric or statement of purpose. Finally, students will be assessed on how well their materials solved the given problem, how well they demonstrated the skills learned, and how they analyzed any social, cultural, legal, etc. impacts.

Throughout the course, students will also be assessed on their presentation skills, their professionalism during class and the collaboration efforts with their peers.

Honors Courses

In this course, students will be expected to not only demonstrate content knowledge through computational tasks but will also be required to design and implement large scale projects individually and in collaborative small groups. Students will complete a Create Performance Task in which they will spend at least 12 hours developing their own computational program that involves nested algorithms and the use of programming abstractions. In this course, students develop core skills for success at university, including applied critical thinking, inquiry, and problem-solving skills and time management and organizational skills necessary to complete the Computer Science unit projects.