Washington Office of Superintendent of
PUBLIC INSTRUCTION

Learning that works for Washington
CTE

Statewide Framework Document for:

110201 AP Computer Science Principles

Standards may be added to this document prior to submission but may not be removed from the framework to meet state credit equivalency requirements. Performance assessments may be developed at the local level. In order to earn state approval, performance assessments must be submitted within this framework. **This course is eligible for the 3rd credit of Math.** Washington Mathematics Standards (Common Core State Standards) support foundational mathematical knowledge and reasoning. While it is important to develop a conceptual understanding of mathematical topics and fluency in numeracy and procedural skills, teachers should also focus on the application of mathematics to career fields to support the three (3) key shifts of CCSS. The Standards for Mathematical Practice develop mathematical habits of mind and are to be modeled and integrated throughout the course.

School District Name	
Course Title: AP Computer Science Principles	**Total Framework Hours:** 180
CIP Code: 110201 ☐ Exploratory ☒ Preparatory	**Date Last Modified:** March 12, 2021
Career Cluster: Information Technology	**Cluster Pathway:** Programming and Software Development
Course Summary: In this course, students will develop computational thinking skills vital for success across all disciplines, such as using computational tools to analyze and study and working with large data sets to analyze, visualize, 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.	
Eligible for Equivalent Credit in: 3rd credit of math	**Total Number of Units:** 6
Course Resources: https://apcentral.collegeboard.org/courses/ap-computer-science-principles/course https://apcentral.collegeboard.org/courses/ap-computer-science-principles/course/2020-21-updates?course=ap-computer-science-principles https://apcentral.collegeboard.org/courses/ap-computer-science-principles/classroom-resources/curricula-pedagogical-support?course=ap-computer-science-principles https://apcentral.collegeboard.org/courses/ap-computer-science-principles/classroom-resources?course=ap-computer-science-principles	

Unit 1: Creative Development	**Total Learning Hours for Unit:** 35

Unit Summary: When developing computing innovations, developers can use a formal, iterative design process or experimentation. While using either approach, developers will encounter phases of investigating and reflecting, designing, prototyping, and testing. Additionally, collaboration is an important tool to use at any phase of development because considering multiple perspectives allows for improvement of innovations.

AP Computer Science Principles
https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-conceptual-framework-2020-21.pdf
Enduring Understanding: Incorporating multiple perspectives through collaboration improves computing innovations as they are developed. Developers create and innovate using an iterative design process that is user focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.

1.1 Collaboration

- 1.C Explain how collaboration affects the development of a solution.
- 6.A A Collaborate in the development of solutions (not assessed).

1.2 Program Function and Purpose

- 1.A Investigate the situation, context, or task.
- 3.A Generalize data sources through variables.
- 4.A Explain how a code segment or program functions.

1.3 Program Design and Development

- 1.B Determine and design an appropriate method or approach to achieve the purpose.
- 1.C Explain how collaboration affects the development of a solution.
- 4.A Explain how a code segment or program functions.
- 6.C Acknowledge the intellectual property of others (not assessed).

1.4 Identifying and Correcting Errors

- 1.B Determine and design an appropriate method or approach to achieve the purpose.
- 4.C Identify and correct errors in algorithms and programs, including error discovery through testing.

Performance Assessments: (Districts to complete for each unit)
Example assessments for this unit include:

- Sharing and responding: Students develop a list of three questions that they would like to use data to answer. Then, in small groups, ask each student to share one of their questions. The group will respond with feedback to improve the focus and direction of the question. Students should take turns sharing their questions until all questions have been considered. Finally, ask each group to come to a consensus on which three questions they will answer with data.
- Diagramming: In small groups, students play a board game for 10 minutes. As they play, ask them to record the actions (such as rolling the dice or moving their piece) and decisions made in a diagram or flowchart. Have students trade games with another group and play the game using the diagram for directions. Students should identify and correct where the diagram might not be accurate or have missing steps. See the Language and Logic of Computing: Algorithmic Thinking Teaching and Assessing Module in the Professional Learning section of AP Classroom for a more detailed lesson plan and video example.

- Student culminates their own website about themselves. (example from CodeHS)

Leadership Alignment: (Districts to complete for each unit)
Leadership alignment must include a unit specific project/activity that aligns with the 21st Century Leadership Skills.
Example:

- Creativity and Innovation demonstrated when students create their own websites.
- Students demonstrate Initiative and Self-Direction when they work alone to write an individual program, answer questions about the individual program and about the collaboration (peer review) with their partner.
- Students demonstrate Information Literacy when they Access and Evaluate Information and Use and Manage Information to explore technical challenges and questions that arise from the need to represent digital information in computers and transfer it between people and computational devices.

Core Leadership Skills- Group Skills Standards
2.1 The student will communicate, participate, and advocate effectively in pairs, small groups, teams, and large groups in order to reach common goals.
2.7 The student will demonstrate the ability to train others to understand the established rules and expectations, rationale and consequences and to follow those rules and expectations.
2.8 The student demonstrates the ability to incorporate and utilize the principles of group dynamics in a variety of settings.

Industry Standards and/or Competencies:
National Business Education Association (**NBEA**) www.nbea.org
PROGRAMMING AND APPLICATION DEVELOPMENT
Level 3-4 Performance Expectations

- Apply design principles to programming tasks
- Test, debug, and document code

PROJECT MANAGEMENT AND SYSTEMS ANALYSIS
Level 1-2 Performance Expectations

- Build timelines for projects
- Apply project management concepts for collaborative works projects

Aligned Washington State Academic Standards	
Computer Science	3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. (P. 5.2) 3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible. (P. 6.3) 3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools. (P. 2.4) 3A-IC-25 Test and refine computational artifacts to reduce bias and equity deficits. (P. 1.2) 3A-IC-27 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. (P. 2.4)

Computer Science Practices	CS_P_1 Fostering an Inclusive Computing Culture CS_P_1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility. CS_P_2 Collaborating Around Computing CS_P_2.4 Evaluate and select technological tools that can be used to collaborate on a project. CS_P_5 Creating Computational Artifacts CS_P_5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue. CS_P_6 Testing and Refining Computational Artifacts CS_P_6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.
Mathematics: Common Core	HS.N.Q.1 Use units as a way to understand problems and to guide the solution of multi-step problems; choose and interpret units consistently in formulas; choose and interpret the scale and the origin in graphs. and data displays. HS.N.Q.2 Define appropriate quantities for the purpose of descriptive modeling. HS.N.Q.3 Choose a level of accuracy appropriate to limitations on measurement when reporting quantities. HS.A.SSE.1 Interpret expressions that represent a quantity in terms of its context.* HS.A.SSE.1a Interpret parts of an expression, such as terms, factors, and coefficients. HS.A.SSE.1b Interpret complicated expressions by viewing one or more of their parts as a single entity. HS.A.SSE.2 Use the structure of an expression to identify ways to rewrite it. HS.A.SSE.3c Use the properties of exponents to transform expressions for exponential functions. HS.A.APR.6 Rewrite simple rational expressions in different forms; write a(x)/b(x) in the form q(x) + r(x)/b(x), where a(x), b(x), q(x), and r(x) are polynomials with the degree of r(x) less than the degree of b(x), using inspection, long division, or, for the more complicated examples, a computer algebra system. HS.A.CED.1 Create equations and inequalities in one variable and use them to solve problems. Include equations arising from linear and quadratic functions, and simple rational and exponential functions. HS.A.CED.2 Create equations in two or more variables to represent relationships between quantities; graph equations on coordinate axes with labels and scales. HS.A.CED.3 Represent constraints by equations or inequalities, and by systems of equations and/or inequalities, and interpret solutions as viable or nonviable options in a modeling context. HS.A.REI.2 Solve simple rational and radical equations in one variable, and give examples showing how extraneous solutions may arise. HS.A.REI.3 Solve linear equations and inequalities in one variable, including equations with coefficients represented by letters. HS.F.IF.2 Use function notation, evaluate functions for inputs in their domains, and interpret statements that use function notation in terms of a context.

	HS.F.BF.1 Write a function that describes a relationship between two quantities.* HS.F.BF.1a Determine an explicit expression, a recursive process, or steps for calculation from a context. HS.S.CP.2 Understand that two events A and B are independent if the probability of A and B occurring together is the product of their probabilities and use this characterization to determine if they are independent.
Mathematical Practices	MP1 Make sense of problems and persevere in solving them. MP2 Reason abstractly and quantitatively. MP4 Model with mathematics. MP5 Use appropriate tools strategically. MP6 Attend to precision. MP7 Look for and make use of structure.

Unit 2: Data	**Total Learning Hours for Unit:** 35

Unit Summary: Data are central to computing innovations because they communicate initial conditions to programs and represent new knowledge. Computers consume data, transform data, and produce new data, allowing users to create new information or knowledge to solve problems through the interpretation of these data. Computers store data digitally, which means that the data must be manipulated on input and output in order to be presented in a useful way to the user.

AP Computer Science Principles
https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-conceptual-framework-2020-21.pdf
Enduring Understanding: The way a computer represents data internally is different from the way the data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people. Programs can be used to process data, which allows users to discover information and create new knowledge.

2.1 Binary Numbers
- 1.D Evaluate solution options.
- 2.B Implement and apply an algorithm.
- 3.C Explain how abstraction manages complexity.

2.2 Data Compression
- 1.D Evaluate solution options.

2.3 Extracting Information from Data
- 5.B Explain how knowledge can be generated from data.
- 5.D Describe the impact of gathering data.

2.4 Using Programs with Data
- 2.B Implement and apply an algorithm.

- 5.B Explain how knowledge can be generated from data.

Performance Assessments: (Districts to complete for each unit)

Example assessments for this unit include:

- Look for a pattern: Provide students with a sentence or paragraph of compressed lossless text and a key. Have them look for patterns in their process of retrieving the original text and evaluate whether this is the best compression algorithm to use. Have them write down the patterns they see along with their evaluation and share these in a large group.
- Diagramming: Give students a question and a list of data. Have them diagram a process that could be used to answer the question using the data, making sure to include the input(s) of information and the output of the transformed data. Have students include an explanation of how the process represented in their diagram would work to find the solution.
- Format Showdown (based on code.org Unit 02 Lesson 06) - students will conduct a small amount of research to explore a file format either currently in use or from history.
- Finding Trends with Visualizations -
- Data Innovation (based on code.org Unit 04 Lesson 04) - Students will research a topic of personal interest and respond to questions about how that innovation produces, uses, or consumes data.
- Divide by 2 (based on code.org Unit 05 Lesson 06) – Students use a for loop to process all numbers in an array, dividing each value by 2.

Leadership Alignment: (Districts to complete for each unit)

Leadership alignment must include a unit specific project/activity that aligns with the 21st Century Leadership Skills.

EXAMPLE:

- Creativity and Innovation demonstrated when students create their own websites.
- Students demonstrate Initiative and Self-Direction when they work alone to write an individual program, answer questions about the individual program and about the collaboration (peer review) with their partner.
- Students demonstrate Information Literacy when they Access and Evaluate Information and Use and Manage Information to explore technical challenges and questions that arise from the need to represent digital information in computers and transfer it between people and computational devices.

Core Leadership Skills- Group Skills Standards

2.1 The student will communicate, participate, and advocate effectively in pairs, small groups, teams, and large groups in order to reach common goals.

2.7 The student will demonstrate the ability to train others to understand the established rules and expectations, rationale and consequences and to follow those rules and expectations.

2.8 The student demonstrates the ability to incorporate and utilize the principles of group dynamics in a variety of settings.

Industry Standards and/or Competencies:

National Business Education Association (**NBEA**) www.nbea.org

DATABASE MANAGEMENT SYSTEMS

Level 2 Performance Expectations

- Describe search strategies and use them to solve common information problems

DATA AND NETWORKING INFRASTRUCTURES

Level 1 - 2 Performance Expectations

• Recognize the impact of the convergence of telephony, data, and video communications on networks SECURITY AND RISK MANAGEMENT Level 1-2 Performance Expectations • Discuss the risks of data loss and methods of prevention Level 4 Performance Expectations • Identify risks to personnel, facilities, data, communications systems, and applications	
Aligned Washington State Academic Standards	
Computer Science	3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. (P. 4.1) 3A-DA-09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. (P. 4.1) 3A-DA-10 Evaluate the tradeoffs in how data elements are organized and where data is stored. (P. 3.3) 3A-DA-11 Create interactive data visualizations using software tools to help others better understand real-world phenomena. (P. 4.4) 3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. (P. 4.4)
Computer Science Practices	CS_P_3 Recognizing and Defining Computational Problems CS_P_3.3 Evaluate whether it is appropriate and feasible to solve a problem computationally. CS_P_4 Developing and Using Abstractions CS_P_4.1 Extract common features from a set of interrelated processes or complex phenomena. CS_P_4.4 Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.
Mathematics: Common Core	HS.N.Q.1 Use units as a way to understand problems and to guide the solution of multi-step problems; choose and interpret units consistently in formulas; choose and interpret the scale and the origin in grapHS. and data displays. HS.N.Q.2 Define appropriate quantities for the purpose of descriptive modeling. HS.N.Q.3 Choose a level of accuracy appropriate to limitations on measurement when reporting quantities. HS.N.VM.1 (+) Recognize vector quantities as having both magnitude and direction. Represent vector quantities by directed line segments, and use appropriate symbols for vectors and their magnitudes (e.g., v, |v|, ||v||, v). HS.N.VM.6 (+) Use matrices to represent and manipulate data, e.g., to represent payoffs or incidence relationships in a network. HS.A.CED.1 Create equations and inequalities in one variable and use them to solve problems. Include equations arising from linear and quadratic functions, and simple rational and exponential functions. HS.A.CED.2 Create equations in two or more variables to represent relationships between quantities; graph equations on coordinate axes with labels and scales.

	HS.A.CED.3 Represent constraints by equations or inequalities, and by systems of equations and/or inequalities, and interpret solutions as viable or nonviable options in a modeling context. HS.A.CED.4 Rearrange formulas to highlight a quantity of interest, using the same reasoning as in solving equations. HS.A.REI.1 Explain each step in solving a simple equation as following from the equality of numbers asserted at the previous step, starting from the assumption that the original equation has a solution. Construct a viable argument to justify a solution method. HS.A.REI.2 Solve simple rational and radical equations in one variable, and give examples showing how extraneous solutions may arise. HS.A.REI.5 Prove that, given a system of two equations in two variables, replacing one equation by the sum of that equation and a multiple of the other produces a system with the same solutions. HS.F.IF.9 Compare properties of two functions each represented in a different way (algebraically, graphically, numerically in tables, or by verbal descriptions). HS.F.BF.1 Write a function that describes a relationship between two quantities.* HS.F.BF.1c (+) Compose functions. HS.F.LE.1 Distinguish between situations that can be modeled with linear functions and with exponential functions. HS.S.ID.1 Represent data with plots on the real number line (dot plots, histograms, and box plots). HS.S.ID.2 Use statistics appropriate to the shape of the data distribution to compare center (median, mean) and spread (interquartile range, standard deviation) of two or more different data sets. HS.S.ID.9 Distinguish between correlation and causation. HS.S.IC.1 Understand statistics as a process for making inferences about population parameters based on a random sample from that population. HS.S.IC.2 Decide if a specified model is consistent with results from a given data-generating process, e.g., using simulation. HS.S.IC.6 Evaluate reports based on data. HS.S.MD.7 (+) Analyze decisions and strategies using probability concepts (e.g., product testing, medical testing, pulling a hockey goalie at the end of a game).
Mathematical Practices	MP2 Reason abstractly and quantitatively. MP3 Construct viable arguments and critique the reasoning of others. MP4 Model with mathematics. MP7 Look for and make use of structure.

Unit 3: Algorithms and Programming	**Total Learning Hours for Unit:** 30

Unit Summary: All programming languages, whether block-based or text-based, use similar programming structures and commands. Having a basic understanding of how these building blocks are combined to form algorithms and abstractions in one language makes it easier to apply these same understandings to other programming languages. This big idea focuses on determining the efficiency of algorithms, as well as writing and implementing algorithms in a program. This big idea can be paired with any of the other big ideas and taught throughout the school year.

AP Computer Science Principles
https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-conceptual-framework-2020-21.pdf
Enduring Understanding: To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways. The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence. There exist problems that computers cannot solve, and even when a computer can solve a problem, it may not be able to do so in a reasonable amount of time.

3.1 Variables and Assignments
- 3.A Generalize data sources through variables.
- 4.B Determine the result of code segments.

3.2 Data Abstraction
- 3.A Generalize data sources through variables.
- 3.B Use abstraction to manage complexity in a program.
- 3.C Explain how abstraction manages complexity.

3.3 Mathematical Expressions
- 2.A Represent algorithmic processes without using a programming language.
- 2.B Implement and apply an algorithm.
- 4.B Determine the result of code segments.

3.4 Strings
- 4.B Determine the result of code segments.

3.5 Boolean Expressions
- 2.B Implement and apply an algorithm.
- 4.B Determine the result of code segments.

3.6 Conditionals
- 2.A Represent algorithmic processes without using a programming language.
- 2.B Implement and apply an algorithm.
- 4.B Determine the result of code segments.

3.7 Nested Conditionals
- 2.B Implement and apply an algorithm.

- 4.B Determine the result of code segments.

3.8 Iteration
- 2.A Represent algorithmic processes without using a programming language.
- 2.B Implement and apply an algorithm.
- 4.B Determine the result of code segments.

3.9 Developing Algorithms.
- 1.D Evaluate solution options.
- 2.A Represent algorithmic processes without using a programming language.
- 2.B Implement and apply an algorithm.

3.10 Lists
- 2.B Implement and apply an algorithm.
- 4.B Determine the result of code segments.

3.11 Binary Search
- 1.A Investigate the situation, context, or task.
- 1.D Evaluate solution options.

3.12 Calling Procedures.
- 3.B Use abstraction to manage complexity in a program.
- 4.B Determine the result of code segments.

3.13 Developing Procedures.
- 3.B Use abstraction to manage complexity in a program.
- 3.C Explain how abstraction manages complexity.

3.14 Libraries
- 2.B Implement and apply an algorithm.

3.15 Random Values
- 2.B Implement and apply an algorithm.
- 4.B Determine the result of code segments.

3.16 Simulations
- 1.A Investigate the situation, context, or task.
- 1.D Evaluate solution options.

3.17 Algorithmic Efficiency
- 1.D Evaluate solution options.

3.18 Undecidable Problems
- 1.A Investigate the situation, context, or task.

Performance Assessments: (Districts to complete for each unit)
Example assessments for this unit include:

- Predict and compare: Provide students with a list of expressions with assignments. Ask them to predict the value of each variable after the assignment and then compare their answers to the output produced when these statements are put into a program.
- Using manipulatives: When learning about conditionals, take a printout of a simple conditional statement and cut it into multiple sections. As students enter the classroom, hand them an envelope full of the paper strips, and ask them to reassemble the conditional in the proper order.
- Marking the text: Provide students with program code that draws a square of side length 10 and a separate set of program code that uses side length 100. Ask students to mark up the sets of code to identify where they are different and to create a generalization by using parameters. Ask them to write a procedure that uses parameters to draw a square of any size.
- Think-pair-share: Have students work in pairs to consider what factors would be the most important to prioritize in writing an algorithm to build the perfect master schedule for the school. Some considerations may include maximum class size, student preferences, and teacher availability. Have the pairs discuss and then report their results. Finally, discuss as a class how such programs may have to settle for a "good enough" solution when an exact solution may not be possible in a reasonable amount of time.
- Students evaluate pseudo code and be able to demonstrate their understanding of programming concepts such as the use of variables, mathematical expressions, strings, Boolean expressions, conditionals, iteration, lists, and random variables.

Leadership Alignment: (Districts to complete for each unit)
Leadership alignment must include a unit specific project/activity that aligns with the 21st Century Leadership Skills.
EXAMPLE:

- Creativity and Innovation demonstrated when students create their own websites.
- Students demonstrate Initiative and Self-Direction when they work alone to write an individual program, answer questions about the individual program and about the collaboration (peer review) with their partner.
- Students demonstrate Information Literacy when they Access and Evaluate Information and Use and Manage Information to explore technical challenges and questions that arise from the need to represent digital information in computers and transfer it between people and computational devices.

Core Leadership Skills- Group Skills Standards
2.1 The student will communicate, participate, and advocate effectively in pairs, small groups, teams, and large groups in order to reach common goals.
2.7 The student will demonstrate the ability to train others to understand the established rules and expectations, rationale and consequences and to follow those rules and expectations.
2.8 The student demonstrates the ability to incorporate and utilize the principles of group dynamics in a variety of settings.

Industry Standards and/or Competencies:
National Business Education Association (**NBEA**) www.nbea.org
PROGRAMMING AND APPLICATION DEVELOPMENT
Level 3-4 Performance Expectations

- Identify and explain programming structures
- Apply design principles to programming tasks
- Test, debug, and document code
- Maintain and reengineer existing code

Aligned Washington State Academic Standards	
Computer Science	3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. (P. 4.1) 3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. (P. 6.2) 3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. (P. 4.4) 3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. (P. 5.2) 3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. (P. 4.1) 3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. (P. 5.2) 3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. (P. 5.2) 3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. (P. 3.2) 3A-AP-18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. (P. 5.2) 3A-AP-19 Systematically design and develop programs for broad audiences by incorporating feedback from users. (P. 5.1) 3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible. (P. 6.3) 3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools. (P. 2.4) 3A-IC-25 Test and refine computational artifacts to reduce bias and equity deficits. (P. 1.2) 3A-IC-26 Demonstrate ways a given algorithm applies to problems across disciplines. (P. 3.1) 3A-IC-27 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. (P. 2.4)
Computer Science Practices	CS_P_1 Fostering an Inclusive Computing Culture CS_P_1.1 Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products. CS_P_1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability. CS_P_2 Collaborating Around Computing CS_P_2.1 Cultivate working relationships with individuals possessing diverse perspectives, skills, and personalities.

	CS_P_2.2 Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness. CS_P_2.3 Solicit and incorporate feedback from, and provide constructive feedback to team members and other stakeholders. CS_P_2.4 Evaluate and select technological tools that can be used to collaborate on a project. CS_P_3 Recognizing and Defining Computational Problems CS_P_3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally. CS_P_3.2 Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures. CS_P_4 Developing and Using Abstractions CS_P_4.1 Extract common features from a set of interrelated processes or complex phenomena. CS_P_4.3 Create modules and develop points of interaction that can apply to multiple situations and reduce complexity. CS_P_4.4 Model phenomena and processes and simulate systems to understand and evaluate potential outcomes. CS_P_5 Creating Computational Artifacts CS_P_5.1 Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations. CS_P_5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue. CS_P_6 Testing and Refining Computational Artifacts CS_P_6.2 Identify and fix errors using a systematic process. CS_P_6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.
Mathematics: Common Core	HS.N.VM.3 (+) Solve problems involving velocity and other quantities that can be represented by vectors. HS.N.VM.4 (+) Add and subtract vectors. HS.A.SSE.1b Interpret complicated expressions by viewing one or more of their parts as a single entity. HS.A.SSE.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression. * HS.A.APR.6 Rewrite simple rational expressions in different forms; write a(x)/b(x) in the form q(x) + r(x)/b(x), where a(x), b(x), q(x), and r(x) are polynomials with the degree of r(x) less than the degree of b(x), using inspection, long division, or, for the more complicated examples, a computer algebra system. HS.A.CED.1 Create equations and inequalities in one variable and use them to solve problems. Include equations arising from linear and quadratic functions, and simple rational and exponential functions.

	HS.A.REI.1 Explain each step in solving a simple equation as following from the equality of numbers asserted at the previous step, starting from the assumption that the original equation has a solution. Construct a viable argument to justify a solution method. HS.A.REI.2 Solve simple rational and radical equations in one variable, and give examples showing how extraneous solutions may arise. HS.F.IF.9 Compare properties of two functions each represented in a different way (algebraically, graphically, numerically in tables, or by verbal descriptions). HS.F.BF.1 Write a function that describes a relationship between two quantities. * HS.F.BF.1a Determine an explicit expression, a recursive process, or steps for calculation from a context. HS.F.BF.1b Combine standard function types using arithmetic operations. HS.F.BF.1c (+) Compose functions. HS.G.GPE.4 Use coordinates to prove simple geometric theorems algebraically.
Mathematical Practices	MP1 Make sense of problems and persevere in solving them. MP2 Reason abstractly and quantitatively. MP3 Construct viable arguments and critique the reasoning of others. MP4 Model with mathematics. MP5 Use appropriate tools strategically. MP7 Look for and make use of structure. MP8 Look for and express regularity in repeated reasoning.

Unit 4: Computing Systems and Networks	**Total Learning Hours for Unit:** 35

Unit Summary: Computer systems and networks are used to transfer data. One of the largest and most commonly used networks is the Internet. Through a series of protocols, the Internet can be used to send and receive information and ideas throughout the world. Transferring and processing information can be slow when done on a single computer but leveraging multiple computers to do the work at the same time can significantly shorten the time it takes to complete tasks or solve problems.

AP Computer Science Principles
https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-conceptual-framework-2020-21.pdf
Enduring Understanding: Computer systems and networks facilitate the transfer of data. Parallel and distributed computing leverage multiple computers to more quickly solve complex problems or process large data sets.
4.1 The Internet

- 5.A Explain how computing systems work.

4.2 Fault Tolerance

- 1.D Evaluate solution options.

- 5.A Explain how computing systems work.

4.3 Parallel and Distributed Computing

- 1.D Evaluate solution options.

Performance Assessments: (Districts to complete for each unit)

Example assessments for this unit include:

- Journaling: Ask students to read about the Internet and packet switching in Blown to Bits. Pose several prompts related to the Internet, such as the following, and have students add their answers to their journals:
  - How is the Internet like the US Post Office?
  - Explain the difference between circuit switching and packet switching.
  - Ask students to use what they learned from reading to make a drawing showing how they think an email travels from one place to another.
- Predict and compare: When introducing parallel and distributed computing, present students with a set of processes and several distributed models. Ask students to compare the models and predict which one is the most efficient, least efficient, or equivalent to other models in the set. Then show students how to determine the efficiency of each model to check if their predictions were correct.
- Students create a model of a computer network and visually explain the relationship between individual devices, the Internet and the World-Wide-Web, and how fault-tolerance prevents data corruption. (Code.org)
- Students invent a file type/protocol for encoding a complex type of information that has some personal significance. (Code.org)
- Students work together to invent solutions and protocols to many of the problems that arise. (Code.org)

Leadership Alignment: (Districts to complete for each unit)

Leadership alignment must include a unit specific project/activity that aligns with the 21st Century Leadership Skills.

EXAMPLE:

- Creativity and Innovation demonstrated when students create their own websites.
- Students demonstrate Initiative and Self-Direction when they work alone to write an individual program, answer questions about the individual program and about the collaboration (peer review) with their partner.
- Students demonstrate Information Literacy when they Access and Evaluate Information and Use and Manage Information to explore technical challenges and questions that arise from the need to represent digital information in computers and transfer it between people and computational devices.

Core Leadership Skills- Group Skills Standards

2.1 The student will communicate, participate, and advocate effectively in pairs, small groups, teams, and large groups in order to reach common goals.

2.7 The student will demonstrate the ability to train others to understand the established rules and expectations, rationale and consequences and to follow those rules and expectations.

2.8 The student demonstrates the ability to incorporate and utilize the principles of group dynamics in a variety of settings.

Industry Standards and/or Competencies:

National Business Education Association (**NBEA**) www.nbea.org

DATA AND NETWORKING-INFRASTRUCTURES

<table>
<tr><td colspan="2">Level 1 - 2 Performance Expectations
• Identify basic network connectivity concepts
• Apply basic networking terminology to a network environment
Level 3 Performance Expectations
• Identify network connectivity hardware and related software
END-USER SUPPORT AND TRAINING
Level 1-2 Performance Expectations
• Work in a team to solve problems and share knowledge
• Develop critical thinking skills to locate resources to solve problems</td></tr>
<tr><td colspan="2">Aligned Washington State Academic Standards</td></tr>
<tr><td>Computer Science</td><td>3A-CS-01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. (P. 4.1)
3A-CS-02 Compare levels of abstraction and interactions between application software, system software, and hardware layers. (P. 4.1)
3A-NI-04 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. (P. 4.1)
3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system. (6.3)
3A-DA-09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. (P. 4.1)
3A-DA-10 Evaluate the tradeoffs in how data elements are organized and where data is stored. (P. 3.3)
3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. (P. 5.2)
3A-AP-20 Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. (P.7.3)
3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible. (P. 6.3)
3A-IC-25 Test and refine computational artifacts to reduce bias and equity deficits. (P. 1.2)
3A-IC-28 Explain the beneficial and harmful effects that intellectual property laws can have on innovation. (P. 7.3)</td></tr>
<tr><td>Computer Science Practices</td><td>CS_P_1 Fostering an Inclusive Computing Culture
CS_P_1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.
CS_P_3 Recognizing and Defining Computational Problems
CS_P_3.3 Evaluate whether it is appropriate and feasible to solve a problem computationally.
CS_P_4 Developing and Using Abstractions
CS_P_4.1 Extract common features from a set of interrelated processes or complex phenomena.
CS_P_5 Creating Computational Artifacts</td></tr>
</table>

	CS_P_5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue. CS_P_6 Testing and Refining Computational Artifacts CS_P_6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility CS_P_7 Communicating About Computing CS_P_7.2 Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose. CS_P_7.3 Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.
Mathematics: Common Core	HS.N.Q.2 Define appropriate quantities for the purpose of descriptive modeling. HS.N.Q.3 Choose a level of accuracy appropriate to limitations on measurement when reporting quantities. HS.N.VM.6 (+) Use matrices to represent and manipulate data, e.g., to represent payoffs or incidence relationships in a network. HS.N.VM.8 (+) Add, subtract, and multiply matrices of appropriate dimensions. HS.A.CED.3 Represent constraints by equations or inequalities, and by systems of equations and/or inequalities, and interpret solutions as viable or nonviable options in a modeling context. HS.A.REI.1 Explain each step in solving a simple equation as following from the equality of numbers asserted at the previous step, starting from the assumption that the original equation has a solution. Construct a viable argument to justify a solution method. HS.F.IF.9 Compare properties of two functions each represented in a different way (algebraically, graphically, numerically in tables, or by verbal descriptions). HS.F.BF.1c (+) Compose functions. HS.F.LE.1 Distinguish between situations that can be modeled with linear functions and with exponential functions. HS.F.LE.1b Recognize situations in which one quantity changes at a constant rate per unit interval relative to another. HS.F.LE.1c Recognize situations in which a quantity grows or decays by a constant percent rate per unit interval relative to another. HS.F.LE.5 Interpret the parameters in a linear or exponential function in terms of a context. HS.S.IC.2 Decide if a specified model is consistent with results from a given data-generating process, e.g., using simulation.
Mathematical Practices	MP4 Model with mathematics. MP7 Look for and make use of structure.

Unit 5: Impact of Computing	**Total Learning Hours for Unit:** 30

Unit Summary: Computers and computing have revolutionized our lives. To use computing safely and responsibly, we need to be aware of privacy, security, and ethical issues. As programmers, we need to understand the potential impacts of our programs and be responsible for the consequences. As computer users, we need to understand any potential beneficial or harmful effects and how to protect ourselves and our privacy when using a computer.

AP Computer Science Principles
https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-conceptual-framework-2020-21.pdf
Enduring Understanding: While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences. The use of computing innovations may involve risks to personal safety and identity.

5.1 Beneficial and Harmful Effects
- 5.C Describe the impact of a computing innovation.

5.2 Digital Divide
- 5.C Describe the impact of a computing innovation.

5.3 Computing Bias
- 5.E Evaluate the use of computing based on legal and ethical factors.

5.4 Crowdsourcing
- 1.C Explain how collaboration affects the development of a solution.

5.5 Legal and Ethical Concerns
- 5.E Evaluate the use of computing based on legal and ethical factors.

5.6 Safe Computing
- 5.D Describe the impact of gathering data.
- 5.E Evaluate the use of computing based on legal and ethical factors.

Performance Assessments: (Districts to complete for each unit)
Example assessments for this unit include:
- Marking the text: Provide students with an article that highlights both beneficial and harmful effects of a specific computing innovation and have them mark which effects are beneficial and which are harmful. For each effect the students mark as harmful, have them add notes about whether they think these effects should have been anticipated in advance. For each effect the students mark as beneficial, have the students make notes indicating if they think these benefits were intended or unintended.
- Kinesthetic learning: In small groups, have students create and act out a play or a scene involving privacy and security risks, especially when it comes to personally identifiable information (PII) and the impact of collecting such data. Sample topics might include not recognizing a phishing email, being careless with passwords, downloading a virus accidentally, or not being aware of a search history being kept on a computer. Students could extend their play to include best practices or ways to stay safer when using computing innovations.
- Students explain how computing innovations can be both beneficial to and harmful for its intended audience and beyond. (Code.org)

- Students develop an infographic comparing the perception and reality of the digital divide. (Code.org)
- Students identify common biases in computing innovations and how software developers can avoid bias during the development process. (Code.org)
- Students understand how personal information is encrypted to protect privacy and how data breaches are examples of old or ineffective encryption methods. (Code.org)
- Cybercrime (based on code.org Unit 04 Lesson 10) - Students perform a Rapid Research project investigating a cybercrime event with a particular focus on the data that was lost or stolen and the concerns that arise as a result.

Leadership Alignment: (Districts to complete for each unit)
Leadership alignment must include a unit specific project/activity that aligns with the 21st Century Leadership Skills.
EXAMPLE:

- Creativity and Innovation demonstrated when students create their own websites.
- Students demonstrate Initiative and Self-Direction when they work alone to write an individual program, answer questions about the individual program and about the collaboration (peer review) with their partner.
- Students demonstrate Information Literacy when they Access and Evaluate Information and Use and Manage Information to explore technical challenges and questions that arise from the need to represent digital information in computers and transfer it between people and computational devices.

Core Leadership Skills- Group Skills Standards
2.1 The student will communicate, participate, and advocate effectively in pairs, small groups, teams, and large groups in order to reach common goals.
2.7 The student will demonstrate the ability to train others to understand the established rules and expectations, rationale and consequences and to follow those rules and expectations.
2.8 The student demonstrates the ability to incorporate and utilize the principles of group dynamics in a variety of settings.

Industry Standards and/or Competencies:
National Business Education Association (**NBEA**) www.nbea.org
IMPACT ON SOCIETY
Level 1 Performance Expectations

- Describe how information technology changes social mores, including approaches toward work, family, school, and other cultures.
- Identify the impact of information technologies on the environment and society-both positive and negative.
- Identify the risks of information technology to personal health, safety and privacy.

Level 2 Performance Expectations

- Describe the impact of technology on the knowledge and skills needed for success in the workplace.

Level 3-4 Performance Expectations

- Analyze and compare society's influence on information technology and information technology's influence on society.

SECURITY AND RISK MANAGEMENT
Level 1-2 Performance Expectations

- Discuss the risks of data loss and methods of prevention.

<table>
<tr><td colspan="2">Level 3 Performance Expectations
• Analyze security, privacy, and risk management issues.
• Identify potential risks to enterprise systems from physical or cyber threats.
DIGITAL CITIZENSHIP
Level 1 -2 Performance Expectations
• Identify and explore basic privacy issues associated with technology.
• Discuss basic issues related to responsible use of technology and describe personal or legal consequences of inappropriate use.
• Explain the consequences of illegal and unethical use of information technologies.
Level 3-4 Performance
• Analyze legal and ethical dilemmas within the framework of current laws and legislation (e.g., virus development, hacking, threats, phishing)</td></tr>
<tr><td colspan="2">Aligned Washington State Academic Standards</td></tr>
<tr><td>Computer Science</td><td>3A-NI-05 Give examples to illustrate how sensitive data can be affected by malware and other attacks. (P. 7.2)
3A-NI-06 Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. (P. 3.3)
3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system. (6.3)
3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools. (P. 2.4)
3A-IC-24 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. (P. 1.2)
3A-IC-25 Test and refine computational artifacts to reduce bias and equity deficits. (P. 1.2)
3A-IC-26 Demonstrate ways a given algorithm applies to problems across disciplines. (P. 3.1)
3A-IC-27 Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. (P. 2.4)
3A-IC-28 Explain the beneficial and harmful effects that intellectual property laws can have on innovation. (P. 7.3)
3A-IC-29 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. (P. 7.2)
3A-IC-30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics. (P. 7.3)</td></tr>
<tr><td>Computer Science Practices</td><td>CS_P_1 Fostering an Inclusive Computing Culture
CS_P_1.1 Include the unique perspectives of others and reflect on one’s own perspectives when designing and developing computational products.
CS_P_1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.</td></tr>
</table>

	CS_P_2 Collaborating Around Computing CS_P_2.4 Evaluate and select technological tools that can be used to collaborate on a project. CS_P_3 Recognizing and Defining Computational Problems CS_P_3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally. CS_P_3.3 Evaluate whether it is appropriate and feasible to solve a problem computationally. CS_P_5 Creating Computational Artifacts CS_P_5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue. CS_P_6 Testing and Refining Computational Artifacts CS_P_6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility. CS_P_7 Communicating About Computing CS_P_7.2 Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose. CS_P_7.3 Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.
Mathematics: Common Core	HS.N.Q.1 Use units as a way to understand problems and to guide the solution of multi-step problems; choose and interpret units consistently in formulas; choose and interpret the scale and the origin in graphs. and data displays. HS.N.Q.2 Define appropriate quantities for the purpose of descriptive modeling. HS.N.Q.3 Choose a level of accuracy appropriate to limitations on measurement when reporting quantities. HS.A.SSE.1 Interpret expressions that represent a quantity in terms of its context.* HS.A.SSE.1a Interpret parts of an expression, such as terms, factors, and coefficients. HS.A.SSE.1b Interpret complicated expressions by viewing one or more of their parts as a single entity. HS.A.SSE.2 Use the structure of an expression to identify ways to rewrite it. HS.A.SSE.3c Use the properties of exponents to transform expressions for exponential functions. HS.A.APR.6 Rewrite simple rational expressions in different forms; write a(x)/b(x) in the form q(x) + r(x)/b(x), where a(x), b(x), q(x), and r(x) are polynomials with the degree of r(x) less than the degree of b(x), using inspection, long division, or, for the more complicated examples, a computer algebra system. HS.A.CED.1 Create equations and inequalities in one variable and use them to solve problems. Include equations arising from linear and quadratic functions, and simple rational and exponential functions. HS.A.CED.2 Create equations in two or more variables to represent relationships between quantities; graph equations on coordinate axes with labels and scales. HS.A.CED.3 Represent constraints by equations or inequalities, and by systems of equations and/or inequalities, and interpret solutions as viable or nonviable options in a modeling context.

	HS.A.REI.2 Solve simple rational and radical equations in one variable, and give examples showing how extraneous solutions may arise. HS.A.REI.3 Solve linear equations and inequalities in one variable, including equations with coefficients represented by letters. HS.F.IF.2 Use function notation, evaluate functions for inputs in their domains, and interpret statements that use function notation in terms of a context. HS.F.BF.1 Write a function that describes a relationship between two quantities.* HS.F.BF.1a Determine an explicit expression, a recursive process, or steps for calculation from a context. HS.S.CP.2 Understand that two events A and B are independent if the probability of A and B occurring together is the product of their probabilities and use this characterization to determine if they are independent.
Mathematical Practices	MP1 Make sense of problems and persevere in solving them. MP2 Reason abstractly and quantitatively. MP3 Construct viable arguments and critique the reasoning of others. MP7 Look for and make use of structure. MP8 Look for and express regularity in repeated reasoning.

Unit 6: Capstone **-** Create performance task	**Total Learning Hours for Unit:** 15

Unit Summary: The Create performance task requires at least 15 hours of dedicated class time for students to complete. Programming is a collaborative and creative process that brings ideas to life through the development of software. In the Create performance task, you will design and implement a program that might solve a problem, enable innovation, explore personal interests, or express creativity. The AP Computer Science Principles Exam is a through-course performance task that assesses Computational Thinking Practices.

AP Computer Science Principles
https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-conceptual-framework-2020-21.pdf
Submission Requirements:
Capture: Select a portion of program code that addresses the prompt(s).
Demonstrate: Provide sufficient evidence for an answer or point being made.
Describe: Provide the relevant features or characteristics of what the program code represents or is being used to accomplish.
Design: Develop a plan for how to accomplish the program specification or requirements.
Explain: Provide information about how or why a relationship, situation, or outcome occurs, listing detailed steps of the algorithm or using evidence and/ or reasoning.
Identify: Provide a name for the specific topic, without elaboration or explanation.
Implement: Recognize and use proper syntax to execute the program design.

Program development

In the program, you must include student-developed program code that contains the following:

- Instructions for input from one of the following: the user (including user actions that trigger events), a device, an online data stream, or a file
- Use of at least one list (or other collection type) to represent a collection of data that is stored and used to manage program complexity and help fulfill the program's purpose. The data abstraction must make the program easier to develop (alternatives would be more complex) or easier to maintain (future changes to the size of the list would otherwise require significant modifications to the program code).
- At least one procedure that contributes to the program's intended purpose, where you have defined: the procedure's name the return type (if necessary) one or more parameters
- An algorithm that includes sequencing, selection, and iteration that is in the body of the selected procedure
- Calls to your student-developed procedure
- Instructions for output (tactile, audible, visual, or textual) based on input and program functionality

Demonstration

A video to demonstrate your program running, including:

- Input to your program
- At least one aspect of the functionality of your program
- Output produced by your program

Performance Assessments: (Districts to complete for each unit)

Assessments for this unit includes:

- Program Purpose and Function assesses students' ability to explain how a code segment or program functions. (Skill 4.A)
- Data Abstraction assesses students' ability to use abstraction to manage complexity in a program. (Skill 3.B)
- Managing Complexity assesses students' ability to explain how abstraction manages complexity. (Skill 3.C)
- Procedural Abstraction assesses students' ability to use abstraction to manage complexity in a program. (Skill 3.B)
- Algorithm Implementation assesses students' ability to implement and apply an algorithm. (Skill 2.B)
- Testing assesses students' ability to investigate the situation, context, or task. (Skill 1.A)

Leadership Alignment: (Districts to complete for each unit)

Leadership alignment must include a unit specific project/activity that aligns with the 21st Century Leadership Skills.

Example:

- Creativity and Innovation demonstrated when students create their own websites.
- Students demonstrate Initiative and Self-Direction when they work alone to write an individual program, answer questions about the individual program and about the collaboration (peer review) with their partner.
- Students demonstrate Information Literacy when they Access and Evaluate Information and Use and Manage Information to explore technical challenges and questions that arise from the need to represent digital information in computers and transfer it between people and computational devices.

Core Leadership Skills- Group Skills Standards

2.1 The student will communicate, participate, and advocate effectively in pairs, small groups, teams, and large groups in order to reach common goals.

2.7 The student will demonstrate the ability to train others to understand the established rules and expectations, rationale and consequences and to

follow those rules and expectations.
2.8 The student demonstrates the ability to incorporate and utilize the principles of group dynamics in a variety of settings.

Industry Standards and/or Competencies:
National Business Education Association (**NBEA**) www.nbea.org
IMPACT ON SOCIETY
Level 1 Performance Expectations

- Describe how information technology changes social mores, including approaches toward work, family, school, and other cultures
- Identify the impact of information technologies on the environment and society-both positive and negative
- Identify the risks of information technology to personal health, safety and privacy

Level 2 Performance Expectations

- Describe the impact of technology on the knowledge and skills needed for success in the workplace.

Level 3-4 Performance Expectations

- Analyze and compare society's influence on information technology and information technology's influence on society.

SECURITY AND RISK MANAGEMENT
Level 1-2 Performance Expectations

- Discuss the risks of data loss and methods of prevention.

Level 3 Performance Expectations

- Analyze security, privacy, and risk management issues
- Identify potential risks to enterprise systems from physical or cyber threats.

DIGITAL CITIZENSHIP
Level 1 -2 Performance Expectations

- Identify and explore basic privacy issues associated with technology.
- Discuss basic issues related to responsible use of technology and describe personal or legal consequences of inappropriate use
- Explain the consequences of illegal and unethical use of information technologies.

Level 3-4 Performance

- Analyze legal and ethical dilemmas within the framework of current laws and legislation (e.g., virus development, hacking, threats, phishing)

Aligned Washington State Academic Standards	
Computer Science	
Computer Science Practices	CS_P_1 Fostering an Inclusive Computing Culture CS_P_1.1 Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products. CS_P_1.2 Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability. CS_P_2 Collaborating Around Computing CS_P_2.4 Evaluate and select technological tools that can be used to collaborate on a project.

	CS_P_3 Recognizing and Defining Computational Problems CS_P_3.1 Identify complex, interdisciplinary, real-world problems that can be solved computationally. CS_P_3.3 Evaluate whether it is appropriate and feasible to solve a problem computationally. CS_P_5 Creating Computational Artifacts CS_P_5.2 Create a computational artifact for practical intent, personal expression, or to address a societal issue. CS_P_6 Testing and Refining Computational Artifacts CS_P_6.3 Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility. CS_P_7 Communicating About Computing CS_P_7.2 Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose. CS_P_7.3 Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.
Mathematics: Common Core	HS.N.Q.1 Use units as a way to understand problems and to guide the solution of multi-step problems; choose and interpret units consistently in formulas; choose and interpret the scale and the origin in graphs. and data displays. HS.N.Q.2 Define appropriate quantities for the purpose of descriptive modeling. HS.N.Q.3 Choose a level of accuracy appropriate to limitations on measurement when reporting quantities. HS.A.SSE.1 Interpret expressions that represent a quantity in terms of its context.* HS.A.SSE.1a Interpret parts of an expression, such as terms, factors, and coefficients. HS.A.SSE.1b Interpret complicated expressions by viewing one or more of their parts as a single entity. HS.A.SSE.2 Use the structure of an expression to identify ways to rewrite it. HS.A.SSE.3c Use the properties of exponents to transform expressions for exponential functions. HS.A.APR.6 Rewrite simple rational expressions in different forms; write a(x)/b(x) in the form q(x) + r(x)/b(x), where a(x), b(x), q(x), and r(x) are polynomials with the degree of r(x) less than the degree of b(x), using inspection, long division, or, for the more complicated examples, a computer algebra system. HS.A.CED.1 Create equations and inequalities in one variable and use them to solve problems. Include equations arising from linear and quadratic functions, and simple rational and exponential functions. HS.A.CED.2 Create equations in two or more variables to represent relationships between quantities; graph equations on coordinate axes with labels and scales. HS.A.CED.3 Represent constraints by equations or inequalities, and by systems of equations and/or inequalities, and interpret solutions as viable or nonviable options in a modeling context. HS.A.REI.2 Solve simple rational and radical equations in one variable, and give examples showing how extraneous solutions may arise.

	HS.A.REI.3 Solve linear equations and inequalities in one variable, including equations with coefficients represented by letters. HS.F.IF.2 Use function notation, evaluate functions for inputs in their domains, and interpret statements that use function notation in terms of a context. HS.F.BF.1 Write a function that describes a relationship between two quantities.* HS.F.BF.1a Determine an explicit expression, a recursive process, or steps for calculation from a context. HS.S.CP.2 Understand that two events A and B are independent if the probability of A and B occurring together is the product of their probabilities and use this characterization to determine if they are independent.
Mathematical Practices	MP1 Make sense of problems and persevere in solving them. MP2 Reason abstractly and quantitatively. MP3 Construct viable arguments and critique the reasoning of others. MP4 Model with mathematics. MP5 Use appropriate tools strategically. MP7 Look for and make use of structure. MP8 Look for and express regularity in repeated reasoning.