Washington Office of Superintendent of
PUBLIC INSTRUCTION

Learning that works for Washington
CTE

Statewide Framework Document for: 110201

Computer Programming

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 and leadership alignment 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 one credit of math beyond Geometry or 3rd math credit.** 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. The details about each mathematical standard can be found at Common Core Mathematics Standards.

School District Name	
Course Title: Computer Programming	**Total Framework Hours:** 180
CIP Code: 110201 ☐ Exploratory ☒ Preparatory	**Date Last Modified:** December 29, 2020
Career Cluster: Information Technology	**Cluster Pathway:** Programming and Software Development
Course Summary: Computer Programming is a preparatory course that counts as one credit of occupational education. In addition, this course will count towards graduation as a mathematics credit. In this full-year class, students learn program design and basic programming in Java. This course is equivalent to a college-level semester introduction to programming and prepares students for the Advanced Placement Computer Science A exam. Topics include primitive types, procedural programming (methods, parameters, return values), basic control structures (if-else, for loop, while loop), array manipulation, file processing, and using and defining objects (identifying reusable components, class relationships). Students learn by designing, writing and testing their own software. Throughout the course, students will explore related topics, such as computer security, ethics, industry opportunities, and career paths.	
Eligible for Equivalent Credit in: Math	**Total Number of Units:** 4

Unit 1: Understand Programs – Concepts and Theory	**Total Learning Hours for Unit:** 20

Unit Summary:

In this unit, students:

- Explain and model concepts in computer science using multiple communication modes (including, for example, oral descriptions, written descriptions, pictures, flowcharts, example code, and pseudo-code).
- Describe what a program will do by tracing each statement by hand, correctly keeping track of flow of control, variable values, and the output of the program.
- Describe and analyze the digital binary representations of primitive data types and the limits they impose; evaluate arithmetic expressions that use numbers in various base systems.
- Construct statements and Boolean expressions that behave as intended using arithmetic, unary, conditional, and relational operators and methods.
- Explain the difference between run-time errors and compile-time errors in programs and use information from exceptions and errors to identify the source of a problem.
- Interpret recursive algorithms, describing how they work at each step and how they compare to non-recursive solutions.

Performance Assessments: (Districts to complete for each unit)

Example assessments for this unit include:

- Explain core concepts in programming using multiple modes of communication.
- Accurately explain the flow of a program.
- Answer conceptual questions about how computers are built and how variables are created, represented, and stored.

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:

- Discuss and examine the cross-platform and device-independent use of Java as a programming language across the globe, allowing teams to work on projects without barriers. Look at companies that utilize a global development concept, like Microsoft and Oracle.
- Compete in local computer programming competitions and STEM events (www.pscsta.org), as well as FBLA competitions.

1. A.1 Use a wide range of idea creation techniques (such as brainstorming).
1. A.3 Elaborate, refine, analyze and evaluate their own ideas in order to improve and maximize creative efforts
2. D.1 Solve different kinds of non-familiar problems in both conventional and innovative ways.
2. D.2 Identify and ask significant questions that clarify various points of view and lead to better solutions.

Industry Standards and/or Competencies:

Computer Science Teachers Association Standards - Computer Science Concepts and Practices:

Computational Thinking

3. Critically examine classical algorithms and implement an original algorithm.
4. Evaluate algorithms by their efficiency, correctness, and clarity.
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

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

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.
6. Anticipate future careers and the technologies that will exist.
8. Deploy various data collection techniques for different types of problems.

Computers and Communications Devices

1. Discuss the impact of modifications on the functionality of application programs.
2. Identify and describe hardware (e.g., physical layers, logic gates, chips, components).
3. Identify and select the most appropriate file format based on trade-offs (e.g., accuracy, speed, ease of manipulation).

Aligned Washington State Academic Standards	
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.6 (+) Use matrices to represent and manipulate data, e.g., to represent payoffs or incidence relationships in a network. HS.N.VM.7 (+) Multiply matrices by scalars to produce new matrices, e.g., as when all of the payoffs in a game are doubled. HS.N.VM.8 (+) Add, subtract, and multiply matrices of appropriate dimensions. HS.N.VM.9 (+) Understand that, unlike multiplication of numbers, matrix multiplication for square matrices is not a commutative operation, but still satisfies the associative and distributive properties. HS.N.VM.10 (+) Understand that the zero and identity matrices play a role in matrix addition and multiplication similar to the role of 0 and 1 in the real numbers. The determinant of a square matrix is nonzero if and only if the matrix has a multiplicative inverse.

	HS.N.VM.11 (+) Multiply a vector (regarded as a matrix with one column) by a matrix of suitable dimensions to produce another vector. Work with matrices as transformations of vectors. HS.N.VM.12 (+) Work with 2 × 2 matrices as a transformations of the plane, and interpret the absolute value of the determinant in terms of area. 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.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.
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. MP6 Attend to precision. MP7 Look for and make use of structure. MP8 Look for and express regularity in repeated reasoning.

Unit 2: Plan Programs – Design and Algorithms	**Total Learning Hours for Unit:** 20

Unit Summary:

In this unit, students:

- Write programs that conform to stylistic conventions for syntax, structure, identifier selection, and commenting.
- Decompose a procedure into multiple methods so that each method has high cohesion with itself and low dependency on other methods; explain why decomposition is useful and how it can be accomplished.
- Compare between and choose appropriate programming constructs, algorithms, and/or data structures that can be used to implement a solution to a specific problem; describe the trade-offs among the solutions in terms of clarity, readability, efficiency, and how idiomatic they are.
- Specify fields and methods for an interface or class, using the public, private, and static keywords appropriately based on object-oriented design
  principles.
- Design, analyze, and implement classes that use abstraction and/or encapsulation to separate the intended behavior of an object from its implementation.
  Design, analyze, and implement classes that use inheritance, composition, and/or polymorphism to consolidate common behavior and structure between multiple objects.

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

- Design programs using the principles of procedural decomposition
- Choose and justify different approaches to solve programming problems
- Plan and design objects using object oriented principles

Possible performance task: *The College Application Quotient Problem*
Students imagine that they have been hired as an intern in their school's counseling office. Their first assignment is to write a simple program that will calculate students' College Application Quotient; basically telling students how strong their application to college would be based on a quick calculation. Students who visit the counseling office will use a computer to enter their GPA, ACT scores, and/or SAT scores into the user interface.
As part of the task, students receive formulas that could have come from the school counselors. Students then write a program in Java that requires basic input from the user. The program should output a College Application Quotient and one of the following messages:

- Way to go! Consider applying to Harvard!
- Your test scores are bringing down your quotient. See a counselor for information about retaking a test.
- Your GPA is lower than it should be to get into college. See your counselor for ideas to improve your grades.

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:

- Examine the economics of the information systems life cycle in terms of implementation and maintenance of information systems solutions, and economic impact of technology in society.
- Compete in local computer programming competitions and STEM events (www.pscsta.org), as well as FBLA competitions.

1. B.1 Develop, implement and communicate new ideas to others effectively.
1. B.2 Be open and responsive to new and diverse perspectives; incorporate group input and feedback into the work.
2. B.1 Analyze how parts of a whole interact with each other to produce overall outcomes in complex systems.
4. A.1 Access information efficiently (time) and effectively (sources).
4. A.2 Evaluate information critically and competently.

Industry Standards and/or Competencies:
Computer Science Teachers Association Standards - Computer Science Concepts and Practices:
Computational Thinking

3. Critically examine classical algorithms and implement an original algorithm.
4. Evaluate algorithms by their efficiency, correctness, and clarity.
6. Compare and contrast simple data structures and their uses (e.g., arrays and lists).
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

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

2. Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).
4. Explore principles of system design in scaling, efficiency, and security.
6. Anticipate future careers and the technologies that will exist.
8. Deploy various data collection techniques for different types of problems.

Computers and Communications Devices

1. Discuss the effect of modifications on the functionality of application programs.

Aligned Washington State Academic Standards	
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.6 (+) Use matrices to represent and manipulate data, e.g., to represent payoffs or incidence relationships in a network. HS.N.VM.7 (+) Multiply matrices by scalars to produce new matrices, e.g., as when all of the payoffs in a game are doubled. HS.N.VM.8 (+) Add, subtract, and multiply matrices of appropriate dimensions. HS.N.VM.9 (+) Understand that, unlike multiplication of numbers, matrix multiplication for square matrices is not a commutative operation, but still satisfies the associative and distributive properties. HS.N.VM.10 (+) Understand that the zero and identity matrices play a role in matrix addition and multiplication similar to the role of 0 and 1 in the real numbers. The determinant of a square matrix is nonzero if and only if the matrix has a multiplicative inverse. HS.N.VM.11 (+) Multiply a vector (regarded as a matrix with one column) by a matrix of suitable dimensions to produce another vector. Work with matrices as transformations of vectors. HS.N.VM.12 (+) Work with 2 × 2 matrices as a transformations of the plane, and interpret the absolute value of the determinant in terms of area. 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.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.* HS.A.SSE.3a Factor a quadratic expression to reveal the zeros of the function it defines. HS.A.SSE.3b Complete the square in a quadratic expression to reveal the maximum or minimum value of the function it defines. HS.A.SSE.3c Use the properties of exponents to transform expressions for exponential functions. HS.A.SSE.4 Derive the formula for the sum of a finite geometric series (when the common ratio is not 1), and use the formula to solve problems. 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.3 Solve linear equations and inequalities in one variable, including equations with coefficients represented by letters. HS.A.REI.4 Solve quadratic equations in one variable. HS.A.REI.4a Use the method of completing the square to transform any quadratic equation in x into an equation of the form (x - p)2 = q that has the same solutions. Derive the quadratic formula from this form. HS.A.REI.4b Solve quadratic equations by inspection (e.g., for x2 = 49), taking square roots, completing the square, the quadratic formula and factoring, as appropriate to the initial form of the equation. Recognize when the quadratic formula gives complex solutions and write them as a ± bi for real numbers a and b. HS.F.IF.1 Understand that a function from one set (called the domain) to another set (called the range) assigns to each element of the domain exactly one element of the range. If f is a function and x is an element of its domain, then f(x) denotes the output of f corresponding to the input x. The graph of f is the graph of the equation y = f(x).

	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.IF.3 Recognize that sequences are functions, sometimes defined recursively, whose domain is a subset of the integers. 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.F.BF.2 Write arithmetic and geometric sequences both recursively and with an explicit formula, use them to model situations, and translate between the two forms.* HS.F.LE.1 Distinguish between situations that can be modeled with linear functions and with exponential functions. HS.F.LE.1a Prove that linear functions grow by equal differences over equal intervals, and that exponential functions grow by equal factors over equal intervals. 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.2 Construct linear and exponential functions, including arithmetic and geometric sequences, given a graph, a description of a relationship, or two input-output pairs (include reading these from a table). HS.F.LE.3 Observe using grapHS. and tables that a quantity increasing exponentially eventually exceeds a quantity increasing linearly, quadratically, or (more generally) as a polynomial function. HS.F.LE.4 For exponential models, express as a logarithm the solution to abct = d where a, c, and dare numbers and the base b is 2, 10, or e; evaluate the logarithm using technology. HS.F.LE.5 Interpret the parameters in a linear or exponential function in terms of a context. HS.G.GMD.1 Give an informal argument for the formulas for the circumference of a circle, area of a circle, volume of a cylinder, pyramid, and cone. Use dissection arguments, Cavalieri's principle, and informal limit arguments. HS.G.GMD.2 (+) Give an informal argument using Cavalieri's principle for the formulas for the volume of a sphere and other solid figures. HS.G.GMD.3 Use volume formulas for cylinders, pyramids, cones, and spheres to solve problems.* HS.S.CP.1 Describe events as subsets of a sample space (the set of outcomes) using characteristics (or categories) of the outcomes, or as unions, intersections, or complements of other events ("or," "and," "not"). HS.S.MD.5 (+) Weigh the possible outcomes of a decision by assigning probabilities to payoff values and finding expected values. HS.S.MD.5a Find the expected payoff for a game of chance.

	HS.S.MD.5b Evaluate and compare strategies on the basis of expected values. HS.S.MD.6 (+) Use probabilities to make fair decisions (e.g., drawing by lots, using a random number generator). 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	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. MP6 Attend to precision. MP7 Look for and make use of structure. MP8 Look for and express regularity in repeated reasoning.

Unit 3: Develop Programs – Implement and Test	**Total Learning Hours for Unit:** 130

Unit Summary:

In this unit, students:

- Develop multiple test cases that verify the behavior of a method or program; describe how testing can be used to support program verification.
- Declare and/or instantiate variables, objects, arrays, and lists, initializing them appropriately when necessary; describe how scope affects where and when variables can be used.
- Design, analyze, and implement programs that use conditional structures to respond to different program states or inputs by executing different blocks of code.
- Design, analyze, and implement methods from an interface or abstract method specification, subject to stated pre-conditions and guaranteeing stated post-conditions.
- Design, analyze, and implement programs that use iteration, both definite and indefinite, to execute repeated and related actions in generalized ways.
- Compose, manipulate, and format String objects; access data from and store data to local objects, user consoles, or external files.
- Access and manipulate data stored in one-dimensional arrays and lists; implement and explain common algorithms such as traversal, sorting, search, insertion, and deletion.
- Use two-dimensional arrays and describe their structure, recognizing that two-dimensional arrays can be operated on either as a collection of one-dimensional arrays or as a two-dimensional grid.

Performance Assessments: (Districts to complete for each unit)

Example assessments for this unit include:

- Write programs that use a variety of coding constructs (if-else statements, for loops, while loops).

- Write programs that create, instantiate, and use a variety of data structures (arrays, lists, user-defined objects).
- Create testing plans for methods that they write and discuss concepts of program verification.

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:

- Investigate and discuss the impact of computing technology in the area of Health Care and Bioinformatics.
- Compete in local computer programming competitions and STEM events (www.pscsta.org), as well as FBLA competitions.

6. A.1 Use technology as a tool to research, organize, evaluate and communicate information.
6. A.3 Apply a fundamental understanding of the ethical/legal issues surrounding the access and use of information technologies.
8. A.3 Utilize time and manage workload efficiently.
10.B.1 Demonstrate additional attributes associated with producing high quality products including the abilities to:
10. B.1a Work positively and ethically.
10. B.1b Manage time and projects effectively.
10. B.1d Participate actively, as well as be reliable and punctual.

Industry Standards and/or Competencies:

Computer Science Teachers Association Standards - Computer Science Concepts and Practices:

Computational Thinking

1. Critically examine classical algorithms and implement an original algorithm.
2. Evaluate algorithms by their efficiency, correctness, and clarity.
6. Compare and contrast simple data structures and their uses (e.g., arrays and lists).
3. Use models and simulations to help formulate, refine, and test scientific hypotheses.
4. Analyze data and identify patterns through modeling and simulation.
5. Decompose a problem by defining new functions and classes. Collaboration
6. Use project collaboration tools, version control systems, and Integrated Development Environments (IDEs) while working on a collaborative software project.
7. Demonstrate the software life cycle process by participating on a software project team.
8. Evaluate programs written by others for readability and usability.

Computing Practice and Programming

2. Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).
4. Explore principles of system design in scaling, efficiency, and security.
6. Anticipate future careers and the technologies that will exist.
8. Deploy various data collection techniques for different types of problems.

Computers and Communications Devices

1. Discuss the impact of modifications on the functionality of application programs.

Aligned Washington State Academic Standards	
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.6 (+) Use matrices to represent and manipulate data, e.g., to represent payoffs or incidence relationships in a network. HS.N.VM.7 (+) Multiply matrices by scalars to produce new matrices, e.g., as when all of the payoffs in a game are doubled. HS.N.VM.8 (+) Add, subtract, and multiply matrices of appropriate dimensions. HS.N.VM.9 (+) Understand that, unlike multiplication of numbers, matrix multiplication for square matrices is not a commutative operation, but still satisfies the associative and distributive properties. HS.N.VM.10 (+) Understand that the zero and identity matrices play a role in matrix addition and multiplication similar to the role of 0 and 1 in the real numbers. The determinant of a square matrix is nonzero if and only if the matrix has a multiplicative inverse. HS.N.VM.11 (+) Multiply a vector (regarded as a matrix with one column) by a matrix of suitable dimensions to produce another vector. Work with matrices as transformations of vectors. HS.N.VM.12 (+) Work with 2 × 2 matrices as a transformations of the plane, and interpret the absolute value of the determinant in terms of area. 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.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.3 Solve linear equations and inequalities in one variable, including equations with coefficients represented by letters. HS.A.REI.4 Solve quadratic equations in one variable. HS.A.REI.4a Use the method of completing the square to transform any quadratic equation in x into an equation of the form (x - p)2 = q that has the same solutions. Derive the quadratic formula from this form. HS.A.REI.4b Solve quadratic equations by inspection (e.g., for x2 = 49), taking square roots, completing the square, the quadratic formula and factoring, as appropriate to the initial form of the equation. Recognize when the quadratic formula gives complex solutions and write them as a ± bi for real numbers a and b. HS.F.IF.1 Understand that a function from one set (called the domain) to another set (called the range) assigns to each element of the domain exactly one element of the range. If f is a function and x is an element of its domain, then f(x) denotes the output of f corresponding to the input x. The graph of f is the graph of the equation y = f(x). 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.IF.3 Recognize that sequences are functions, sometimes defined recursively, whose domain is a subset of the integers. 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.F.BF.2 Write arithmetic and geometric sequences both recursively and with an explicit formula, use them to model situations, and translate between the two forms.* HS.F.LE.1 Distinguish between situations that can be modeled with linear functions and with exponential functions. HS.F.LE.1a Prove that linear functions grow by equal differences over equal intervals, and that exponential functions grow by equal factors over equal intervals. 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.2 Construct linear and exponential functions, including arithmetic and geometric sequences, given a graph, a description of a relationship, or two input-output pairs (include reading these from a table). HS.F.LE.3 Observe using grapHS. and tables that a quantity increasing exponentially eventually exceeds a quantity increasing linearly, quadratically, or (more generally) as a polynomial function.

	HS.F.LE.4 For exponential models, express as a logarithm the solution to abct = d where a, c, and dare numbers and the base b is 2, 10, or e; evaluate the logarithm using technology. HS.F.LE.5 Interpret the parameters in a linear or exponential function in terms of a context. HS.G.GMD.1 Give an informal argument for the formulas for the circumference of a circle, area of a circle, volume of a cylinder, pyramid, and cone. Use dissection arguments, Cavalieri's principle, and informal limit arguments. HS.G.GMD.2 (+) Give an informal argument using Cavalieri's principle for the formulas for the volume of a sphere and other solid figures. HS.G.GMD.3 Use volume formulas for cylinders, pyramids, cones, and spheres to solve problems.* HS.S.CP.1 Describe events as subsets of a sample space (the set of outcomes) using characteristics (or categories) of the outcomes, or as unions, intersections, or complements of other events ("or," "and," "not"). HS.S.MD.5 (+) Weigh the possible outcomes of a decision by assigning probabilities to payoff values and finding expected values. HS.S.MD.5a Find the expected payoff for a game of chance. HS.S.MD.5b Evaluate and compare strategies on the basis of expected values. HS.S.MD.6 (+) Use probabilities to make fair decisions (e.g., drawing by lots, using a random number generator). 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	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. MP6 Attend to precision. MP7 Look for and make use of structure. MP8 Look for and express regularity in repeated reasoning.

Unit 4: Ethics in Computing and Society	**Total Learning Hours for Unit:** 10

Unit Summary:

In this unit, students will understand and discuss issues regarding:

- Intellectual property rights and fair use of intellectual property.
- The impact of applications that use databases, particularly over the Internet, on an individual’s right to privacy.
- Economic and legal impact of viruses and other malicious attacks on computer systems.
- The need for fault-tolerant and highly reliable systems for life-critical applications and the resulting need for software engineering

standards.

Performance Assessments: (Districts to complete for each unit)

Example assessments for this unit include:

- Take positions on ethical issues that relate to the use of computers in society.
- Discuss what it means to make an ethical decision.
- Discuss the implicit ethical responsibilities involved in creating software.

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:

- Investigate and discuss the civic responsibility of software engineers and the impact of their decisions on business, society and individuals.
- Compete in local computer programming competitions and STEM events (www.pscsta.org), as well as FBLA competitions.

A.2 Evaluate information critically and competently.

6. A.3 Apply a fundamental understanding of the ethical/legal issues surrounding the access and use of information technologies.

9. B.1 Respect cultural differences and work effectively with people from a range of social and cultural backgrounds.

9. B.2 Respond open-mindedly to different ideas and values.

Industry Standards and/or Competencies:

Computer Science Teachers Association Standards - Computer Science Concepts and Practices:

Computing Practice and Programming

5. Anticipate future careers and the technologies that will exist.

Community, Global, and Ethical Impacts

1. Demonstrate ethical use of modern communication media and devices.
2. Analyze the beneficial and harmful effects of computing innovations.
5. Identify laws and regulations that affect the development and use of software.
6. Analyze the effect 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.

Aligned Washington State Academic Standards	
Mathematics: Common Core	
Mathematical Practices	