ESL
SCIENCE
BUSINESS
BILINGUAL
PRESCHOOL
MATHEMATICS
LIBRARY MEDIA
SOCIAL STUDIES
WORLD LANGUAGES
GIFTED & TALENTED
TECHNOLOGY EDUCATION
ENGLISH LANGUAGE ARTS
FINE & PERFORMING ARTS
FAMILY & CONSUMER SCIENCE
HEALTH & PHYSICAL EDUCATION

RAHWAY PUBLIC SCHOOLS

CURRICULUM & INSTRUCTION

Content Area: Mathematics

Course: Advanced Placement Computer Science Principles

Grade Level: 10-12

This curriculum is part of the Educational Program of Studies of the Rahway Public Schools.

ACKNOWLEDGMENTS

Jeffery Kurczeski,

Program Supervisor of 7-12 Math and Science and 9-12 Business & Technology Education

The Board acknowledges the following who contributed to the preparation of this curriculum.

Ambika Bhosale, Computer Science Teacher

Dr. Tiffany A. Beer, Director of Curriculum and Instruction

Dr. Aleya Shoieb, Superintendent of Schools

Subject/Course Title:
Advanced Placement Computer Science Principles
Grades 10-12

Date of Board Adoption:
August 27, 2024

RAHWAY PUBLIC SCHOOLS CURRICULUM

Advanced Placement Computer Science Principles: Grades 10-12

PACING GUIDE

Unit	Title	Pacing
1	Creative Development	3 weeks
2	Data	4 weeks
3	Algorithms and Programming (Variables & Assignments, Data Abstraction, Mathematical Expressions, Strings and Boolean Expressions)	4 weeks
4	Algorithms and Programming (Conditionals, Nested Conditionals and Iteration)	4 weeks
5	Algorithms and Programming (Developing Algorithms and Lists)	4 weeks
6	Algorithms and Programming (Binary Search, Calling Procedures and Developing Procedures)	4 weeks
7	Algorithms and Programming (Libraries, Random Values, Simulations, Algorithmic Efficiency and Undecidable Problems)	4 weeks
8	Computer Systems and Networks	4 weeks
9	Impacts of Computing	2 weeks

ACCOMMODATIONS

504 Accommodations:

- Provide scaffolded vocabulary and vocabulary lists.
- Provide extra visual and verbal cues and prompts.
- Provide adapted/alternate/excerpted versions of the text and/or modified supplementary materials.
- Provide links to audio files and utilize video clips.
- Provide graphic organizers and/or checklists.
- Provide modified rubrics.
- Provide a copy of teaching notes, especially any key terms, in advance.
- Allow additional time to complete assignments and/or assessments.
- Provide shorter writing assignments.
- Provide sentence starters.
- Utilize small group instruction.
- Utilize Think-Pair-Share structure.
- Check for understanding frequently.
- Have student restate information.
- Support auditory presentations with visuals.
- Weekly home-school communication tools (notebook, daily log, phone calls or email messages).
- Provide study sheets and teacher outlines prior to assessments.
- Quiet corner or room to calm down and relax when anxious.
- Reduction of distractions.
- Permit answers to be dictated.
- Hands-on activities.
- Use of manipulatives.
- Assign preferential seating.
- No penalty for spelling errors or sloppy handwriting.
- Follow a routine/schedule.
- Provide student with rest breaks.
- Use verbal and visual cues regarding directions and staying on task.
- Assist in maintaining agenda book.

IEP Accommodations:

- Provide scaffolded vocabulary and vocabulary lists.
- Differentiate reading levels of texts (e.g., Newsela).
- Provide adapted/alternate/excerpted versions of the text and/or modified supplementary materials.
- Provide extra visual and verbal cues and prompts.
- Provide links to audio files and utilize video clips.
- Provide graphic organizers and/or checklists.
- Provide modified rubrics.
- Provide a copy of teaching notes, especially any key terms, in advance.
- Provide students with additional information to supplement notes.
- Modify questioning techniques and provide a reduced number of questions or items on tests.
- Allow additional time to complete assignments and/or assessments.
- Provide shorter writing assignments.
- Provide sentence starters.
- Utilize small group instruction.
- Utilize Think-Pair-Share structure.
- Check for understanding frequently.
- Have student restate information.
- Support auditory presentations with visuals.
- Provide study sheets and teacher outlines prior to assessments.
- Use of manipulatives.
- Have students work with partners or in groups for reading, presentations, assignments, and analyses.
- Assign appropriate roles in collaborative work.
- Assign preferential seating.
- Follow a routine/schedule.

Gifted and Talented Accommodations:

- Differentiate reading levels of texts (e.g., Newsela).
- Offer students additional texts with higher lexile levels.
- Provide more challenging and/or more supplemental readings and/or activities to deepen understanding.
- Allow for independent reading, research, and projects.
- Accelerate or compact the curriculum.
- Offer higher-level thinking questions for deeper analysis.
- Offer more rigorous materials/tasks/prompts.
- Increase number and complexity of sources.
- Assign group research and presentations to teach the class.
- Assign/allow for leadership roles during collaborative work and in other learning activities.

ML Accommodations:

- Provide extended time.
- Assign preferential seating.
- Assign peer buddy who the student can work with.
- Check for understanding frequently.
- Provide language feedback often (such as grammar errors, tenses, subject-verb agreements, etc…).
- Have student repeat directions.
- Make vocabulary words available during classwork and exams.
- Use study guides/checklists to organize information.
- Repeat directions.
- Increase one-on-one conferencing.
- Allow student to listen to an audio version of the text.
- Give directions in small, distinct steps.
- Allow copying from paper/book.
- Give student a copy of the class notes.

	● Provide written and oral instructions. ● Differentiate reading levels of texts (e.g., Newsela). ● Shorten assignments. ● Read directions aloud to student. ● Give oral clues or prompts. ● Record or type assignments. ● Adapt worksheets/packets. ● Create alternate assignments. ● Have student enter written assignments in criterion, where they can use the planning maps to help get them started and receive feedback after it is submitted. ● Allow student to resubmit assignments. ● Use small group instruction. ● Simplify language. ● Provide scaffolded vocabulary and vocabulary lists. ● Demonstrate concepts possibly through the use of visuals. ● Use manipulatives. ● Emphasize critical information by highlighting it for the student. ● Use graphic organizers. ● Pre-teach or pre-view vocabulary. ● Provide student with a list of prompts or sentence starters that they can use when completing a written assignment. ● Provide audio versions of the textbooks. ● Highlight textbooks/study guides. ● Use supplementary materials. ● Give assistance in note taking ● Use adapted/modified textbooks. ● Allow use of computer/word processor. ● Allow student to answer orally, give extended time (time-and-a-half). ● Allow tests to be given in a separate location (with the ESL teacher). ● Allow additional time to complete assignments and/or assessments. ● Read question to student to clarify. ● Provide a definition or synonym for words on a test that do not impact the validity of the exam. ● Modify the format of assessments. ● Shorten test length or require only selected test items. ● Create alternative assessments. ● On an exam other than a spelling test, don’t take points off for spelling errors.

UNIT 1 OVERVIEW

Content Area: Mathematics

Unit Title: Creative Development

Target Course/Grade Level: Advanced Placement Computer Science Principles/Grades 10-12

Unit Summary: Collaboration is crucial when developing computing innovations because having multiple perspectives offers opportunities to improve the design innovations. In the Big Idea, students work collaboratively to design and develop programs using an iterative development process. They identify the needs of all users by gathering input from different backgrounds and demographics. Once the program is developed, they test it to ensure it meets these needs. When designing a solution to a problem, programmers consider both the program itself and the way the user will interact with the program: the user interface. A well-designed user interface makes it easy for the user to understand what data is required as input for the program to complete its tasks. Students will identify and correct errors.

Approximate Length of Unit: 3 weeks

LEARNING TARGETS

NJ Student Learning Standards:

8.1.12.AP.1 Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.7 Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
8.1.12.AP.8 Evaluate and refine computational artifacts to make them more usable and accessible.
8.1.12.AP.9 Collaboratively document and present design decisions in the development of complex programs.
8.2.12.ED.1 Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.
8.2.12.ED.4 Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.
8.1.12.CS.4 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

College Board Standards:

CRD-1.A Explain how computing innovations are improved through collaboration.
CRD-1.B Explain how computing innovations are developed by groups of people.
CRD-1.C Demonstrate effective interpersonal skills during collaboration.
CRD-2.A Describe the purpose of a computing innovation.
CRD-2.B Explain how a program of code segment functions.
CRD-2.C Identify input(s) to a program.
CRD-2.D Identify output(s) produced by a program.
CRD-2.E Develop a program using a development process.

CRD-2.F Design a program and its user interface.
CRD-2.G Describe the purpose of a code segment or program by writing documentation.
CRD-2.H Acknowledge code segments used from other sources.
CRD-2.I For errors in an algorithm or program:
a. Identify the error.
b. Correct the error.
CRD-2.J Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program.

Career Readiness, Life Literacies, and Key Skills:
9.3.IT-PRG.1 Analyze customer software needs and requirements.
9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications.
9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.
9.3.IT-PRG.6 Program a computer application using the appropriate programming language.
9.3.IT-WD.10 Comply with intellectual property laws, copyright laws and ethical practices when creating Web/digital communications.
9.4.12.CI.1 Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.IML.3 Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions.
9.4.12.TL.1 Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.
9.4.12.TL.4 Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.
9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving.

Interdisciplinary Connections and Standards:
ELA
RI.CI.11–12.2 Determine two or more central ideas of an informational text and analyze how they are developed and refined over the course of a text, including how they interact and build on one another to provide a complex account or analysis; provide an objective summary of the text.
RI.MF.11–12.6 Synthesize complex information across multiple sources and formats to develop ideas, resolve conflicting information, or develop an interpretation that goes beyond explicit text information (e.g., express a personal point of view, new interpretation of the concept).
SL.II.11–12.2 Integrate multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) in order to make informed decisions and solve problems, evaluating the credibility and accuracy of each source and noting any discrepancies among the data.
Mathematics
A.SSE.A.1 Interpret expressions that represent a quantity in terms of its context.
a. Interpret parts of an expression, such as terms, factors, and coefficients.
b. Interpret complicated expressions by viewing one or more of their parts as a single entity.
A.SSE.A.2 Use the structure of an expression to identify ways to rewrite it.
A.SSE.B.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.
a. Factor a quadratic expression to reveal the zeros of the function it defines.
b. Complete the square in a quadratic expression to reveal the max or min value of the function it defines.
c. Use the properties of exponents to transform expressions for exponential functions.
A.SSE.B.4 Derive and/or explain the formula for the sum of a finite geometric series (when the common ratio is not 1), and use the formula to solve problems.

Unit Understandings:

Students will understand that…

- Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.
- Developers create and innovate using an iterative design process that's user-focused, incorporates implementation/feedback cycles, and leaves ample room for experimentation and risk-taking.

Unit Essential Questions:

- How has working collaboratively with other students improved an overall project?
- What are some ways you can collect additional feedback on your program to use for improvements?
- What are some ways you currently plan your work before starting a project?
- What apps or programs have you stopped using because you didn't like the design of how you interacted with it?

Knowledge and Skills:

Students will know...

- A computing innovation includes a program as an integral part of its function.
- A computing innovation can be physical (e.g., self-driving car), non-physical computing software (e.g., picture editing software), or a nonphysical computing concept (e.g., e-commerce).
- Effective Collaboration Produces Computing innovation that reflects the diversity of talents and perspectives of those who designed it.
- Collaboration that includes diverse perspectives helps avoid bias in the development of computing innovations.
- Consultation and communication with users are important aspects of the development of computing innovations.
- Information gathered from potential users can be used to understand the purpose of a program from diverse perspectives and to develop a program that fully incorporates these perspectives.
- Online tools support collaboration by allowing programmers to share and provide feedback on ideas and documents.
- Common models such as pair programming exist to facilitate collaboration.
- Effective Collaborative Team Practice interpersonal skills, including but not limited to:
    - communication
    - consensus building
    - conflict resolution
    - negotiation
- The purpose of computing innovations is to solve problems or to pursue interests through creative expression.
- An understanding of the purpose of a computing innovation provides developers with an improved ability to develop that computing innovation.
- A program is a collection of program statements that perform a specific task when run by a computer. A program is often referred to as software.
- A code segment is a collection of program statements that is part of a program.
- A program needs to work for a variety of inputs and situations.
- The behavior of a program is how a program functions during execution and is often described by how a user interacts with it.
- A program can be described broadly by what it does, or in more detail by both what the program does and how the program statements accomplish this function.
- Program inputs are data sent to a computer for processing by a program. Input can come in a variety of forms, such as tactile, audio, visual, or text.
- An event is associated with an action and supplies input data to a program.

- Events can be generated when a key is pressed, a mouse is clicked, a program is started, or any other defined action occurs that affects the flow of execution.
- Inputs usually affect the output produced by a program.
- In event-driven programming, program statements are executed when triggered rather than through the sequential flow control.
- Input can come from a user or other programs.
- Program outputs are any data sent from a program to a device. Program output can come in a variety of forms, such as tactile, audio, visual, or text.
- Program output is usually based on a program's input or prior state (e.g., internal values).
- A development process can be ordered and intentional, or exploratory in nature.
- There are multiple development processes. The following phases are commonly used when developing a program:
    - investigating and reflecting
    - designing
    - prototyping
    - testing
- A development process that is iterative requires refinement and revision based on feedback, testing, for reflection throughout the process. This may require revisiting earlier phases of the process.
- A development process that is incremental is one that breaks the problem into smaller pieces and makes sure each piece works before adding it to the whole.
- The design of a program incorporates investigation to determine its requirements.
- Investigation in a development process is useful for understanding and identifying the program constraints, as well as the concerns and interests of the people who will use the program.
- Some ways investigation can be performed are as follows:
    - collecting data through surveys
    - user testing
    - interviews
    - direct observations
- Program requirements describe how a program functions and may include a description of user interactions that a program must provide.
- A Program's Specification Defines the requirements for the program.
- In the development process, the design phase outlines how to accomplish a given program specification.
- The design phase of a program may include:
    - brainstorming
    - planning and storyboarding
    - organizing the program into modules and functional components
    - creation of diagrams that represent the layouts of the user interface
    - development of a testing strategy for the program
- Program documentation is a written description of the function of a code segment, event, procedure, or program and how it was developed.
- Comments are a form of program documentation written into the program to be read by people and do not affect how a program runs.
- Programmers should document a program throughout its development.
- Program documentation helps in developing and maintaining correct programs when working individually or in collaborative programming environments.
- Not all programming environments support comments, so other methods of documentation may be required.
- It is important to acknowledge any code segments that were developed collaboratively or by another source.

- Acknowledgment of a code segment(s) written by someone else and used in a program can be in the program documentation. The acknowledgment should include the origin or original author's name.
- A logic error is a mistake in the algorithm or program that causes it to behave incorrectly or unexpectedly.
- A syntax error is a mistake in the program where the rules of the programming language are not followed.
- A run-time error is a mistake in the program that occurs during the execution of a program. Programming languages define their own runtime errors.
- An overflow error is an error that occurs when a computer attempts to handle a number that is outside of the defined range of values.
- The following are effective ways to find and correct errors: test cases, hand tracing, visualizations, debuggers, and adding extra output statement(s).
- In the development process, testing uses defined inputs to ensure that an algorithm or program is producing the expected outcomes. Programmers use the results from testing to revise their algorithms or programs.
- Defined inputs used to test a program should demonstrate the different expected outcomes that are at or just beyond the extremes (minimum and maximum) of input data.
- Program requirements are needed to identify appropriate defined inputs for testing.

Students will be able to...

- Explain how collaboration affects the development of a solution.
- Collaborate in the development of solutions.
- Investigate the situation, context, or task.
- Generalize Data Sources through variables.
- Explain how a code segment or program functions.
- Determine and design an appropriate method or approach to achieve the purpose.
- Explain how collaboration affects the development of a solution.
- Explain how a code segment or program functions.
- Acknowledge the intellectual property of others.
- Identify and correct errors in algorithms and programs, including error discovery through testing.

EVIDENCE OF LEARNING

Assessment:

What evidence will be collected and deemed acceptable to show that students truly "understand"?

- End of Unit Common Assessment - See folder for assessment links.
- CodeHS.com
- Lesson exercises
- Class participation
- Class discussion

Learning Activities:

What differentiated learning experiences and instruction will enable all students to achieve the desired results?

- Group programming projects
- Teacher demonstrations over Screen Share software
- Reinforcement worksheets and extra practice

- Paired-programming challenges
- Individualized student-to-teacher code demonstrations
- Sharing ideas and classroom discussions
- Sample Activities:
    - **Sharing and Responding:** Have 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, have 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.

RESOURCES

Teacher Resources:

- Demonstrations of worked-out solutions from CodeHS.com
- Teacher designed worksheets
- CodeHS.com lesson exercises and videos
- College Board AP Classroom Resources and Quizzes

Equipment Needed:

- Classroom Computers/Chromebooks
- Access to high-speed internet
- AP Classroom College Board login
- CodeHS.org login

UNIT 2 OVERVIEW

Content Area: Mathematics

Unit Title: Data

Target Course/Grade Level: Advanced Placement Computer Science Principles/Grades 10-12

Unit Summary: Students will gain a deep understanding of how information is stored on a computer in binary and seamlessly translated into what is seen on the screen or heard through speakers. Students will also learn how data is processed. Connecting the foundational principles of how number systems operate to the decimal number system is likely to help students lean on their prior knowledge when working with binary numbers. Students will learn the difference between the decimal number system, hexadecimal, and the binary number system, and how to convert between the number systems. Various image manipulation filters and data compression algorithms can be applied to store pixel data. Students will focus on the benefits and disadvantages of using data compression algorithms and data encryption and decryption algorithms. Students will learn how computers are used to collect, store, manipulate, and visualize data in order to answer questions gain knowledge of the world, and learn about the impact of visually representing data to make information easier to analyze and recognize patterns.

Approximate Length of Unit: 4 weeks

LEARNING TARGETS

NJ Student Learning Standards:

8.1.12.DA.1 Create interactive data visualizations using software tools to help others better understand real world phenomena, including climate change.
8.1.12.DA.2 Describe the trade-offs in how and where data is organized and stored.
8.1.12.DA.3 Translate between decimal numbers and binary numbers.
8.1.12.DA.4 Explain the relationship between binary numbers and the storage and use of data in a computing device.
8.1.12.DA.5 Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.
8.1.12.DA.6 Create and refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.

College Board Standards:

DAT-1.A Explain how data can be represented using bits
DAT-1.B Explain the consequences of using bits to represent data
DAT-1.C For binary numbers:
a. Calculate the binary (base 2) equivalent of a positive integer (base 10) and vice versa.
b. Compare and order binary numbers.
DAT-1.D Compare data compression algorithms to determine which is best in a particular context
DAT-2.A Describe what information can be extracted from data.
DAT-2.B Describe what information can be extracted from metadata.

DAT-2.C Identify the challenges associated with processing data
DAT-2.D Extract information from data using a program.
DAT-2.E Explain how programs can be used to gain insight and knowledge from data.

Career Readiness, Life Literacies, and Key Skills:
9.3.IT‑PRG.10 Design, create and maintain a database.
9.4.12.CI.1 Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.IML.3 Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions.
9.4.12.IML.4 Assess and critique the appropriateness and impact of existing data visualizations for an intended audience.
9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving.

Interdisciplinary Connections and Standards:
ELA
RI.CI.11–12.2 Determine two or more central ideas of an informational text and analyze how they are developed and refined over the course of a text, including how they interact and build on one another to provide a complex account or analysis; provide an objective summary of the text.
RI.MF.11–12.6 Synthesize complex information across multiple sources and formats to develop ideas, resolve conflicting information, or develop an interpretation that goes beyond explicit text information (e.g., express a personal point of view, new interpretation of the concept).
SL.II.11–12.2 Integrate multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) in order to make informed decisions and solve problems, evaluating the credibility and accuracy of each source and noting any discrepancies among the data.
Mathematics
A.SSE.A.1 Interpret expressions that represent a quantity in terms of its context.
a. Interpret parts of an expression, such as terms, factors, and coefficients.
b. Interpret complicated expressions by viewing one or more of their parts as a single entity.
A.SSE.A.2 Use the structure of an expression to identify ways to rewrite it.
A.SSE.B.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.
a. Factor a quadratic expression to reveal the zeros of the function it defines.
b. Complete the square in a quadratic expression to reveal the max or min value of the function it defines.
c. Use the properties of exponents to transform expressions for exponential functions.
A.SSE.B.4 Derive and/or explain the formula for the sum of a finite geometric series (when the common ratio is not 1), and use the formula to solve problems.

Unit Understandings:
Students will understand that…
- 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.

Unit Essential Questions:
- How can we use 1s and 0s to represent something complex like a video of the marching band playing a song?
- How can you predict the attendance at a school event using data gathered from social media?

- When is it more appropriate to use a computer to analyze data than to complete the analysis by hand?

Knowledge and Skills:
Students will know...

- Data values can be stored in variables, lists of items, or standalone constants and can be passed as input to (or output from) procedures.
- Computing devices represent data digitally, meaning that the lowest-level components of any value are bits.
- A bit is shorthand for a binary digit and is either 0 or 1.
- A byte is 8 bits.
- Abstraction is the process of reducing complexity by focusing on the main idea. By hiding details irrelevant to the question at hand and bringing together related and useful details, abstraction reduces complexity and allows one to focus on the idea.
- Bits are grouped to represent abstractions. These abstractions include, but are not limited to, numbers, characters, and color.
- The same sequence of bits may represent different types of data in different contexts.
- Analog data have values that change smoothly, rather than in discrete intervals, over time. Some examples of analog data include the pitch and volume of music, the colors of a painting, or the position of a sprinter during a race.
- The use of digital data to approximate real-world analog data is an example of abstraction.
- Analog data can be closely approximated digitally using a sampling technique, which means measuring values of the analog signal at regular intervals called samples. The samples are measured to figure out the exact bits required to store each sample.
- In many programming languages, integers are represented by a fixed number of bits, which limits the range of integer values and mathematical operations on those values. This limitation can result in overflow or other errors.
- Other programming languages provide an abstraction through which the size of representable integers is limited only by the size of the computer's memory; this is the case for the language defined in the exam reference sheet.
- In programming languages, the fixed number of bits used to represent real numbers limits the range and mathematical operations on these values; this limitation can result in round-offs and other errors. Some real numbers are represented as approximations in computer storage
- Number bases, including binary and decimal, are used to represent data.
- Binary (base 2) uses only combinations of the digits zero and one.
- Decimal (base 10) uses only combinations of the digits 0 – 9.
- As with decimals, a digit's position in the binary sequence determines its numeric value. The numeric value is equal to the bit's value (0 or 1) multiplied by the place value of its position.
- The place value of each position is determined by the base raised to the power of the position. Positions are numbered starting at the rightmost position with 0 and increasing by 1 for each subsequent position to the left.
- Data compression can reduce the size (number of bits) of transmitted or stored data.
- Fewer bits do not necessarily mean less information.
- The amount of size reduction from compression depends on both the amount of redundancy in the original data representation and the compression algorithm applied.
- Lossless data compression algorithms can usually reduce the number of bits stored or transmitted while guaranteeing complete reconstruction of the original data.
- Lossy data compression algorithms can significantly reduce the number of bits stored or transmitted but only allow the reconstruction of an approximation of the original data.
- Lossy data compression algorithms can usually reduce the number of bits stored or transmitted more than lossless compression algorithms.

- In situations where quality or ability to reconstruct the original is maximally important, lossless compression algorithms are typically chosen.
- In situations where minimizing data size transmission time is maximally important, lossy compression algorithms are typically chosen
- Information is the collection of facts and patterns extracted from data.
- Data provides opportunities for identifying trends, making connections, and addressing problems.
- Digitally processed data may show a correlation between variables. A correlation found in data does not necessarily indicate that a causal relationship exists. Additional research is needed to understand the exact nature of the relationship.
- A single source does not contain the data needed to draw a conclusion. It may be necessary to combine data from a variety of sources to formulate a conclusion.
- Metadata is data about data. For example, the piece of data may be an image, while the metadata may include the date of creation or the file size of the image.
- Changes and deletions made to metadata do not change the primary data.
- Metadata is used for finding, organizing, and managing information.
- Metadata can increase the effective use of data or data sets by providing additional information.
- Metadata allows data to be structured and organized.
- The ability to process data depends on the capabilities of the users and their tools.
- Datasets pose challenges regardless Of Size, such as:
    - the need to clean data
    - incomplete data
    - invalid data
    - the need to combine data sources
- Depending on how data were collected, they may not be uniform. For example, if users enter data into an open field, the way they choose to abbreviate, spell, or capitalize something may vary from user to user.
- Cleaning data is a process that makes the data uniform without changing its meaning (e.g., replacing all equivalent abbreviations, spellings, and capitalizations with the same word).
- Problems of bias are often created by the type or source of data being collected. Bias is not eliminated by simply collecting more data.
- The size of a dataset affects the amount of information that can be extracted from it.
- Large data sets are difficult to process using a single computer and may require parallel systems.
- The scalability of systems is an important consideration when working with data sets, as the computational capacity of a system affects how data sets can be processed and stored.
- Programs can be used to process data to acquire information.
- Tables, diagrams, text, and other visual tools can be used to communicate insight and knowledge gained from data.
- Search tools are useful for efficiently finding information.
- Data filtering systems are important tools for finding information and recognizing patterns in data.
- Programs such as spreadsheets help efficiently organize and find trends in information.
- Some processes that can be used to extract or modify information from data include the following:
    - transforming every element of a data set, such as doubling every element in a list, or adding a parent's email to every student record
    - filtering dataset, keeping only the positive numbers from a list, or keeping only students who signed up for band from a record of all the students
    - combining or comparing data in some way, such as adding up a list of numbers, or finding the student who has the highest GPA
    - visualizing a dataset through a chart, graph, or other visual representation

- Programs are used in an iterative and interactive way when processing information to allow users to gain insight and knowledge about data.
- Programmers can use programs to filter and clean digital data, thereby gaining insight and knowledge.
- Combining data sources, clustering data, and classifying data are parts of the process of using programs to gain insight and knowledge from data.
- Insight and knowledge can be obtained from translating and transforming digitally represented information.
- Patterns can emerge when data is transformed using programs.

Students will be able to...

- Evaluate solution options.
- Implement and apply an algorithm.
- Explain how abstraction manages complexity
- Explain how knowledge can be generated from data.
- Describe the impact of gathering data.
- Implement and apply an algorithm.
- Explain how knowledge can be generated from data.

EVIDENCE OF LEARNING

Assessment:

What evidence will be collected and deemed acceptable to show that students truly "understand"?

- End of Unit Common Assessment - See folder for assessment links.
- CodeHS.com
- Lesson exercises
- Class participation
- Class discussion
- AP College Board practice Questionnaire at the link below

Learning Activities:

What differentiated learning experiences and instruction will enable all students to achieve the desired results?

- Group programming projects
- Teacher demonstrations over Screen Share software
- Reinforcement worksheets and extra practice
- Paired-programming challenges
- Individualized student-to-teacher code demonstrations
- Sharing ideas and classroom discussions
- Sample Activities:
    - **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.

RESOURCES

Teacher Resources:

- Demonstrations of worked-out solutions from CodeHS.com
- Teacher designed worksheets
- CodeHS.com lesson exercises and videos
- College Board AP Classroom Resources and Quizzes

Equipment Needed:

- Classroom Computers/Chromebooks
- Access to high-speed internet
- AP Classroom College Board login
- CodeHS.org login

UNIT 3 OVERVIEW

Content Area: Mathematics

Unit Title: Algorithms and Programming (Variables & Assignments, Data Abstraction, Mathematical Expressions, Strings and Boolean Expressions)

Target Course/Grade Level: Advanced Placement Computer Science Principles/Grades 10-12

Unit Summary: This unit introduces students to the basics of Python, including variables, user input, data types, and code comments. This provides students with an entirely new dynamic when designing and solving programming tasks. In data abstraction, programmers use a list as a representation of multiple pieces of data together. This unit focuses on how the use of data abstraction manages complexity in program code. Students will focus on evaluating mathematical expressions and boolean expressions to a single value. Mathematical operators include addition, subtraction, multiplication, division, and modulus. Implement logical operators like AND, NOT, and OR. Boolean expressions involve using relational operators: =, ≠, >, <, ≥, and ≤. Manipulate strings using concatenation and various other string functions.

Approximate Length of Unit: 4 weeks

LEARNING TARGETS

NJ Student Learning Standards:

8.1.12.AP.1 Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.7 Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
8.1.12.AP.8 Evaluate and refine computational artifacts to make them more usable and accessible.
8.1.12.AP.9 Collaboratively document and present design decisions in the development of complex programs.
8.1.12.AP.2 Create generalized computational solutions using collections instead of repeatedly using simple variables.

College Board Standards:

AAP-1.A Represent a value with a variable.
AAP-1.B Determine the value of a variable as a result of an assignment
AAP-1.C Represent a list or string using a variable.
AAP-1.D For data abstraction:
- a. Develop data abstraction using lists to store multiple elements.
- b. Explain how the use of data abstraction manages complexity in program code.

AAP-2.A Express an algorithm that uses sequencing without using a programming language
AAP-2.B Represent a step-by-step algorithmic process using sequential code statements.
AAP-2.C Evaluate expressions that use arithmetic operators**.**
AAP-2.D Evaluate expressions that manipulate strings**.**
AAP-2.E For relationships between two variables, expressions, or values:
- a. Write expressions using relational operators.

b. Evaluate expressions that use relational operators

AAP-2.F For relationships between Boolean values:

a. Write expressions using logical operators.

b. Evaluate expressions that use logic operators.

Career Readiness, Life Literacies, and Key Skills:

9.3.IT‑PRG.4 Demonstrate the effective use of software development tools to develop software applications.

9.3.IT‑PRG.6 Program a computer application using the appropriate programming language.

9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice.

9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving.

Interdisciplinary Connections and Standards:

ELA

RI.CI.11–12.2 Determine two or more central ideas of an informational text and analyze how they are developed and refined over the course of a text, including how they interact and build on one another to provide a complex account or analysis; provide an objective summary of the text.

RI.MF.11–12.6 Synthesize complex information across multiple sources and formats to develop ideas, resolve conflicting information, or develop an interpretation that goes beyond explicit text information (e.g., express a personal point of view, new interpretation of the concept).

SL.II.11–12.2 Integrate multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) in order to make informed decisions and solve problems, evaluating the credibility and accuracy of each source and noting any discrepancies among the data.

Mathematics

A.SSE.A.1 Interpret expressions that represent a quantity in terms of its context.

a. Interpret parts of an expression, such as terms, factors, and coefficients.

b. Interpret complicated expressions by viewing one or more of their parts as a single entity.

A.SSE.A.2 Use the structure of an expression to identify ways to rewrite it.

A.SSE.B.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.

a. Factor a quadratic expression to reveal the zeros of the function it defines.

b. Complete the square in a quadratic expression to reveal the max or min value of the function it defines.

c. Use the properties of exponents to transform expressions for exponential functions.

A.SSE.B.4 Derive and/or explain the formula for the sum of a finite geometric series (when the common ratio is not 1), and use the formula to solve problems.

Unit Understandings:

Students will understand that…

- 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.

Unit Essential Questions:

- How can we store data in a program to solve problems?
- What might happen if you completed the steps in your regular morning routine to get ready and go to school in a different order? How might the reordering affect the decisions you make each morning?

Knowledge and Skills:

Students will know...

- A variable is an abstraction inside a program that can hold a value. Each variable has associated data storage that represents one value at a time, but that value can be a list or other collection that in turn contains multiple values.
- Using meaningful variable names helps with the readability of program code and understanding of what values are represented by the variables.
- Some programming languages provide types to represent data, which are referenced using variables. These types include numbers, Booleans, lists, and strings.
- The assignment operator allows a program to change the value represented by a variable.
- The value stored in a variable will be the most recent value assigned.
- A list is an ordered sequence of elements and is used to create abstract data.
- An element is an individual value in a list that is assigned a unique index.
- An index is a common method for referencing the elements in a list or string using natural numbers.
- Data abstraction provides a separation between the abstract properties of a data type and the concrete details of its representation.
- Data abstractions manage complexity in programs by giving a collection of data a name without referencing the specific details of the representation.
- Developing a data abstraction to implement in a program can result in a program that is easier to develop and maintain.
- An algorithm is a finite set of instructions that accomplish a specific task.
- Every algorithm can be constructed using combinations of sequencing, selection, and iteration.
- Sequencing is the application of each step of an algorithm in the order in which the code statements are given.
- A code statement is a part of program code that expresses an action to be carried out.
- An expression can consist of a value, a variable, an operator, or a procedure call that returns a value.
- The evaluation of expressions follows a set order of operations defined by the programming language.
- Sequential statements execute in the order they appear in the code segment.
- Arithmetic operators are part of most programming languages and include addition, subtraction, multiplication, division, and modulus operators.
- The order of operations used in mathematics applies when evaluating expressions.
- String concatenation joins together two or more strings end-to-end to make a new string.
- A substring is part of an existing string.
- A Boolean value is either true or false.
- The operand for a logical operator is either a Boolean expression or a single Boolean value.

Students will be able to...

- Generalize data sources through variables.
- Determine the result of code segments.
- Use abstraction to manage complexity in a program.
- Explain how abstraction manages complexity.
- Represent algorithmic processes without using a programming language.
- Implement and apply an algorithm.
- Evaluate expressions that use arithmetic operators.

EVIDENCE OF LEARNING

Assessment:
What evidence will be collected and deemed acceptable to show that students truly "understand"?

- End of Unit Common Assessment - See folder for assessment links.
- CodeHS.com
- Lesson exercises
- Class participation
- Class discussion

Learning Activities:
What differentiated learning experiences and instruction will enable all students to achieve the desired results?

- Group programming projects
- Teacher demonstrations over Screen Share software
- Reinforcement worksheets and extra practice
- Paired-programming challenges
- Individualized student-to-teacher code demonstrations
- Sharing ideas and classroom discussions
- Sample Activity:
    - **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.

RESOURCES

Teacher Resources:

- Demonstrations of worked-out solutions from CodeHS.com
- Teacher designed worksheets
- CodeHS.com lesson exercises and videos
- College Board AP Classroom Resources and Quizzes

Equipment Needed:

- Classroom Computers/Chromebooks
- Access to high-speed internet
- AP Classroom College Board login
- CodeHS.org login

UNIT 4 OVERVIEW

Content Area: Mathematics

Unit Title: Algorithms and Programming (Conditionals, Nested Conditionals and Iteration)

Target Course/Grade Level: Advanced Placement Computer Science Principles/Grades 10-12

Unit Summary: In this unit, students will learn how to create the various conditional statements in Python and to create the various looping statements. With the introduction to Booleans, Logical Operators, Comparison Operators, and conditional IF-ELSE statements, students can design complex algorithms to solve problems. These new topics and concepts play an integral role in the functionality of many of the looping statements that students will use to create complex applications. Students will learn how to create the various control statements like FOR loops and WHILE loops students can design complex algorithms to solve problems. These new topics and concepts play an integral role in the functionality of many of the looping statements that students will use to create complex applications.

Approximate Length of Unit: 4 weeks

LEARNING TARGETS

NJ Student Learning Standards:

8.1.12.AP.1 Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.3 Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.4 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
8.1.12.AP.7 Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
8.1.12.AP.8 Evaluate and refine computational artifacts to make them more usable and accessible.
8.1.12.AP.9 Collaboratively document and present design decisions in the development of complex programs.
8.2.12.ED.1 Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.
8.2.12.ED.4 Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.
8.1.12.CS.4 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
8.2.12.NT.2 Redesign an existing product to improve form or function.

College Board Standards:

AAP-2.G Express an algorithm that uses selection without using a programming language.
AAP-2.H For selection:

a. Write conditional statements.

b. Determine the result of conditional statements.

AAP-2.I For nested selection:

a. Write nested conditional statements.

b. Determine the result of nested conditional statements.

AAP-2.J Express an algorithm that uses iteration without using a programming language.

AAP-2.K For iteration:

a. Write iteration statements.

b. Determine the result or side effect of iteration statements.

Career Readiness, Life Literacies, and Key Skills:

9.3.IT-PRG.1 Analyze customer software needs and requirements.

9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications.

9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.

9.3.IT-PRG.6 Program a computer application using the appropriate programming language.

9.4.12.CI.1 Demonstrate the ability to reflect, analyze, and use creative skills and ideas.

9.4.12.IML.3 Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions.

9.4.12.TL.1 Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.

9.4.12.TL.4 Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.

9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice.

9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving.

Interdisciplinary Connections and Standards:

ELA

RI.CI.11–12.2 Determine two or more central ideas of an informational text and analyze how they are developed and refined over the course of a text, including how they interact and build on one another to provide a complex account or analysis; provide an objective summary of the text.

RI.MF.11–12.6 Synthesize complex information across multiple sources and formats to develop ideas, resolve conflicting information, or develop an interpretation that goes beyond explicit text information (e.g., express a personal point of view, new interpretation of the concept).

SL.II.11–12.2 Integrate multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) in order to make informed decisions and solve problems, evaluating the credibility and accuracy of each source and noting any discrepancies among the data.

Mathematics

A-SSE.A.1 Interpret expressions that represent a quantity in terms of its context.

a. Interpret parts of an expression, such as terms, factors, and coefficients.

b. Interpret complicated expressions by viewing one or more of their parts as a single entity.

A-SSE.A.2 Use the structure of an expression to identify ways to rewrite it.

A-SSE.B.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.

a. Factor a quadratic expression to reveal the zeros of the function it defines.

b. Complete the square in a quadratic expression to reveal the max or min value of the function it defines.

c. Use the properties of exponents to transform expressions for exponential functions.

A-SSE.B.4 Derive and/or explain the formula for the sum of a finite geometric series (when the common ratio is not 1), and use the formula to solve problems.

Unit Understandings:
Students will understand that…

- 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.

Unit Essential Questions:

- What might happen if you completed the steps in your regular morning routine to get ready and go to school in a different order? How might the reordering affect the decisions you make each morning?
- Why are IF statements useful?
- What is the need for using control structures within control structures, which are referred to as nested control structures?
- How do IF statements control the flow of the program?

Knowledge and Skills:
Students will know...

- Selection determines which parts of an algorithm are executed based on a condition being true or false.
- Conditional Statements, or "if-statements", affect the sequential flow of control by executing different statements based on the value of a Boolean expression.
- Nested conditional statements consist of conditional statements within conditional statements.
- Iteration is a repeating portion of an algorithm. Iteration repeats a specified number of times or until a given condition is met.
- Iteration statements change the sequential flow of control by repeating a set of statements zero or more times until the stopping condition is met.
- In REPEAT UNTIL(condition) iteration, an infinite loop occurs when the ending condition will never evaluate to true.
- In REPEAT UNTIL(condition) iteration, if the conditional evaluates to true initially, the loop body is not executed at all, due to the condition being checked before the loop.

Students will be able to...

- Represent algorithmic processes without using a programming language.
- Implement and apply an algorithm.
- Determine the result of code segments.

EVIDENCE OF LEARNING

Assessment:
What evidence will be collected and deemed acceptable to show that students truly "understand"?

- End of Unit Common Assessment - See folder for assessment links.
- CodeHS.com
- Lesson exercises
- Class participation
- Class discussion

Learning Activities:

What differentiated learning experiences and instruction will enable all students to achieve the desired results?

- Group programming projects
- Teacher demonstrations over Screen Share software
- Reinforcement worksheets and extra practice
- Paired-programming challenges
- Individualized student-to-teacher code demonstrations
- Sharing ideas and classroom discussions
- Sample Activity:
    - **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.

RESOURCES

Teacher Resources:

- Demonstrations of worked-out solutions from CodeHS.com
- Teacher designed worksheets
- CodeHS.com lesson exercises and videos
- College Board AP Classroom Resources and Quizzes

Equipment Needed:

- Classroom Computers/Chromebooks
- Access to high-speed internet
- AP Classroom College Board login
- CodeHS.org login

UNIT 5 OVERVIEW

Content Area: Mathematics

Unit Title: Algorithms and Programming (Developing Algorithms and Lists)

Target Course/Grade Level: Advanced Placement Computer Science Principles/Grades 10-12

Unit Summary: Algorithms can be created from an idea, by combining existing algorithms, or by modifying existing algorithms to find a solution to the problem.s. This big idea focuses on determining the efficiency of algorithms, as well as writing and implementing algorithms in a program. Data abstraction is introduced in this unit using Lists and Tuples. Students will learn that tuples are a heterogeneous, immutable data type that stores an ordered sequence of things that can be accessed using indices. In addition, students will learn that a list is a mutable, heterogeneous data type that stores an ordered sequence of things. Students will be able to edit lists and iterate through them. Various list methods will be implemented to solve complex problems.

Approximate Length of Unit: 4 weeks

LEARNING TARGETS

NJ Student Learning Standards:

8.1.12.AP.1 Design algorithms to solve computational problems using a combination of original and existing algorithms.

8.1.12.AP.3 Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.

8.1.12.AP.4 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.

8.1.12.AP.7 Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.

8.1.12.AP.8 Evaluate and refine computational artifacts to make them more usable and accessible.

8.1.12.AP.9 Collaboratively document and present design decisions in the development of complex programs.

8.2.12.ED.1 Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.

8.2.12.ED.4 Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.

8.1.12.CS.4 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

8.2.12.NT.2 Redesign an existing product to improve form or function.

College Board Standards:

AAP-2.L Compare multiple algorithms to determine if they yield the same side effect or result.

AAP-2.M For algorithms:

a. Create algorithms.

b. Combine and modify existing algorithms.

AAP-2.O For algorithms involving elements of a list:

a. Write iteration statements to traverse a list.

b. Determine the result of an algorithm that includes list traversals.

AAP-2.N For list operations:

a. Write expressions that use list indexing and list procedures.

b. Evaluate expressions that use list indexing and list procedures.

Career Readiness, Life Literacies, and Key Skills:

9.3.IT-PRG.1 Analyze customer software needs and requirements.

9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications.

9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.

9.3.IT-PRG.6 Program a computer application using the appropriate programming language.

9.4.12.CI.1 Demonstrate the ability to reflect, analyze, and use creative skills and ideas.

9.4.12.IML.3 Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions.

9.4.12.TL.1 Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.

9.4.12.TL.4 Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.

9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice.

9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving.

Interdisciplinary Connections and Standards:

ELA

RI.CI.11–12.2 Determine two or more central ideas of an informational text and analyze how they are developed and refined over the course of a text, including how they interact and build on one another to provide a complex account or analysis; provide an objective summary of the text.

RI.MF.11–12.6 Synthesize complex information across multiple sources and formats to develop ideas, resolve conflicting information, or develop an interpretation that goes beyond explicit text information (e.g., express a personal point of view, new interpretation of the concept).

SL.II.11–12.2 Integrate multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) in order to make informed decisions and solve problems, evaluating the credibility and accuracy of each source and noting any discrepancies among the data.

Mathematics

A-SSE.A.1 Interpret expressions that represent a quantity in terms of its context.

a. Interpret parts of an expression, such as terms, factors, and coefficients.

b. Interpret complicated expressions by viewing one or more of their parts as a single entity.

A-SSE.A.2 Use the structure of an expression to identify ways to rewrite it.

A-SSE.B.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.

a. Factor a quadratic expression to reveal the zeros of the function it defines.

b. Complete the square in a quadratic expression to reveal the max or min value of the function it defines.

c. Use the properties of exponents to transform expressions for exponential functions.

A-SSE.B.4 Derive and/or explain the formula for the sum of a finite geometric series (when the common ratio is not 1), and use the formula to solve problems.

Unit Understandings:
Students will understand that…

- 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.

Unit Essential Questions:

- What makes an algorithm efficient?
- Can we create different algorithms to solve the same problem?
- Why might we want to go through a list one item at a time?

Knowledge and Skills:
Students will know...

- Algorithms can be written in different ways and still accomplish the same tasks.
- Algorithms that appear similar can yield different effects or results.
- Some conditional statements can be written as equivalent Boolean expressions.
- Some Boolean expressions can be written as equivalent conditional statements.
- Different algorithms can be developed or used to solve the same problem.
- Algorithms can be created from an idea, by combining existing algorithms, or by modifying existing algorithms.
- Knowledge of existing algorithms can help in constructing new ones.
- Using existing correct algorithms as building blocks for constructing another algorithm has benefits such as reducing development time, reducing testing, and simplifying the identification of errors.
- The exam reference sheet provides basic operations on lists, including: accessing an element by index, assigning a value of an element of a list to a variable, assigning a value to an element of a list, inserting elements at a given index, adding elements to the end of the list, § removing elements, and determining the length of a list.
- List procedures are implemented in accordance with the syntax rules of the programming language.

Students will be able to...

- Evaluate solution options.
- Represent algorithmic processes without using a programming language.
- Implement and apply an algorithm.
- Determine the result of code segments.

EVIDENCE OF LEARNING

Assessment:
What evidence will be collected and deemed acceptable to show that students truly "understand"?

- End of Unit Common Assessment - See folder for assessment links.
- CodeHS.com
- Lesson exercises
- Class participation
- Class discussion

Learning Activities:

What differentiated learning experiences and instruction will enable all students to achieve the desired results?

- Group programming projects
- Teacher demonstrations over Screen Share software
- Reinforcement worksheets and extra practice
- Paired-programming challenges
- Individualized student-to-teacher code demonstrations
- Sharing ideas and classroom discussions

RESOURCES

Teacher Resources:

Demonstrations of worked-out solutions from CodeHS.com

- Teacher designed worksheets
- CodeHS.com lesson exercises and videos
- College Board AP Classroom Resources and Quizzes

Equipment Needed:

- Classroom Computers/Chromebooks
- Access to high-speed internet
- AP Classroom College Board login
- CodeHS.org login

UNIT 6 OVERVIEW

Content Area: Mathematics

Unit Title: Algorithms and Programming (Binary Search, Calling Procedures, and Developing Procedures)

Target Course/Grade Level: Advanced Placement Computer Science Principles/Grades 10-12

Unit Summary: This unit focuses on analyzing the binary search algorithm. The binary search algorithm starts in the middle of a sorted data set of numbers and eliminates half of the data; this process repeats until the desired value is found or all elements have been eliminated. Creating an efficient search algorithm leads to implementing complex search engines. Students learn about Functions and parameters in the context of Python. Procedures let us break our program into different parts that we can organize and reuse however we like. Functions are the main building block of complex Python programs. We dive deeper into the concept of functions by exploring how to use parameters and understand the difference between a local and global variable. Students will learn what parameters are, and how to send them along while calling a function. Students explore functions with return values and deepen their understanding of and ability to use functions. Students explore Python's way of handling errors with exceptions.

Approximate Length of Unit: 4 weeks

LEARNING TARGETS

NJ Student Learning Standards:

8.1.12.AP.1 Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.2 Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3 Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.4 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
8.1.12.AP.7 Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
8.1.12.AP.8 Evaluate and refine computational artifacts to make them more usable and accessible.
8.1.12.AP.9 Collaboratively document and present design decisions in the development of complex programs.
8.2.12.ED.1 Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.
8.2.12.ED.4 Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.
8.1.12.CS.4 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
8.2.12.NT.2 Redesign an existing product to improve form or function.

College Board Standards:

AAP-2.P For binary search algorithms:

a. Determine the number of iterations required to find a value in a dataset.

b. Explain the requirements necessary to complete a binary search.

AAP-3.A For procedure calls:

a. Write statements to call procedures.

b. Determine the result or effect of procedure call.

AAP-3.B Explain how the use of procedural abstraction manages complexity in a program.

AAP-3.C Develop procedural abstractions to manage complexity in a program by writing procedures.

Career Readiness, Life Literacies, and Key Skills:

9.3.IT-PRG.1 Analyze customer software needs and requirements.

9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications.

9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.

9.3.IT-PRG.6 Program a computer application using the appropriate programming language.

9.4.12.CI.1 Demonstrate the ability to reflect, analyze, and use creative skills and ideas.

9.4.12.IML.3 Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions.

9.4.12.TL.1 Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.

9.4.12.TL.4 Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.

9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice.

9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving.

Interdisciplinary Connections and Standards:

ELA

RI.CI.11–12.2 Determine two or more central ideas of an informational text and analyze how they are developed and refined over the course of a text, including how they interact and build on one another to provide a complex account or analysis; provide an objective summary of the text.

RI.MF.11–12.6 Synthesize complex information across multiple sources and formats to develop ideas, resolve conflicting information, or develop an interpretation that goes beyond explicit text information (e.g., express a personal point of view, new interpretation of the concept).

SL.II.11–12.2 Integrate multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) in order to make informed decisions and solve problems, evaluating the credibility and accuracy of each source and noting any discrepancies among the data.

Mathematics

A-SSE.A.1 Interpret expressions that represent a quantity in terms of its context.

a. Interpret parts of an expression, such as terms, factors, and coefficients.

b. Interpret complicated expressions by viewing one or more of their parts as a single entity.

A-SSE.A.2 Use the structure of an expression to identify ways to rewrite it.

A-SSE.B.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.

a. Factor a quadratic expression to reveal the zeros of the function it defines.

b. Complete the square in a quadratic expression to reveal the max or min value of the function it defines.

c. Use the properties of exponents to transform expressions for exponential functions.

A-SSE.B.4 Derive and/or explain the formula for the sum of a finite geometric series (when the common ratio is not 1), and use the formula to solve problems.

Unit Understandings:

Students will understand that…

- 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.

Unit Essential Questions:

- How do video games group the different actions for a player based on what key is pressed on the keyboard or controller? How do apps group different actions together based on user interaction, such as pressing buttons?
- Why are search algorithms important in computer science?
- Why is Binary search more efficient than sequential search?
- Why are procedures useful?

Knowledge and Skills:

Students will know...

- The binary search algorithm starts in the middle of a sorted data set of numbers and eliminates half of the data; this process repeats until the desired value is found or all elements have been eliminated.
- Data must be in sorted order to use the binary search algorithm.
- Binary search is often more efficient than sequential/linear search when applied to sorted data.
- A procedure is a named group of programming instructions that may have parameters and return values.
- Procedures are referred to by different names, such as method or function, depending on the programming language.
- Parameters are input variables of a procedure. Arguments specify the values of the parameters when a procedure is called.
- A procedure call interrupts the sequential execution of statements, causing the program to execute the statements within the procedure before continuing. Once the last statement in the procedure (or a return statement) has been executed, the flow of control is returned to the point immediately following where the procedure was called.
- One common type of abstraction is procedural abstraction, which provides a name for a process and allows a procedure to be used only knowing what it does, not how it does it.
- The subdivision of a computer program into separate subprograms is called modularity.
- Procedural abstraction allows a solution to a large problem to be based on the solutions of smaller subproblems. This is accomplished by creating procedures to solve each of the subproblems.
- Using parameters allows procedures to be generalized, enabling the procedures to be reused with a range of input values or arguments.

Students will be able to...

- Investigate the situation, context, or task.
- Evaluate solution options.
- Use abstraction to manage complexity in a program.
- Determine the result of code segments.
- Explain how abstraction manages complexity.

EVIDENCE OF LEARNING

Assessment:
What evidence will be collected and deemed acceptable to show that students truly "understand"?

- End of Unit Common Assessment - See folder for assessment links.
- CodeHS.com
- Lesson exercises
- Class participation
- Class discussion

Learning Activities:
What differentiated learning experiences and instruction will enable all students to achieve the desired results?

- Group programming projects
- Teacher demonstrations over Screen Share software
- Reinforcement worksheets and extra practice
- Paired-programming challenges
- Individualized student-to-teacher code demonstrations
- Sharing ideas and classroom discussions
- Sample Activity:
    - **Marking the Text:** Provide students with a program code that draws a square of side length 10 and a separate set of program codes that uses side length 100. Ask students to mark up thesetsofcodetoidentifywheretheyaredifferentandtocreateageneralization by using parameters. Ask them to write a procedure that uses parameters to draw a square of any size.

RESOURCES

Teacher Resources:

- Demonstrations of worked-out solutions from CodeHS.com
- Teacher designed worksheets
- CodeHS.com lesson exercises and videos
- College Board AP Classroom Resources and Quizzes

Equipment Needed:

- Classroom Computers/Chromebooks
- Access to high-speed internet
- AP Classroom College Board login
- CodeHS.org login

UNIT 7 OVERVIEW

Content Area: Mathematics

Unit Title: Algorithms and Programming (Libraries, Random Values, Simulations, Algorithmic Efficiency and Undecidable Problems)

Target Course/Grade Level: Advanced Placement Computer Science Principles/Grades 10-12

Unit Summary: This unit focuses on using software libraries that contain procedures that can be used in creating new programs. The random library will be used to showcase how API libraries can be used to create complex programs. This unit focuses on developing computer programs that simulate and mimic real-world events with the purpose of drawing inferences, allowing investigation of a phenomenon without the constraints of the real world. Investigate computational resources required to determine the efficiency of any given algorithm. This unit will focus on explaining the existence of undecidable problems in computer science.

Approximate Length of Unit: 4 weeks

LEARNING TARGETS

NJ Student Learning Standards:

8.1.12.AP.1 Design algorithms to solve computational problems using a combination of original and existing algorithms.

8.1.12.AP.2 Create generalized computational solutions using collections instead of repeatedly using simple variables.

8.1.12.AP.3 Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.

8.1.12.AP.4 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.

8.1.12.AP.5 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

8.1.12.AP.6 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

8.1.12.AP.7 Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.

8.1.12.AP.8 Evaluate and refine computational artifacts to make them more usable and accessible.

8.1.12.AP.9 Collaboratively document and present design decisions in the development of complex programs.

8.2.12.ED.1 Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.

8.2.12.ED.4 Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.

8.1.12.CS.4 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

8.2.12.NT.2 Redesign an existing product to improve form or function.

College Board Standards:

AAP-3.D Select appropriate libraries or existing code segments to use in creating new programs.

AAP-3.E For generating random values:

a. Write expressions to generate possible values.
b. Evaluate expressions to determine the possible results.

AAP-3.F For simulations:

a. Explain how computers can be used to represent real-world phenomena or outcomes.
b. Compare simulations with real-world contexts.

AAP-4.A For determining the efficiency of an algorithm:

a. Explain the difference between algorithms that run in reasonable time and those that do not.
b. Identify situations where a heuristic solution may be more appropriate.

AAP-4.B Explain the existence of undecidable problems in computer science.

Career Readiness, Life Literacies, and Key Skills:

9.3.IT-PRG.1 Analyze customer software needs and requirements.

9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications.

9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.

9.3.IT-PRG.6 Program a computer application using the appropriate programming language.

9.4.12.CI.1 Demonstrate the ability to reflect, analyze, and use creative skills and ideas.

9.4.12.IML.3 Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions.

9.4.12.TL.1 Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.

9.4.12.TL.4 Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.

9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice.

9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving.

Interdisciplinary Connections and Standards:

ELA

RI.CI.11–12.2 Determine two or more central ideas of an informational text and analyze how they are developed and refined over the course of a text, including how they interact and build on one another to provide a complex account or analysis; provide an objective summary of the text.

RI.MF.11–12.6 Synthesize complex information across multiple sources and formats to develop ideas, resolve conflicting information, or develop an interpretation that goes beyond explicit text information (e.g., express a personal point of view, new interpretation of the concept).

SL.II.11–12.2 Integrate multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) in order to make informed decisions and solve problems, evaluating the credibility and accuracy of each source and noting any discrepancies among the data.

Mathematics

A-SSE.A.1 Interpret expressions that represent a quantity in terms of its context.

a. Interpret parts of an expression, such as terms, factors, and coefficients.
b. Interpret complicated expressions by viewing one or more of their parts as a single entity.

A-SSE.A.2 Use the structure of an expression to identify ways to rewrite it.

A-SSE.B.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.

a. Factor a quadratic expression to reveal the zeros of the function it defines.
b. Complete the square in a quadratic expression to reveal the max or min value of the function it defines.

c. Use the properties of exponents to transform expressions for exponential functions.

A-SSE.B.4 Derive and/or explain the formula for the sum of a finite geometric series (when the common ratio is not 1), and use the formula to solve problems.

Unit Understandings:

Students will understand that…

- 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.

Unit Essential Questions:

- How do video games group the different actions for a player based on what key is pressed on the keyboard or controller? How do apps group different actions together based on user interaction, such as pressing buttons?
- What types of problems can be solved more easily with a computer, and what types can be solved more easily without a computer? Why?
- Why is it useful to be able to create random values in our programs?
- How can we compare the efficiency of different algorithms?
- How can we investigate an undecidable problem?

Knowledge and Skills:

Students will know...

- A software library contains procedures that may be used in creating new programs.
- Existing code segments can come from internal or external sources, such as libraries or previously written code.
- The use of libraries simplifies the task of creating complex programs.
- Documentation for an API/library is necessary for understanding the behaviors provided by the API/library and how to use them.
- Using random number generation in a program means each execution may produce a different result.
- Simulations are abstractions of more complex objects or phenomena for a specific purpose
- A simulation is a representation that uses varying sets of value stores to reflect the changing state of a phenomenon.
- Simulations often mimic real-world events with the purpose of drawing inferences, allowing investigation of a phenomenon without the constraints of the real world.
- The process of developing an abstract simulation involves removing specific details or simplifying functionality.
- Simulations can contain bias derived from the choices of real-world elements that were included or excluded.
- Simulations are most useful when real-world events are impractical for experiments.
- Simulations facilitate the formulation and refinement of hypotheses related to the objects or phenomena under consideration.
- A problem is a general description of a task that can (or cannot) be solved algorithmically. An instance of a problem also includes specific input.
- A decision problem is a problem with a yes/no answer.
- Efficiency is an estimation of the amount of computational resources used by an algorithm.
- An algorithm’s efficiency is determined through formal or mathematical reasoning.

- Analgorithm'sefficiencycanbeinformally measured by determining the number of times a statement or group of statements executes.
- Different correct algorithms for the same problem can have different efficiencies.
- Some problems cannot be solved in a reasonable amount of time because there is no efficient algorithm for solving them.
- A heuristic is an approach to a problem that produces a solution that is not guaranteed to be optimal but may be used when techniques that are guaranteed to always find an optimal solution are impractical.
- A decidable problem is a decision problem for which an algorithm can be written to produce a correct output for all inputs.
- An undecidable problem is one for which no algorithm can be constructed that is always capable of providing a correct yes-or-no answer.
- An undecidable problem may have some instances that have an algorithmic solution, but there is no algorithmic solution that could solve all instances of the problem.

Students will be able to...

- Implement and apply an algorithm.
- Determine the result of code segments.
- Investigate the situation, context, or task.
- Evaluate solution options.

EVIDENCE OF LEARNING

Assessment:

What evidence will be collected and deemed acceptable to show that students truly "understand"?

- End of Unit Common Assessment - See folder for assessment links.
- CodeHS.com
- Lesson exercises
- Class participation
- Class discussion

Learning Activities:

What differentiated learning experiences and instruction will enable all students to achieve the desired results?

- Group programming projects
- Teacher demonstrations over Screen Share software
- Reinforcement worksheets and extra practice
- Paired-programming challenges
- Individualized student-to-teacher code demonstrations
- Sharing ideas and classroom discussions
- Sample Activity:
    - **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.

RESOURCES

Teacher Resources:

- Demonstrations of worked-out solutions from CodeHS.com
- Teacher designed worksheets
- CodeHS.com lesson exercises and videos
- College Board AP Classroom Resources and Quizzes

Equipment Needed:

- Classroom Computers/Chromebooks
- Access to high-speed internet
- AP Classroom College Board login
- CodeHS.org login

UNIT 8 OVERVIEW

Content Area: Mathematics

Unit Title: Computer Systems and Networks

Target Course/Grade Level: Advanced Placement Computer Science Principles/Grades 10-12

Unit Summary: Students will learn about how information is transmitted on the Internet and about the safeguards that have been put in place to keep this system from breaking down. In addition, students will learn the effect that dividing tasks across multiple computing devices can have on the speed at which processes can occur. Students will be presented with scenarios for how information could be passed via the Internet along with illustrations of interconnected computers in given networks. Students will be asked to select which choice best explains how information is passed through these networks from one computing device to another or how designing systems to include redundancy helps make them fault-tolerant. Redundant routing options help ensure that the Internet is more reliable and stable. Using kinesthetic learning techniques to simulate the connections and routers in a system can help students make this abstract idea more concrete.

Approximate Length of Unit: 4 weeks

LEARNING TARGETS

NJ Student Learning Standards:

8.1.12.AP.1 Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.NI.1 Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.
8.1.12.NI.2 Evaluate security measures to address various common security threats.
8.1.12.NI.3 Explain how the needs of users and the sensitivity of data determine the level of security implemented.
8.1.12.NI.4 Explain how decisions on methods to protect data are influenced by whether the data is at rest, in transit, or in use.
8.1.12.AP.8 Evaluate and refine computational artifacts to make them more usable and accessible.
8.1.12.AP.9 Collaboratively document and present design decisions in the development of complex programs.
8.2.12.ED.1 Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.
8.2.12.ED.4 Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.
8.1.12.CS.4 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
8.2.12.NT.2 Redesign an existing product to improve form or function.

College Board Standards:

CSN-1.A Explain how computing devices work together in a network.
CSN-1.B Explain how the Internet works
CSN-1.C Explain how data are sent through the Internet via packets.
CSN-1.D Describe the differences between the Internet and the World Wide Web.
CSN-1.E For fault-tolerant systems, like the Internet:

a. Describe the benefits of fault tolerance.
b. Explain how a given system is fault-tolerant.
c. Identify vulnerabilities to failure in a system.

CSN-2.A For sequential, parallel, and distributed computing:

a. Compare problem solutions.
b. Determine the efficiency of solutions.

CSN-2.B Describe benefits and challenges of parallel and distributed computing.

Career Readiness, Life Literacies, and Key Skills:

9.3.IT-NET.1 Analyze customer or organizational network system needs and requirements.
9.3.IT-NET.2 Analyze wired and wireless network systems to determine if they meet specifications (e.g., IEEE, power and security).
9.3.IT-NET.3 Design a network system using technologies, tools and standards.
9.3.IT-NET.4 Perform network system installation and configuration.
9.3.IT-NET.5 Perform network administration, monitoring and support to maintain a network system.
9.4.12.CI.1 Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.IML.3 Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions.
9.4.12.TL.1 Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.
9.4.12.TL.4 Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.
9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving.

Interdisciplinary Connections and Standards:

ELA

RI.CI.11–12.2 Determine two or more central ideas of an informational text and analyze how they are developed and refined over the course of a text, including how they interact and build on one another to provide a complex account or analysis; provide an objective summary of the text.
RI.MF.11–12.6 Synthesize complex information across multiple sources and formats to develop ideas, resolve conflicting information, or develop an interpretation that goes beyond explicit text information (e.g., express a personal point of view, new interpretation of the concept).
SL.II.11–12.2 Integrate multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) in order to make informed decisions and solve problems, evaluating the credibility and accuracy of each source and noting any discrepancies among the data.

Mathematics

A-SSE.A.1 Interpret expressions that represent a quantity in terms of its context.

a. Interpret parts of an expression, such as terms, factors, and coefficients.
b. Interpret complicated expressions by viewing one or more of their parts as a single entity.

A-SSE.A.2 Use the structure of an expression to identify ways to rewrite it.
A-SSE.B.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.

a. Factor a quadratic expression to reveal the zeros of the function it defines.
b. Complete the square in a quadratic expression to reveal the max or min value of the function it

defines.

c. Use the properties of exponents to transform expressions for exponential functions.

A-SSE.B.4 Derive and/or explain the formula for the sum of a finite geometric series (when the common ratio is not 1), and use the formula to solve problems.

Unit Understandings:

Students will understand that…

- 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.

Unit Essential Questions:

- Why are long text messages sometimes delivered out of order?
- When an Internet service outage occurs in a different part of your town or city, how are you still able to access the Internet?
- What are the benefits of dividing tasks among network hubs?
- Is there a point where adding another group member would not make completing the task faster? Why?

Knowledge and Skills:

Students will know...

- A computing device is a physical artifact that can run a program.
- A computing system is a group of computing devices and programs working together for a common purpose.
- A computer network is a group of interconnected computing devices capable of sending or receiving data.
- A computer network is a type of computing system.
- Routing is the process of finding a path from sender to receiver.
- The bandwidth of a computer network is the maximum amount of data that can be sent in a fixed amount of time.
- Bandwidth is usually measured in bits per second.
- The Internet is a computer network consisting of interconnected networks that use standardized, open (nonproprietary) communication protocols.
- Access to the Internet depends on the ability to connect a computing device to an Internet-connected device.
- The protocols used on the Internet are open, which allows users to easily connect additional computing devices to the Internet.
- Routing on the Internet is usually dynamic; it is not specified in advance.
- The scalability of a system is the capacity for the system to change in size and scale to meet new demands.
- Information is passed through the Internet as a data stream. Data streams contain chunks of data, which are encapsulated in packets.
- Packets contain a chunk of data and metadata used for routing the packet between the origin and the destination on the Internet, as well as for data reassembly.
- Packets may arrive at the destination in order, out of order, or not at all.
- IP, TCP, and UDP are common protocols used on the Internet.
- The World Wide Web is a system of linked pages, programs, and files.
- HTTP is a protocol used by the World Wide Web.
- The Internet has been engineered to be fault tolerant, with abstractions for routing and transmitting data.

- Redundancy is the inclusion of extra components that can be used to mitigate the failure of a system if other components fail.
- One way to accomplish network redundancy is by having more than one path between any two connected devices.
- When a system can support failures and still continue to function, it is called fault-tolerant. This is important because elements of complex systems fail at unexpected times, often in groups, and fault tolerance allows users to continue to use the network.
- The redundancy of routing options between two points increases the reliability of the Internet and helps it scale to more devices and more people.
- Sequential computing is a computational model in which operations are performed in order one at a time.
- Parallel computing is a computational model where the program is broken into multiple smaller sequential computing operations, some of which are performed simultaneously.
- Distributed computing is a computational model in which multiple devices are used to run a program.
- Comparing the efficiency of solutions can be done by comparing the time it takes them to perform the same task.
- A sequential solution takes as long as the sum of all of its steps.
- A parallel computing solution takes as long as its sequential tasks plus the longest of its parallel tasks.
- The "speedup" of a parallel solution is measured in the time it took to complete the task sequentially divided by the time it took to complete the task when done in parallel.
- Parallel computing consists of a parallel portion and a sequential portion.
- Solutions that use parallel computing can scale more effectively than solutions that use sequential computing.
- Distributed computing allows problems to be solved that could not be solved on a single computer because of either the processing time or storage needs involved.
- Distributed computing allows much larger problems to be solved quicker than they could be solved using a single computer.
- When increasing the use of parallel computing in a solution, the efficiency of the solution is still limited by the sequential portion. This means that at some point, adding parallel portions will no longer meaningfully increase efficiency.

Students will be able to...

- Explain how computing systems work.
- Evaluate solution options.
- Explain how the internet works.
- Explain what is fault tolerance.

EVIDENCE OF LEARNING

Assessment:

What evidence will be collected and deemed acceptable to show that students truly "understand"?

- End of Unit Common Assessment - See folder for assessment links.
- CodeHS.com
- Lesson exercises
- Class participation

- Class discussion

Learning Activities:
What differentiated learning experiences and instruction will enable all students to achieve the desired results?

- Group programming projects
- Teacher demonstrations over Screen Share software
- Reinforcement worksheets and extra practice
- Paired-programming challenges
- Individualized student-to-teacher code demonstrations
- Sharing ideas and classroom discussions
- Sample Activities:
    - **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 the set. Then show students how to determine the efficiency of each model to check if their predictions were correct.

RESOURCES

Teacher Resources:

- Demonstrations of worked-out solutions from CodeHS.com
- Teacher designed worksheets
- CodeHS.com lesson exercises and videos
- College Board AP Classroom Resources and Quizzes

Equipment Needed:

- Classroom Computers/Chromebooks
- Access to high-speed internet
- AP Classroom College Board login
- CodeHS.org login

UNIT 9 OVERVIEW

Content Area: Mathematics

Unit Title: Impact of Computing

Target Course/Grade Level: Advanced Placement Computer Science Principles/Grades 10-12

Unit Summary: In this big idea, students explore these effects, the legal and ethical concerns that come with programs, and the responsibilities of programmers. When using computing innovations and transmitting information via the Internet, students should be aware of the risk of sharing personally identifiable information about themselves, such as their age or address, and actively take steps to keep this information safe. This big idea can be integrated throughout the course and works well with the Creative Development, Data, and Computing Systems and Networks big ideas. Investigating the impact of existing computing innovations can help students avoid unintentional negative effects of their own innovations.

Approximate Length of Unit: 2 weeks

LEARNING TARGETS

NJ Student Learning Standards:

8.1.12.AP.1 Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.IC.1 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
8.1.12.IC.2 Test and refine computational artifacts to reduce bias and equity deficits.
8.1.12.IC.3 Predict the potential impacts and implications of emerging technologies on larger social, economic, and political structures, using evidence from credible sources.
8.2.12.EC.1 Analyze controversial technological issues and determine the degree to which individuals, businesses, and governments have an ethical role in decisions that are made.
8.2.12.EC.2 Assess the positive and negative impacts of emerging technologies on developing countries and evaluate how individuals, non-profit organizations, and governments have responded.
8.2.12.EC.3 Synthesize data, analyze trends, and draw conclusions regarding the effect of a technology on the individual, culture, society, and environment and share this information with the appropriate audience.
8.2.12.ETW.4 Research historical tensions between environmental and economic considerations as driven by human needs and wants in the development of a technological product and present the competing viewpoints.
8.2.12.ED.1 Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.
8.2.12.ED.4 Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.

College Board Standards:

IOC-1.A Explain how an effect of a computing innovation can be both beneficial and harmful.
IOC-1.B Explain how a computing innovation can have an impact beyond its intended purpose.
IOC-1.C Describe issues that contribute to the digital divide
IOC-1.D Explain how bias exists in computing innovations.
IOC-1.E Explain how people participate in problem-solving processes at scale.
IOC-1.F Explain how the use of computing can raise legal and ethical concerns
IOC-2.A Describe the risks to privacy from collecting and storing personal data on a computer system
IOC-2.B Explain how computing resources can be protected and can be misused.
IOC-2.C Explain how unauthorized access to computing resources is gained

Career Readiness, Life Literacies, and Key Skills:

9.3.IT-NET.1 Analyze customer or organizational network system needs and requirements.
9.3.IT-WD.10 Comply with intellectual property laws, copyright laws and ethical practices when creating Web/digital communications.
9.3.IT.10 Describe the use of computer forensics to prevent and solve information technology crimes and security breaches.
9.4.12.TL.4 Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.
9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving.
9.4.12.DC.3 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.
9.4.12.DC.2 Compare and contrast international differences in copyright laws and ethics.

Interdisciplinary Connections and Standards:

ELA

RI.CI.11–12.2 Determine two or more central ideas of an informational text and analyze how they are developed and refined over the course of a text, including how they interact and build on one another to provide a complex account or analysis; provide an objective summary of the text.
RI.MF.11–12.6 Synthesize complex information across multiple sources and formats to develop ideas, resolve conflicting information, or develop an interpretation that goes beyond explicit text information (e.g., express a personal point of view, new interpretation of the concept).
SL.II.11–12.2 Integrate multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) in order to make informed decisions and solve problems, evaluating the credibility and accuracy of each source and noting any discrepancies among the data.

Mathematics

A-SSE.A.1 Interpret expressions that represent a quantity in terms of its context.
a. Interpret parts of an expression, such as terms, factors, and coefficients.
b. Interpret complicated expressions by viewing one or more of their parts as a single entity.
A-SSE.A.2 Use the structure of an expression to identify ways to rewrite it.
A-SSE.B.3 Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.
a. Factor a quadratic expression to reveal the zeros of the function it defines.
b. Complete the square in a quadratic expression to reveal the max or min value of the function it defines.
c. Use the properties of exponents to transform expressions for exponential functions.
A-SSE.B.4 Derive and/or explain the formula for the sum of a finite geometric series (when the common ratio is not 1), and use the formula to solve problems.

Unit Understandings:
Students will understand that…

- While computing innovations are typically designed to achieve specific purpose, they may have unintended consequences.
- The use of computing innovations may involve risks to personal safety and identity.

Unit Essential Questions:

- Are innovators responsible for the harmful effects of their computing innovations, even if those effects were unintentional? Why or why not?
- What data is generated by smartphones, and what are they being used for?

Knowledge and Skills:
Students will know...

- People create computing innovations.
- Not every effect of a computing innovation is anticipated in advance.
- A single effect can be viewed as both beneficial and harmful by different people, or even by the same person.
- Advances in computing have generated and increased creativity in other fields, such as medicine, engineering, communications, and the arts.
- Some of the ways computing innovations can be used may have a harmful impact on society, the economy, or culture.
- Responsible programmers try to consider the unintended ways their computing innovations can be used and the potential beneficial and harmful effects of these new uses.
- Rapid sharing of a program or running a program with a large number of users can result in significant impacts beyond the intended purpose or control of the programmer.
- Internet access varies between socioeconomic, geographic, and demographic characteristics, as well as between countries.
- The“digital divide” refers to differing access to computing devices and the Internet, based on socioeconomic, geographic, or demographic characteristics.
- The digital divide can affect both groups and individuals.
- The digital divide raises issues of equity, access, and influence, both globally and locally
- Computing innovations can reflect existing human biases because of biases written into the algorithms or biases in the data used by the innovation.
- Programmers should take action to reduce bias in algorithms used for computing innovations as a way of combating existing human biases.
- Biases can be embedded at all levels of software development.
- Widespread access to information and public data facilitates the identification of problems, the development of solutions, and the dissemination of results.
- Citizen science is scientific research conducted in whole or part by distributed individuals, many of whom may not be scientists, who contribute relevant data to research using their own computing devices.
- Crowdsourcing is the practice of obtaining input or information from a large number of people via the Internet.
- Crowdsourcing offers new models for collaboration, such as connecting businesses or social causes with funding.
- Material created on a computer is the intellectual property of the creator or an organization
- Measures should be taken to safeguard intellectual property.
- The use of material created by someone else without permission and presented as one’s own is plagiarism and may have legal consequences.
- The use of material created by someone other than you should always be cited.

- Creative Commons, open source, and open access have enabled broad access to digital information.
- As with any technology or medium, using computing to harm individuals or groups of people raises legal and ethical concerns.
- Computing can play a role in social and political issues, which in turn often raises legal and ethical concerns.
- The digital divide raises ethical concerns around computing.
- Personally identifiable information (PII) is information about an individual that identifies, links, relates, or describes them.
- Websites can record and maintain a history of individuals who have viewed their pages.
- Search engines can record and maintain a history of searches made by users.
- Technology enables the collection, use, and exploitation of information about, by, and for individuals, groups, and institutions.
- Search engines can use search history to suggest websites or for targeted marketing.
- Disparate personal data, such as geolocation, cookies, and browsing history, can be aggregated to create knowledge about an individual.
- PII and other information placed online can be used to enhance a user's online experiences.
- PII can be used to stalk or steal the identity of a person or to aid in the planning of other criminal acts.
- Programs can collect your location and record where you have been, how you got there, and how long you were at a given location.
- Once information is placed online, it is difficult to delete
- A strong password is something that is easy for a user to remember but would be difficult for someone else to guess based on the knowledge of that user.
- Multi-factor authentication requires at least two steps to unlock protected information; each step adds a new layer of security that must be broken to gain unauthorized access.
- Encryption is the process of encoding data to prevent unauthorized access. Decryption is the process of decoding the data. Two common encryption approaches are: Symmetric key encryption and Public key encryption.
- Computer virus and malware scanning software can help protect a computing system against infection.
- Malware is software intended to damage a computing system or to take partial control over its operation.
- All real-world systems have errors or design flaws that can be exploited to compromise them. Regular software updates help fix errors that could compromise a computing system.
- Phishing is a technique that attempts to trick a user into providing personal information. That personal information can then be used to access sensitive online resources, such as bank accounts and emails.
- Keylogging is the use of a program to record every keystroke made by a computer user in order to gain fraudulent access to passwords and other confidential information.
- Data sent over public networks can be intercepted, analyzed, and modified. One Way that this can happen is through a rogue access point.
- A malicious link can be disguised on a web page or in an email message.
- Unsolicited emails, attachments, links, and forms in emails can be used to compromise the security of a computing system. These can come from unknown senders or from known senders whose security has been compromised.
- Untrustworthy (often free) downloads from freeware or shareware sites can contain malware.

Students will be able to...

- Describe the impact of a computing innovation.
- Evaluate the use of computing based on legal and ethical factors.

- Explain how collaboration affects the development of a solution.
- Describe the impact of gathering data.

EVIDENCE OF LEARNING

Assessment:
What evidence will be collected and deemed acceptable to show that students truly "understand"?

- End of Unit Common Assessment - See folder for assessment links.
- CodeHS.com
- Lesson exercises
- Class participation
- Class discussion

Learning Activities:
What differentiated learning experiences and instruction will enable all students to achieve the desired results?

- Group programming projects
- Teacher demonstrations over Screen Share software
- Reinforcement worksheets and extra practice
- Paired-programming challenges
- Individualized student-to-teacher code demonstrations
- Sharing ideas and classroom discussions
- Sample Activities:
    - **Marking the Text:** Provide students with an article that highlights both beneficial and harmful effects of 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.

RESOURCES

Teacher Resources:

- Demonstrations of worked-out solutions from CodeHS.com
- Teacher designed worksheets
- CodeHS.com lesson exercises and videos
- College Board AP Classroom Resources and Quizzes

Equipment Needed:

- Classroom Computers/Chromebooks
- Access to high-speed internet
- AP Classroom College Board login
- CodeHS.org login