

Summit Public Schools
Summit, New Jersey
Grade Level: 9-12 / Content Area: Mathematics

Overview:

The AP Computer Science: Principles course is designed to provide high school students with a broad overview of a number of computer science ideas including abstraction, collaboration and analysis. They will also create computational artifacts to demonstrate these abilities. Upon completion of this course students will have the necessary tools to create applications, investigate computing advances and discuss topics relevant to the modern, global computer science environment.

Students in this course have already completed an intro to programming course and have some experience programming in Java.

Prerequisite Course:

Intro to Java (Or department approval)

Suggested Timeline:

Each unit can start with a class discussion, but most of the time in this course should be spent with the students programming, or working with existing code. Within each unit 1-2 days may be spent with new material, with the rest of the time being spent with students working on projects, saving 2 days for a review and a test or quiz depending on the unit.

The following timeline is intended for a rotating-drop schedule where classes meet 3 times every 4 days for 58 minutes. Adjustments may be necessary depending on the timing of the AP Exam.

Marking Period 1	
Programing Basics	10 Days
Control Structures	6 Days
Methods and Parameters	6 Days
Basic Data Structures	8 Days
Marking Period 2	
Digital Information	10 Days
Practice Create Projects	10 Days
The Internet	10 Days
Marking Period 3	
Data	9 Days
The Impact of Computing	8 Days
Create Project	13 (Required minimum)
Marking Period 4	
AP Exam Review	10 Dyas
Final Project	20 Days

Unit 1: Programing Basics

Big Ideas: *Course Objectives/ Content Statement(s)*

Students will understand...

- What programs are made up of
- What an abstraction is
- How programs input and output data
- When to use control statements

Essential Questions

What provocative questions will foster inquiry, understanding, and transfer of learning?

- What is a program made up of?
- What are different ways a program can accept input, and produce output?
- What is an Abstraction?
- How can formatting and commenting code be used effectively?
- What are control statements? How can they be used to create patterns?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Every program needs code statements to run. Multiple statements can be combined to make a code segment, these can be contained in methods or functions that may have parameters.
- Any time a user gives the program information that is considered input, examples include a keyboard, mouse, or a touchscreen. Any time the program gives information to the user that is considered output, examples include information displayed on the screen, or sound.
- An abstraction is a way to reduce information in a program while maintaining the functionality. Functions or methods are an example of abstractions
- A code statement is intended to show that it is contained within another code segment (for example a function, method, control statement). Comments can be used as a planning tool before writing actual code, and as a way to explain to someone reading code what certain code segments do.
- If statements, and loops are types of control statements. They control when and how many times certain code segments are executed

Areas of Focus: Proficiencies
(New Jersey Student Learning Standards)

Topics

Students will

Suggested Activities:

- Write an effectively formatted program that can produce output
- Be able to provide examples of, and create abstractions
- Write a program containing basic if statements and loops
- Write a program that accepts user input

NJ Computer Science and Design Thinking Standards

- **8.1.12.CS.3** Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences.
- **8.1.12.IC.1** The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities.

Career-Ready Practices

CRP1: Act as a responsible and contributing citizen and employee.

CRP2: Apply appropriate academic and technical skills.

CRP3: Attend to personal health and financial well-being.

CRP4: Communicate clearly and effectively and with reason.

CRP5: Consider the environmental, social and economic impacts of decisions.

CRP6: Demonstrate creativity and innovation.

CRP7: Employ valid and reliable research strategies.

CRP8: Utilize critical thinking to make sense of problems and persevere in solving them.

CRP9: Model integrity, ethical leadership and effective management.

CRP10: Plan education and career paths aligned to personal goals.

CRP11: Use technology to enhance productivity.

CRP12: Work productively in teams while using cultural global competence.

- CodeHS Units 1-3 - Or teacher developed lessons covering:
 - Review control statements, code formatting
 - Basic JavaScript input and output
 - JavaScript graphics
- Hello World program in Java and JavaScript
- Differences between Java and JavaScript
 - Main method
 - variables
 - printing
- Introduce College board's programming language

Differentiation/Technology

Assessments

Differentiation strategies:

- Additional assignments or extensions to existing assignments for students who are moving quickly
- Partnering students who need support with students who can assist
- Collaboration on all programming projects is encouraged. Collaboration is required as part of the course for all students

Technology integration

- Online platforms (CodeHS e.g.)
- Desktop Java IDEs (IntelliJ, Eclipse, BlueJ e.g.)

- Graded programming projects
- 2-3 question pop quizzes
- End of unit CodeHS Quiz
- AP Exam style end of unit quiz

Unit 2: Control Structures

Big Ideas: *Course Objectives/ Content Statement(s)*

Students will understand...

- What a loop and if statement are, and how to use them
- How to write a program containing for loops
- How to write a program containing while loops
- How to write a program containing if statements
- How to select the appropriate control statement to meet a goal

Essential Questions

What provocative questions will foster inquiry, understanding, and transfer of learning?

- What is a boolean?
- What are logical and comparison operators?
- What is a control statement?
- How are control statements used in programs?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- A boolean is a value that can either be true or false
- Logical operators are operators that can be used with two booleans. The ones tested on the AP Exam are Not, Or, And. Comparison operators are used with numerical values and return booleans. The operators tested on the AP Exam are Less Than (Or Equal To), Greater Than (Or Equal To), Equal To, Not Equal To
- A control statement is a way to control when or if a code segment is executed, and how many times. Types of control statements are If, For, and While
- There are multiple ways control statements are used in programming. They can help the programmer save time by shortening code that would otherwise be repetitive, they can work with user input to make the program behave differently depending on what the user inputs, and they are necessary when the programmer is unsure of how many times a code segment will be executed

Areas of Focus: Proficiencies (New Jersey Student Learning Standards)

Students will

- Read code containing logical operators, and boolean expressions and decide the result or output

Topics/Lessons

Suggested Activities:

- CodeHS Unit 4 - Or teacher developed activities covering:
 - Booleans

- Write programs containing control statements
- Choose an appropriate control statement to complete a given task
- Combine multiple control statements to develop algorithms

NJ Computer Science and Design Thinking Standards

- **8.1.12.CS.3** Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences.
- **8.1.12.IC.1** The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities.
- **8.1.12.AP.3** Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.

Career-Ready Practices

CRP1: Act as a responsible and contributing citizen and employee.

CRP2: Apply appropriate academic and technical skills.

CRP3: Attend to personal health and financial well-being.

CRP4: Communicate clearly and effectively and with reason.

CRP5: Consider the environmental, social and economic impacts of decisions.

CRP6: Demonstrate creativity and innovation.

CRP7: Employ valid and reliable research strategies.

CRP8: Utilize critical thinking to make sense of problems and persevere in solving them.

CRP9: Model integrity, ethical leadership and effective management.

CRP10: Plan education and career paths aligned to personal goals.

CRP11: Use technology to enhance productivity.

CRP12: Work productively in teams while using cultural global competence.

- Logical and Comparison Operators
- If Statements (Practice program)
- For loops
- Generating Random Numbers
- While Loops
- Loop practice program
- Java program that computes federal income tax based on the user's information

Differentiation/Technology	Assessments
Differentiation strategies: • Additional assignments or extensions to existing assignments for students who are moving quickly • Partnering students who need support with students who can assist • Collaboration on all programming projects is encouraged. Collaboration is required as part of the course for all students Technology integration • Online platforms (CodeHS e.g.) • Desktop Java IDEs (IntelliJ, Eclipse, BlueJ e.g.)	• Graded programing projects • 2-3 question pop quizzes • End of unit CodeHS Quiz • AP Exam style end of unit quiz

Unit 3: Methods and Parameters

Big Ideas: *Course Objectives/ Content Statement(s)*

Students will understand...

- How to use existing functions or methods with and without parameters
- What a function or method is, and when to use one
- What a parameter is, and how to create one
- That some languages require functions or methods to work
- How to use functions or methods to make programs more readable and efficient

Essential Questions

What provocative questions will foster inquiry, understanding, and transfer of learning?

- What is a method?
- What is a parameter?
- How can methods help to write programs?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Methods are structures that can contain multiple code segments.
- A parameter is a way to give information to a method or function. Parameters are variables that are visible throughout a method, and must be assigned a value at the time the method or function is called.
- Methods are used as an organizational tools to assign one name to multiple lines of code, and can shorten code if certain segments of code need to be called multiple times in a program

Areas of Focus: Proficiencies (New Jersey Student Learning Standards)

Students will

- Write code to calls existing methods with and without parameters
- Write custom methods with and without parameters
- Write a program that contains all covered programing topics so far

NJ Computer Science and Design Thinking Standards

- **8.1.12.CS.3** Successful troubleshooting of complex problems involves multiple approaches

Topics/Lessons

Suggested Activities:

- CodeHS Unit 5 - Or teacher developed activities covering:
 - Methods/Functions
 - Methods/Functions with parameters
 - Scope of variables/Local variables
- CodeHS Unit 6 - Or teacher developed practice program containing:
 - Methods/Functions
 - Parameters
 - Variables
 - If Statements
 - Loops

including research, analysis, reflection, interaction with peers, and drawing on past experiences. ● 8.1.12.IC.1 The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities. ● 8.1.12.AP.4 Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks. Career-Ready Practices CRP1: Act as a responsible and contributing citizen and employee. CRP2: Apply appropriate academic and technical skills. CRP3: Attend to personal health and financial well-being. CRP4: Communicate clearly and effectively and with reason. CRP5: Consider the environmental, social and economic impacts of decisions. CRP6: Demonstrate creativity and innovation. CRP7: Employ valid and reliable research strategies. CRP8: Utilize critical thinking to make sense of problems and persevere in solving them. CRP9: Model integrity, ethical leadership and effective management. CRP10: Plan education and career paths aligned to personal goals. CRP11: Use technology to enhance productivity. CRP12: Work productively in teams while using cultural global competence.	
Differentiation/Technology	Assessments
Differentiation strategies: ● Additional assignments or extensions to existing assignments for students who are moving quickly ● Partnering students who need support with students who can assist ● Collaboration on all programming projects is encouraged. Collaboration is required as part of	● Graded programing projects ● 2-3 question pop quizzes ● End of unit CodeHS Quiz ● The practice program can be graded like the Create Performance Task

the course for all students

Technology integration

- Online platforms (CodeHS e.g.)
- Desktop Java IDEs (IntelliJ, Eclipse, BlueJ e.g.)

Unit 4: Basic Data Structures

Big Ideas: *Course Objectives/ Content Statement(s)*

Students will understand...

- What a data structure is
- That there are many types of data structures for different purposes

Essential Questions

What provocative questions will foster inquiry, understanding, and transfer of learning?

- What is a data structure?
- What is the difference between a list (Java or JavaScript) and an array (Java)?
- How can lists or arrays expand the functionality of programs?
- How can lists make programs more efficient to code?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- A data structure is a way to store multiple values in one place with one name.
- Lists and arrays both store data at a specific position - called an index. Lists in Java or JavaScript can change size, while arrays in Java cannot
- Data structures allow programs to write programs when they do not know how much data will be stored, so the program can work for any size data set
- Data structures can be used even if the programmer knows the size of the data set, but does not want to write the same code multiple times.

Areas of Focus: Proficiencies (New Jersey Student Learning Standards)

Students will

- Read and analyze code that deals with initializing, modifying, and searching through arrays
- Write programs that contain arrays
- Write algorithms that use arrays
- Write methods/functions that take arrays as parameters, and/or return arrays

NJ Computer Science and Design Thinking Standards

- **8.1.12.CS.3** Successful troubleshooting of complex problems involves multiple approaches

Topics/Lessons

Suggested Activities:

- CodeHS Unit 7 - Or teacher developed activities covering:
 - Creating an array or list
 - Indexing structure of an array or list
 - Adding and removing elements from lists
 - Iterating over an array or list
 - Searching for data in an array or list
- Java program that can keep track of numerical values and find averages and statistical information

including research, analysis, reflection, interaction with peers, and drawing on past experiences.

- **8.1.12.IC.1** The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities.
- **8.1.12.DA.1** Individuals select digital tools and design automated processes to collect, transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information.
- **8.1.12.DA.2** Choices individuals make about how and where data is organized and stored affects cost, speed, reliability, accessibility, privacy, and integrity.
- **8.1.12.DA.3** Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.
- **8.1.12.AP.1** Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.
- **8.1.12.AP.2** Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.
- **8.1.12.AP.3** Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.
- **8.1.12.AP.4** Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.
- **8.1.12.AP.5** Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths using a variety of resources, libraries, and tools.

Career-Ready Practices CRP1: Act as a responsible and contributing citizen and employee. CRP2: Apply appropriate academic and technical skills. CRP3: Attend to personal health and financial well-being. CRP4: Communicate clearly and effectively and with reason. CRP5: Consider the environmental, social and economic impacts of decisions. CRP6: Demonstrate creativity and innovation. CRP7: Employ valid and reliable research strategies. CRP8: Utilize critical thinking to make sense of problems and persevere in solving them. CRP9: Model integrity, ethical leadership and effective management. CRP10: Plan education and career paths aligned to personal goals. CRP11: Use technology to enhance productivity. CRP12: Work productively in teams while using cultural global competence.	
Differentiation/Technology	Assessments
Differentiation strategies: • Additional assignments or extensions to existing assignments for students who are moving quickly • Partnering students who need support with students who can assist • Collaboration on all programming projects is encouraged. Collaboration is required as part of the course for all students Technology integration • Online Learning Platforms (ex: CodeHS.) • Online IDEs (ex: repl.it, CodeHS) • Desktop Java IDEs (ex: IntelliJ, Eclipse, BlueJ)	• Graded programing projects • 2-3 question pop quizzes • End of unit CodeHS Quiz • AP Exam style end of unit quiz

Unit 5: Digital Information

Big Ideas: *Course Objectives/ Content Statement(s)*

Students will understand...

- What makes information and data digital
- How different types of information is stored in a computer
- How binary numbers can represent different types of information
- How different number systems are related to each other
- Differences between different types of data compression algorithms

Essential Questions

What provocative questions will foster inquiry, understanding, and transfer of learning?

- What is binary, and how is it useful for computers?
- How is hexadecimal related to binary, and decimal?
- How are different types of data stored in computers, and how can we tell which type of data is which?
- What are some ways storage space can be conserved using compression techniques?
- What is cryptography, and how can it help keep digital information safe?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Binary is a system of numbers, just like the decimal system that humans use to count, except binary uses only two digits: 0 and 1. Binary is the bridge between software and hardware in a computer because current is either flowing through a circuit (represented by 1) or it is not (represented by a 0)
- Hexadecimal is a number system that uses 16 digits (0-9 and A-F). A group of 4 binary digits represents 1 hexadecimal digit, meaning large numbers can be stored with fewer digits.
- All data is stored as binary digits at a hardware level. The only way to tell which type of data is which is by the context in which it is stored. It is impossible to look at a sequence of 1's and 0's with no context and tell what it represents.
- Data compression techniques can either be lossy or lossless. Lossless compressions may take up more space, but the data is guaranteed to be able to be restored to its original state. Lossy compressions take up less space, but some of the details of the data may be lost during compression.
- Cryptography is a way to encode messages so that the reader needs a "key" to read the message. There are many different forms of cryptography, some are very difficult to decode without the key, and some are easier. Cryptography is applied to

	data so that it can be safely stored or transmitted without being read by an unintended recipient.
Areas of Focus: Proficiencies (New Jersey Student Learning Standards)	Topics/Lessons
Students will ● Convert numbers between decimal, binary, and hexadecimal ● Use numerical representations of characters to convert between letters and numbers ● Identify lossy or lossless compression techniques ● Represent different types of data using binary or hexadecimal numbers NJ Computer Science and Design Thinking Standards ● 8.1.12.CS.3 Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences. ● 8.1.12.IC.1 The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities. ● 8.1.12.DA.1 Individuals select digital tools and design automated processes to collect, transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information. ● 8.1.12.DA.2 Choices individuals make about how and where data is organized and stored affects cost, speed, reliability, accessibility, privacy, and integrity. ● 8.1.12.DA.3 Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information. ● 8.1.12.AP.1 Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.	Suggested Activities: ● CodeHS Unit 8 - Or teacher developed activities covering: ○ Number Systems: Decimal, Binary, Hexadecimal ○ Encoding Text with Binary ○ Numerical representations of color to form images ○ Types of Data Compression (Lossy vs. Lossless) ○ Basic Cryptography: Caser Cypher, difference between public key and symmetric key encryption ● Java Program to convert between binary and decimal ○ Can be extended to Hexadecimal ● Java Program to encrypt and decrypt messages using a Caesar Cypher or a multiple digit symmetric key

- **8.1.12.AP.2** Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.
- **8.1.12.AP.3** Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.
- **8.1.12.AP.4** Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.
- **8.1.12.AP.5** Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths using a variety of resources, libraries, and tools.

Career-Ready Practices

CRP1: Act as a responsible and contributing citizen and employee.

CRP2: Apply appropriate academic and technical skills.

CRP3: Attend to personal health and financial well-being.

CRP4: Communicate clearly and effectively and with reason.

CRP5: Consider the environmental, social and economic impacts of decisions.

CRP6: Demonstrate creativity and innovation.

CRP7: Employ valid and reliable research strategies.

CRP8: Utilize critical thinking to make sense of problems and persevere in solving them.

CRP9: Model integrity, ethical leadership and effective management.

CRP10: Plan education and career paths aligned to personal goals.

CRP11: Use technology to enhance productivity.

CRP12: Work productively in teams while using cultural global competence.

Differentiation/Technology

Differentiation strategies:

Assessments

- Graded programing projects

- Additional assignments or extensions to existing assignments for students who are moving quickly
- Partnering students who need support with students who can assist
- Collaboration on all programming projects is encouraged. Collaboration is required as part of the course for all students

Technology integration

- Online platforms (CodeHS e.g.)
- Desktop Java IDEs (IntelliJ, Eclipse, BlueJ e.g.)

- 2-3 question pop quizzes
- End of unit CodeHS Quiz
- AP Exam style end of unit quiz

Unit 6: Practice Create Projects

Big Ideas: *Course Objectives/ Content Statement(s)*

Students will understand...

- Collaboration is a key aspect of the program development process
- Abstractions are a key part of many aspects of programming
- How to debug and test a program

Essential Questions

What provocative questions will foster inquiry, understanding, and transfer of learning?

- What is the creative development process?
- How can abstractions and algorithms help make developing a program more efficient?
- How can I effectively find issues in my program and fix them?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- The creative development process involves a goal, breaking down the goal into manageable pieces, creating abstractions when necessary, then testing and fixing any problems that occur. Collaboration is a big part of all of these steps.
- Abstractions are ways of removing detail from a part of a program to only focus on the important parts. For example creating a function that contains many lines of code means once that function is created, the programmer only needs to know the name of the function, its parameters, and return type to be able to use it. Many abstractions contain algorithms, which are segments of code that accomplish a specific task.
- Debugging involves ensuring that every line of the program is tested. Choose test cases that will use different parts of the program, add print statements to see the values of variables if an issue is discovered to help find when a value is not what the programmer intended. Use a classmate to test a program and be a second set of eyes to help find issues that you may have missed

Areas of Focus: Proficiencies (New Jersey Student Learning Standards)

Students will

- Create two practice programs that are aligned with the portfolio submission required by CollegeBoard.

Topics/Lessons

Suggested Activities:

- CodeHS Units 9 and 10 - Or teacher developed programming project to prepare for the portfolio submission to CollegeBoard

- Collaborate with classmates while coding and testing.
- Use the rubric provided by CollegeBoard to ensure all requirements are met.

NJ Computer Science and Design Thinking Standards

- **8.1.12.CS.3** Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences.
- **8.1.12.IC.1** The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities.
- **8.1.12.DA.1** Individuals select digital tools and design automated processes to collect, transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information.
- **8.1.12.DA.2** Choices individuals make about how and where data is organized and stored affects cost, speed, reliability, accessibility, privacy, and integrity.
- **8.1.12.DA.3** Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.
- **8.1.12.AP.1** Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.
- **8.1.12.AP.2** Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.
- **8.1.12.AP.3** Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.
- **8.1.12.AP.4** Complex programs are designed as systems of interacting modules, each with a

- Review the rubric used for the AP Exam, give students time to read and ask questions. Show sample projects from past

specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks. ● 8.1.12.AP.5 Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths using a variety of resources, libraries, and tools. Career-Ready Practices CRP1: Act as a responsible and contributing citizen and employee. CRP2: Apply appropriate academic and technical skills. CRP3: Attend to personal health and financial well-being. CRP4: Communicate clearly and effectively and with reason. CRP5: Consider the environmental, social and economic impacts of decisions. CRP6: Demonstrate creativity and innovation. CRP7: Employ valid and reliable research strategies. CRP8: Utilize critical thinking to make sense of problems and persevere in solving them. CRP9: Model integrity, ethical leadership and effective management. CRP10: Plan education and career paths aligned to personal goals. CRP11: Use technology to enhance productivity. CRP12: Work productively in teams while using cultural global competence.	
Differentiation/Technology	Assessments
Differentiation strategies: ● Additional assignments or extensions to existing assignments for students who are moving quickly ● Partnering students who need support with students who can assist ● Collaboration on all programming projects is encouraged. Collaboration is required as part of the course for all students Technology integration ● Online platforms (CodeHS e.g.)	● Graded programming projects using the Create rubric provided by CollegeBoard

- Desktop Java IDEs (IntelliJ, Eclipse, BlueJ e.g.)

Unit 7: The Internet

Big Ideas: *Course Objectives/ Content Statement(s)*

Students will understand...

- What the internet is, and how it works on a basic level
- What hardware and software is used to make the internet work
- How data can be transmitted securely, and the risks of transmitting data on the internet
- How the internet has had an impact on our society

Essential Questions

What provocative questions will foster inquiry, understanding, and transfer of learning?

- What is the internet?
- What hardware is used by the internet?
- How do devices communicate with each other?
- What are routing, packets, protocols, and the DNS?
- How can the internet help computers work together?
- How has the internet impacted our society?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- The internet is a network of computers and other hardware devices that spreads across the entire planet
- The internet was set up so many different types of devices can communicate. Internet enabled devices just need to use the same protocols - or commands - to communicate with each other. Every device on the internet has a unique Internet Protocol (or IP) Address
- The Domain Name System (DNS) is a collection of servers that translate an URL (Uniform Resource Locator) or web address like google.com into an IP address.
- Routing is a way to assign a path to data transmitted on the internet so it can get to its destination. Large files can be broken up into packets and reassembled at the destination to make the transmission more efficient.
- Computers can work together to solve problems using a process called Distributed computing. When a computer works by itself it is either called Sequential (one processor) or Parallel (multiple processors) computing.

	The internet has affected most aspects of how we live today, including the need for new laws and regulations.
Areas of Focus: Proficiencies (New Jersey Student Learning Standards)	Topics/Lessons
Students will - Research different ways the internet has changed our world - Identify different aspects of the internet, and explain what they do - Identify security risks NJ Computer Science and Design Thinking Standards 8.1.12.CS.3 Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences. 8.1.12.NI.1 The scalability and reliability of the Internet are enabled by the hierarchy and redundancy in networks. 8.1.12.NI.2 Network topology is determined by many characteristics. 8.1.12.NI.3 Network security depends on a combination of hardware, software, and practices that protect data while it is at rest, in transit, and in use. 8.1.12.NI.4 The needs of users and the sensitivity of data determine the level of security implemented. Advanced attacks take advantage of common security vulnerabilities. Career-Ready Practices CRP1: Act as a responsible and contributing citizen and employee. CRP2: Apply appropriate academic and technical skills. CRP3: Attend to personal health and financial well-being. CRP4: Communicate clearly and effectively and with reason. CRP5: Consider the environmental, social and economic	Suggested Activities: - CodeHS Unit 11 - Or teacher developed lessons covering: - Basics of the internet - Internet Hardware - Websites, DNS, Routing - Packets and Protocols - Sequential, Parallel, and Distributed Computing - Cybersecurity - Impacts of the Internet - Copyright, Licensing, and Crediting Creators - CodeHS Unit 12 - Or teacher developed activities covering the effects of the internet

impacts of decisions.

CRP6: Demonstrate creativity and innovation.

CRP7: Employ valid and reliable research strategies.

CRP8: Utilize critical thinking to make sense of problems and persevere in solving them.

CRP9: Model integrity, ethical leadership and effective management.

CRP10: Plan education and career paths aligned to personal goals.

CRP11: Use technology to enhance productivity.

CRP12: Work productively in teams while using cultural global competence.

Differentiation/Technology	Assessments
Differentiation strategies: - Additional assignments or extensions to existing assignments for students who are moving quickly - Partnering students who need support with students who can assist - Collaboration on all programming projects is encouraged. Collaboration is required as part of the course for all students Technology integration - Online platforms (CodeHS e.g.) - Desktop Java IDEs (IntelliJ, Eclipse, BlueJ e.g.)	- Graded programing projects 2-3 question pop quizzes - End of unit CodeHS Quiz - AP Exam style end of unit quiz - Graded project on the effects of the internet

Unit 8: Data (CHS 13-14)

Big Ideas: *Course Objectives/ Content Statement(s)*

Students will understand...

- How computers can be used to process data
- How computers collect and store data
- How computers can help answer questions by analyzing data

Essential Questions

What provocative questions will foster inquiry, understanding, and transfer of learning?

- How can computers help process and analyze data?
- What ways can computers store data?
- How have computers helped answer questions that previously would have taken an unreasonable amount of time?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Computers can process data much faster than humans. In many cases it is much more efficient for a person to write a program to analyze data than to analyze the data themselves.
- Data sets are stored in data structures.
- Abstractions and algorithms have helped programmers to build on existing systems to create programs that can efficiently collect and analyze large sets of real world data. Computers can find patterns that humans may not have been able to.

Areas of Focus: Proficiencies (New Jersey Student Learning Standards)

Students will

- Explore the importance of data in technology
- Explain how representing data visually can help show trends or patterns

NJ Computer Science and Design Thinking Standards

- **8.1.12.CS.3** Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences.
- **8.1.12.NI.1** The scalability and reliability of the Internet are enabled by the hierarchy and redundancy in networks.

Topics/Lessons

Suggested Activities:

- CodeHS Unit 13 - Or teacher developed activities covering:
 - The importance of data in computers
 - Visualizing and interpreting data
 - Data collection and limitations
- CodeHS Unit 14 - Or teacher developed project where students answer questions about a publicly available large data set.
- Java program that can find a pattern in a large data set (ex: longest run of numbers within a tolerance)

- **8.1.12.NI.2** Network topology is determined by many characteristics.
- **8.1.12.NI.3** Network security depends on a combination of hardware, software, and practices that protect data while it is at rest, in transit, and in use.
- **8.1.12.NI.4** The needs of users and the sensitivity of data determine the level of security implemented. Advanced attacks take advantage of common security vulnerabilities.
- **8.1.12.DA.1** Individuals select digital tools and design automated processes to collect, transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information.
- **8.1.12.DA.2** Choices individuals make about how and where data is organized and stored affects cost, speed, reliability, accessibility, privacy, and integrity.
- **8.1.12.DA.3** Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.
- **8.1.12.AP.1** Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.
- **8.1.12.AP.2** Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.
- **8.1.12.AP.3** Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.
- **8.1.12.AP.4** Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.

● 8.1.12.AP.5 Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths using a variety of resources, libraries, and tools. Career-Ready Practices CRP1: Act as a responsible and contributing citizen and employee. CRP2: Apply appropriate academic and technical skills. CRP3: Attend to personal health and financial well-being. CRP4: Communicate clearly and effectively and with reason. CRP5: Consider the environmental, social and economic impacts of decisions. CRP6: Demonstrate creativity and innovation. CRP7: Employ valid and reliable research strategies. CRP8: Utilize critical thinking to make sense of problems and persevere in solving them. CRP9: Model integrity, ethical leadership and effective management. CRP10: Plan education and career paths aligned to personal goals. CRP11: Use technology to enhance productivity. CRP12: Work productively in teams while using cultural global competence.	
Differentiation/Technology	Assessments
Differentiation strategies: ● Additional assignments or extensions to existing assignments for students who are moving quickly ● Partnering students who need support with students who can assist ● Collaboration on all programming projects is encouraged. Collaboration is required as part of the course for all students Technology integration ● Online platforms (CodeHS e.g.) ● Desktop Java IDEs (IntelliJ, Eclipse, BlueJ e.g.)	● Graded programing projects ● 2-3 question pop quizzes ● End of unit CodeHS Quiz ● AP Exam style end of unit quiz ● Graded data project

Unit 9: Impact of Computing

Big Ideas: *Course Objectives/ Content Statement(s)*

Students will understand...

- What is considered a computing innovation
- How computing innovations collect, use, and output data
- Security risks of computing innovations

Essential Questions

What provocative questions will foster inquiry, understanding, and transfer of learning?

- What is a computing innovation?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Computing innovations have a program as their main feature. They collect data, and output data. They can be either hardware, or software that runs on existing hardware.

Areas of Focus: Proficiencies (New Jersey Student Learning Standards)

Students will

- Choose and research a computing innovation
- Describe how the innovation inputs and outputs data

NJ Computer Science and Design Thinking Standards

- **8.1.12.CS.3** Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences.
- **8.1.12.IC.1** The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities.
- **8.1.12.DA.1** Individuals select digital tools and design automated processes to collect, transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information.

Topics/Lessons

Suggested Activities:

- Partner activity where students look up various innovations and decided if they are computing innovations or not, and explain why
- CodeHS Unit 15 - Or teacher a teacher developed project covering the impact of computing, including:
 - Creating an artifact
 - Identifying "Computing Innovations" and selecting one to research
 - Describing how data is used as an input and output in the innovation
 - Identifying privacy and security concerns

- **8.1.12.DA.2** Choices individuals make about how and where data is organized and stored affects cost, speed, reliability, accessibility, privacy, and integrity.
- **8.1.12.DA.3** Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.
- **8.1.12.AP.1** Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.
- **8.1.12.AP.2** Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.
- **8.1.12.AP.3** Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.
- **8.1.12.AP.4** Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.
- **8.1.12.AP.5** Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths using a variety of resources, libraries, and tools.

Career-Ready Practices

CRP1: Act as a responsible and contributing citizen and employee.

CRP2: Apply appropriate academic and technical skills.

CRP3: Attend to personal health and financial well-being.

CRP4: Communicate clearly and effectively and with reason.

CRP5: Consider the environmental, social and economic impacts of decisions.

CRP6: Demonstrate creativity and innovation.

CRP7: Employ valid and reliable research strategies.

CRP8: Utilize critical thinking to make sense of problems

and persevere in solving them. CRP9: Model integrity, ethical leadership and effective management. CRP10: Plan education and career paths aligned to personal goals. CRP11: Use technology to enhance productivity. CRP12: Work productively in teams while using cultural global competence.	
Differentiation/Technology	Assessments
Differentiation strategies: - Additional assignments or extensions to existing assignments for students who are moving quickly - Partnering students who need support with students who can assist - Collaboration on all programming projects is encouraged. Collaboration is required as part of the course for all students Technology integration - Online platforms (CodeHS e.g.) - Desktop Java IDEs (IntelliJ, Eclipse, BlueJ e.g.)	- Checkpoints after each part of the project - Graded computing innovation project

Unit 10: Create Performance Task	
Big Ideas: <i>Course Objectives/ Content Statement(s)</i> Students will understand... - How to combine previous knowledge into a new program - How to work collaboratively to debug and test programs	
Essential Questions <i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	Enduring Understandings <i>What will students understand about the big ideas?</i>
- How can collaboration help to develop programs? - How can ideas be combined into one new product?	Students will understand that: Collaboration can be helpful to find errors that would have been missed by the programmer

	● It is helpful to work with another programmer to think through ideas and develop more efficient algorithms ● Abstractions can help us combine many different programming statements/segments to form a larger program
Areas of Focus: Proficiencies (New Jersey Student Learning Standards)	Topics/Lessons
Students will ● Create a program without the help of the teacher, but with the help of classmates ● Answer questions about the program and the development process NJ Computer Science and Design Thinking Standards ● 8.1.12.CS.3 Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences. ● 8.1.12.IC.1 The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities. ● 8.1.12.DA.1 Individuals select digital tools and design automated processes to collect, transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information. ● 8.1.12.DA.2 Choices individuals make about how and where data is organized and stored affects cost, speed, reliability, accessibility, privacy, and integrity. ● 8.1.12.DA.3 Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.	Suggested Activities: ● Review the Create Performance Task rubric provided by CollegeBoard ● Create Performance Task as described in the CollegeBoard Course Description ○ Create a program collaboratively ○ Independently answer written questions about the development process and the final product ○ Independently create a video demonstrating the functionality of the program ○ Submit the program, written questions, and video to the CollegeBoard Digital Portfolio

- **8.1.12.AP.1** Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.
- **8.1.12.AP.2** Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.
- **8.1.12.AP.3** Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.
- **8.1.12.AP.4** Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.
- **8.1.12.AP.5** Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths using a variety of resources, libraries, and tools.

Career-Ready Practices

CRP1: Act as a responsible and contributing citizen and employee.

CRP2: Apply appropriate academic and technical skills.

CRP3: Attend to personal health and financial well-being.

CRP4: Communicate clearly and effectively and with reason.

CRP5: Consider the environmental, social and economic impacts of decisions.

CRP6: Demonstrate creativity and innovation.

CRP7: Employ valid and reliable research strategies.

CRP8: Utilize critical thinking to make sense of problems and persevere in solving them.

CRP9: Model integrity, ethical leadership and effective management.

CRP10: Plan education and career paths aligned to personal goals.

CRP11: Use technology to enhance productivity.

CRP12: Work productively in teams while using cultural global competence.

Differentiation/Technology	Assessments
Differentiation strategies: - Additional assignments or extensions to existing assignments for students who are moving quickly - Partnering students who need support with students who can assist - Collaboration on all programming projects is encouraged. Collaboration is required as part of the course for all students Technology integration - Online platforms (CodeHS e.g.) - Desktop Java IDEs (IntelliJ, Eclipse, BlueJ e.g.)	- Create Project after completion - CollegeBoard requires that no teacher feedback is given until the student submits their program to the digital portfolio website

Unit 11: AP Exam Review	
Big Ideas: <i>Course Objectives/ Content Statement(s)</i> Students will be prepared to take the AP Exam	
Essential Questions <i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	Enduring Understandings <i>What will students understand about the big ideas?</i>
- What were the important concepts covered this year? - What do I need to work on before the AP Exam?	Students will understand... - The concepts covered on the AP Exam: - Basic programming concepts - Algorithms - Abstractions - Data - The Internet - Computing Innovations - The exam format: - All multiple choice questions with 4 answer choices. Some questions will ask students to select two answers
Areas of Focus: Proficiencies (New Jersey Student Learning Standards)	Topics/Lessons
Students will review material learned during the year to prepare for the AP Exam	Suggested Activities:

NJ Computer Science and Design Thinking Standards

- **8.1.12.CS.3** Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences.
- **8.1.12.IC.1** The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities.
- **8.1.12.DA.1** Individuals select digital tools and design automated processes to collect, transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information.
- **8.1.12.DA.2** Choices individuals make about how and where data is organized and stored affects cost, speed, reliability, accessibility, privacy, and integrity.
- **8.1.12.DA.3** Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.
- **8.1.12.AP.1** Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.
- **8.1.12.AP.2** Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.
- **8.1.12.AP.3** Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.
- **8.1.12.AP.4** Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.

- Review old AP Style quizzes on each topic
- Practice AP Exam from CodeHS (Unit 17)
- Practice AP Exam from CollegeBoard

- **8.1.12.AP.5** Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths using a variety of resources, libraries, and tools.
- **8.1.12.NI.1** The scalability and reliability of the Internet are enabled by the hierarchy and redundancy in networks.
- **8.1.12.NI.2** Network topology is determined by many characteristics.
- **8.1.12.NI.3** Network security depends on a combination of hardware, software, and practices that protect data while it is at rest, in transit, and in use.
- **8.1.12.NI.4** The needs of users and the sensitivity of data determine the level of security implemented. Advanced attacks take advantage of common security vulnerabilities.

Career-Ready Practices

CRP1: Act as a responsible and contributing citizen and employee.

CRP2: Apply appropriate academic and technical skills.

CRP3: Attend to personal health and financial well-being.

CRP4: Communicate clearly and effectively and with reason.

CRP5: Consider the environmental, social and economic impacts of decisions.

CRP6: Demonstrate creativity and innovation.

CRP7: Employ valid and reliable research strategies.

CRP8: Utilize critical thinking to make sense of problems and persevere in solving them.

CRP9: Model integrity, ethical leadership and effective management.

CRP10: Plan education and career paths aligned to personal goals.

CRP11: Use technology to enhance productivity.

CRP12: Work productively in teams while using cultural global competence.

Differentiation/Technology

Assessments

Differentiation strategies:

- Additional assignments or extensions to existing assignments for students who are moving quickly
- Partnering students who need support with students who can assist
- Collaboration on all programming projects is encouraged. Collaboration is required as part of the course for all students

Technology integration

- Online platforms (CodeHS e.g.)
- Desktop Java IDEs (IntelliJ, Eclipse, BlueJ e.g.)

- Practice AP Exam

Unit 12: Introduction to Graphics in Java

Big Ideas: *Course Objectives/ Content Statement(s)*

Students will understand...

- How to create multiple windows in Java
- How to use the Graphics class to create shapes and colors
- How to combine ideas to make a basic Graphical User Interface (GUI)

Essential Questions

What provocative questions will foster inquiry, understanding, and transfer of learning?

- How can a Java program be created without using the console?
- What can be placed in a window?
- How does the coordinate system in Java work?

Enduring Understandings

What will students understand about the big ideas?

Students will understand...

- The JFrame class is what Java uses for windows.
- In Java (and many other languages) the top left corner of a window is represented by the coordinate (0,0) moving to the left increases the x (first) value, and moving down increases the y (second) value of the coordinate. This way all of the values that are visible have positive coordinate values.
- There are many types of components that can go in windows. Ex: JButton, JLabel, JTextField, or JPanel
- The Graphics class can be used to draw shapes and display different colors
- Multiple shapes can be combined to make new images
- We can use variables for coordinates to make shapes and images move on the screen

Areas of Focus: Proficiencies (New Jersey Student Learning Standards)

Students will

- Use the JFrame and JPanel class to create a window with Graphics
- Use the JLabel and JButton class to create components in a window

NJ Computer Science and Design Thinking Standards

Topics/Lessons

Suggested Activities:

- Class discussion of how to set up a GUI in Java using at least one of each of the components listed above.
- Students draw an image on paper, and attempt to recreate it in Java
- Group project where students create a game or other application that involves shapes or images to move on the screen. Students should submit a

- **8.1.12.CS.3** Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences.
- **8.1.12.IC.1** The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities.
- **8.1.12.DA.1** Individuals select digital tools and design automated processes to collect, transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information.
- **8.1.12.DA.2** Choices individuals make about how and where data is organized and stored affects cost, speed, reliability, accessibility, privacy, and integrity.
- **8.1.12.DA.3** Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.
- **8.1.12.AP.1** Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.
- **8.1.12.AP.2** Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.
- **8.1.12.AP.3** Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.
- **8.1.12.AP.4** Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.
- **8.1.12.AP.5** Complex programs are developed, tested, and analyzed by teams drawing on the

proposal to the teacher for approval before starting work on the project.

members' diverse strengths using a variety of resources, libraries, and tools. Career-Ready Practices CRP1: Act as a responsible and contributing citizen and employee. CRP2: Apply appropriate academic and technical skills. CRP3: Attend to personal health and financial well-being. CRP4: Communicate clearly and effectively and with reason. CRP5: Consider the environmental, social and economic impacts of decisions. CRP6: Demonstrate creativity and innovation. CRP7: Employ valid and reliable research strategies. CRP8: Utilize critical thinking to make sense of problems and persevere in solving them. CRP9: Model integrity, ethical leadership and effective management. CRP10: Plan education and career paths aligned to personal goals. CRP11: Use technology to enhance productivity. CRP12: Work productively in teams while using cultural global competence.	
Differentiation/Technology	Assessments
Differentiation strategies: • Additional assignments or extensions to existing assignments for students who are moving quickly • Partnering students who need support with students who can assist • Collaboration on all programming projects is encouraged. Collaboration is required as part of the course for all students Technology integration • Online platforms (CodeHS e.g.) • Desktop Java IDEs (IntelliJ, Eclipse, BlueJ e.g.)	• Picture Project • Group Project