

Summit Public Schools
Summit, New Jersey
Grade Level: 11-12 / Content Area: Mathematics

Course Overview:

AP Computer Science A is a college-level computer science course that addresses a variety of core Java programming concepts. Throughout the course, students will review and extend previously learned programming skills as well as learn new advanced concepts. Students will gain a variety of Java skills that serve not only to develop strong programming ability but also to assist them in their conceptual understanding of programming as a whole.

This class will have a minimum of 20 hours of programming time during class before the AP Exam. This course covers all required material to prepare students to take the AP Computer Science A Exam.

Prerequisite Course:

AP Computer Science Principles

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	
Primitive Types	6 Days
Using Objects	7 Days
Boolean Expressions and if Statements	8 Days
Iteration	9 Days
Marking Period 2	
Writing Classes	10 Days
Arrays	10 Days
ArrayLists	10 Days
Marking Period 3	
2D Arrays	10 Days
Inheritance	10 Days
Recursion	10 Days
Marking Period 4	
AP Exam Review	10 Dyas
Final Project	20 Days

Unit 1: Primitive Types

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

Students will understand...

- How variables are a necessary part of creating any program
- Different variable types have different possible values and ranges
- How to use operators to get the desired result from a program

Essential Questions

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

- How can programs be used to solve problems?
- How are numerical values used in programs?
- How are mathematical concepts used in programs?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Abstractions are used to help keep programmers from rewriting the same code more than once
- Variables are used to make the same algorithms output different values based on user input
- Variables and operators have a specific order of operations that determines the result of a code segment

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

Students will:

- Use different variable types in programs
- Create programs that use mathematical operators
- Create programs that use assignment operators
- Apply the most effective data type to a given situation

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.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

Suggested Topics/Lessons

- Intro/Review of Java
 - Types of errors
- Variables and Data Types
 - int
 - double
 - boolean
- Expressions and Assignment Statements
 - +, -, *, /, % operators
 - Predict result given code
 - complete code segments to produce desired output
- Compound Assignment Operators
 - Predict result given code
 - complete code segments to produce desired output
- Casting and Ranges of Variables
 - Predict result given code
 - complete code segments to produce desired output

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. 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.	○ Explain why code will not compile ● Programming Project(s) ○ Practice with operations along with user input
Differentiation/Technology	Assessments
Differentiation Strategies ● Partner students based on ability level ● Provide additional assignments to students who are ahead of the course expectations Technology Integration ● Students will program using a Java IDE ○ IntelliJ ○ Repl.it ● Online questions and activities on CollegeBoard Global Perspectives ● Throughout the course students will discuss how	● Programs and practice code ● AP Exam Style Primitive type quiz

programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact

Unit 2: Using Objects

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

Students will understand...

- What the state and behavior of an object is
- Existing classes in Java can be used to simplify programs
- The relationship between a class and an object

Essential Questions

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

- What is the relationship between a class and an Object?
- How can existing methods be used?
- What is a return type?
- What is the difference between static and non-static methods?
- What is a wrapper class?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- An object is an instance of a class. The class is a representation of the attributes and behaviors of an object
- To use a method the programmer must know its "signature" or the name, return type, and parameters
- Parameters are used to pass information into a method
- Overloaded methods are methods that have the same name, but different parameters
- Variables must be the correct type in Java to store data. The return type of a method should match (or be a subtype of) the variable in which it is being stored
- Void methods do not return any data, every other method must return the data type that matches its return type
- Static methods and variables belong to the class, not the object
- Existing class information can be found in the Java API
- Wrapper classes are classes that represent primitive types.

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

Students will:

- Call methods that already exists in Java
- Use return values of methods

Suggested Topics/Lessons

- Instances of Classes
 - Describe behavior of a code segment
- Creating and Storing Objects

● Use the Math and String class to create programs 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.DA.2 Choices individuals make about how and where data is organized and stored affects cost, speed, reliability, accessibility, privacy, and integrity. 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.	○ Write code to create a class ○ Determine appropriate code to use methods from a class ○ null, and NullPointerExceptions ● Calling a void method ○ Determine appropriate code to call a method ○ write code that calls void methods ○ With and without parameters ○ Static vs non-static ● Calling a non-void method ○ Determine appropriate code to call a method ○ Write code that appropriately uses a methods return value ○ Static vs non-static ● Strings ○ Concatenation ○ Literals ○ String Methods ○ Escape sequences: /n, /t, // ● Existing Classes ○ Math ■ Ranges for random numbers ○ Integer, Double, Boolean ■ Autoboxing and unboxing ● Programming Project(s) ○ Predicting output of string methods ○ Predicting output of Math class methods
Differentiation/Technology	Assessments
Differentiation Strategies ● Partner students based on ability level ● Provide additional assignments to students who are ahead of the course expectations Technology Integration ● Students will program using a Java IDE	● Programs and practice code ● AP Style multiple choice questions ● AP Exam Style Test on Units 1 and 2 (Primitive and Reference Data Types)

- IntelliJ
 - Repl.it
- Online questions and activities on CollegeBoard

Global Perspectives

- Throughout the course students will discuss how programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact

Unit 3: Boolean Expressions and If Statements

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

Students will understand...

- Booleans and if statements are how programs can behave differently each time they are executed
- Why selection statements are a necessary part of programming
- How to apply selection statements to a new program
- How the way a reference type is saved in the memory affects the output of boolean expressions

Essential Questions

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

- How does the computer read relational operators?
- How can booleans be used to control the flow of a program?
- How can we verify the output of a complex boolean expression?
- How can we compare the values of two objects?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Variables can be compared using == and !=
- Numerical values can be compared using <, >, <=, and >=
- if statements can be used to execute or not execute code segments depending on a boolean value or expression
- if else statements can be used to run a specific code segment depending on the value of a boolean expression
- if statements can be placed inside other if statements creating a nested if statement
- !, &&, and || can be used to combine or negate boolean values and expressions
- Truth tables can be used to verify or come up with boolean identities
- Objects can be compared using the equals() method, variables can be compared using the == operator

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

Students will:

- Predict and verify the output of boolean expressions
- Trace the flow of a program containing if and if else statements depending on the values of variable and/or user input
- Write programs containing if and if else

Suggested Topics/Lessons

- Boolean Expressions
 - Use operators with booleans
 - compound boolean expressions
 - Equivalent expressions
 - !, && and || operators
 - DeMorgans Law
 - Short circuit evaluation

statements

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.AP.1** Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.
- **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.

- if statements and control flow
 - Determine the result given a code segment
 - Write code to satisfy specifications
 - Nested if statements
- is-else statements
 - Determine the result given a code segment
 - Write code to satisfy specifications
 - use test cases to find errors
- Comparing objects
 - == vs .equals
 - Predict output given a code segment
- Programming Project(s)
 - Program to compute federal income tax based on income and other factors
 - Choose your own adventure game

Differentiation/Technology

Assessments

Differentiation Strategies

- Partner students based on ability level
- Provide additional assignments to students who

- Programs and practice code
- Teacher provided feedback during programming
- AP Exam Style Quiz

are ahead of the course expectations

Technology Integration

- Students will program using a Java IDE
 - IntelliJ
 - Repl.it
- Online questions and activities on CollegeBoard

Global Perspectives

- Throughout the course students will discuss how programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact

Unit 4: Iteration

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

Students will understand...

- Loops are used to control how many times a program segment is executed
- Iteration provides instructions to the computer to process many different input values

Essential Questions

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

- How can a while loop be used to repeat code statements?
- How can a for loop be used to repeat code statements?
- Which type of loop makes the most sense to complete a specific task?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Loops are used to control how many times code is executed
- Both types of loops contain a boolean expression, the loop is executed as long as the boolean expression evaluates to true
- A return statement inside a loop will end the iteration
- While loops only contain a boolean expression, for loops contain an initialization statement, a boolean expression, and an increment or decrement statement
- Off by one errors are when a loop is executed one more or one fewer time than expected
- Every for loop could be written as a while loop, and ever while loop could be written as a for loop

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

Students will:

- Trace loops to find output given values of variables or user input
- Write program segments containing while and for loops
- Write program segments containing nested loops

NJ Computer Science and Design Thinking Standards

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

Suggested Topics/Lessons

- While loops
 - Complete loops to produce desired output
 - Determine the result of a given code segment
 - Write loops to satisfy specific requirements
 - Determine the number of times a code segment will be executed
- For loops
 - Complete loops to produce desired output

including research, analysis, reflection, interaction with peers, and drawing on past experiences. ● 8.1.12.AP.1 Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation. ● 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.	○ Determine the result of a given code segment, and if two code segments produce the same output ○ Write loops to satisfy specific requirements ○ Determine the number of times a code segment will be executed ○ Standard algorithms to find if a number is divisible by another, identify the digits in a multiple digit integer, determine the frequency with which a certain condition is met, determine a minimum or maximum value, compute a sum, average, or mode ● Nested Iteration ○ Complete loops to produce desired output ○ Determine the result of a given code segment ○ Write loops to satisfy specific requirements ○ Explain how the output changes given a change to the code ○ Determine the number of times a code segment will be executed ● Developing algorithms using Strings ○ Determine the result of a given algorithm ○ Create algorithms using loops and methods from the String class ○ Standard algorithms to find if substrings have a specific property, create a reversed version of a String ● Programming Project(s) ○ Write methods that manipulate Strings
Differentiation/Technology	Assessments
Differentiation Strategies ● Partner students based on ability level ● Provide additional assignments to students who are ahead of the course expectations Technology Integration ● Students will program using a Java IDE ○ IntelliJ ○ Repl.it	● Programs and practice code ● Free Response Questions from previous AP Exams ● Teacher provided feedback during programming projects ● Practice multiple choice questions from CollegeBoard ● AP Exam Style Test on Units 3 and 4 (If Statements and Loops)

- Online questions and activities on CollegeBoard

Global Perspectives

- Throughout the course students will discuss how programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact

Unit 5: Writing Classes

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

Students will understand...

- How a class is related to an Object
- What components can be a part of a class
- When static methods and variables are appropriate, and how they function compared to non-static variables and methods

Essential Questions

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

- What is the difference between a class and an Object?
- What are static variables and methods?
- Why are accessor and mutator methods necessary?
- What is the difference between private and public variables and methods?
- When are we able to call certain methods, and use certain variables?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- A class is the formal representation of an Object, like a blueprint compared to a building
- Multiple Objects can be created from the same class
- Static variables and methods belong to the class, not the Object. No matter how many instances (Objects) there are of the same type, each static variable will only be created once.
- Public and private are visibility modifiers for variables and methods that determine where a method or variable can be accessed. Private is only within the class, while public is anywhere that has access to the class or Object
- A variable or method's scope is determined by where it is created. It can only be used where it was made, if it was created in an if statement then it can only be used in that if statement for example.

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

Students will:

- Create custom classes to meet specific goals
- Create a program that use custom classes to create objects
- Create a program where multiple Objects can interact with each other

Suggested Topics/Lessons

- Anatomy of a class
 - Determine an appropriate class design
 - Compete code segments within a class
 - private vs. public
- Constructors
 - Default and Parameters
 - Overloaded

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.DA.2** Choices individuals make about how and where data is organized and stored affects cost, speed, reliability, accessibility, privacy, and integrity.

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.

- Complete constructors given variables, and the intended functionality
- Documentation with Comments
 - Parameter descriptions, Return values, Method's purpose
 - Write comments given a method
- Accessor Methods
 - Write methods given a class's variables
 - Explain why code will not work as intended
- Mutator methods
 - Write methods given a class's variables
 - Explain why code will not work as intended
- Static Variables and Methods
 - When static is appropriate
- Scope and Access
 - Local vs global variables
- The keyword "this"
- Programming Project(s)
 - Create a class that represents a Rectangle
 - Create a program that keeps track of multiple students using a Course class and a Student class. Each Student has 3 Course instance variables

Differentiation/Technology

Assessments

Differentiation Strategies

- Partner students based on ability level
- Provide additional assignments to students who are ahead of the course expectations

Technology Integration

- Students will program using a Java IDE
 - IntelliJ
 - Repl.it

- Programs and practice code
- Free Response Questions from previous AP Exams
- Teacher provided feedback during programming projects
- Practice multiple choice questions from CollegeBoard
- AP Exam Style Test on Unit 5 (Writing Classes)

- Online questions and activities on CollegeBoard

Global Perspectives

- Throughout the course students will discuss how programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact

Unit 6: Arrays

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

Students will understand...

- How to store multiple values in one variable
- How to develop algorithms that use arrays

Essential Questions

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

- How can multiple values of the same type be stored in one place?
- How can elements of arrays be accessed?
- What can we do with arrays that was not possible with individual variables?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Arrays allow multiple values of the same type (reference or primitive) to be stored with one variable
- The size of an array is set upon creation, and cannot be changed
- Every type of data in Java is 0 by default. For numerical values this is interpreted as the number 0 or 0.0, for booleans this means false, and for reference types this means null. In an array these are the default values for each type, there will not be an error if values are not initialized.
- Elements can be accessed by their index using the square brackets []. Indexing starts at 0 and ends at length-1.
- Iteration can be used to access all elements of an array
- for and while loops require an index, while an enhanced for loop does not
- An `ArrayIndexOutOfBoundsException` will be thrown if attempting to access an invalid index
- Enhanced for loops cannot modify the values saved in the array using an `=`, but they can call methods on reference types saved in an array
- Standard algorithm (listed below) can be modified to accomplish more specific tasks

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

Students will:

- Write code to create arrays of a specific type

Suggested Topics/Lessons

- Array creation and access

- Access individual elements of arrays
- Write standard algorithms that utilize array traversals
- Create custom algorithms using arrays

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

- Write code that creates and accesses elements in arrays
- Complete code to accomplish a specific task using an array
- Initializer lists
- Traversing arrays
 - Determine the output given a code segment
 - Identify errors in given code
 - Write code to traverse an array, and manipulate elements
- Enhanced for loops
 - Write code to traverse an array
 - Limitations and advantages of enhanced for (for-each) vs standard for loop
- Developing algorithms using arrays
 - Standard algorithms: Determine a maximum or minimum; compute sum, mean, mode; determine if at least one element has a specific property; determines if all elements have a specific property; access all consecutive pairs of elements; determine the presence or absence of duplicates; determine the number of elements with a specific property; shift or rotate elements; reverse the order of elements
 - Modify standard algorithm for more specific purposes
- Programming Project(s)
 - Write code that can traverse and modify (if possible) arrays using for, and for-each loops to see limitations and advantages of each
 - Create custom methods, one for each standard algorithm
 - Modify the grading program from Unit 5 to work with arrays, so Students can have an array of Courses instead of 3 separate variables

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 ● Partner students based on ability level ● Provide additional assignments to students who are ahead of the course expectations Technology Integration ● Students will program using a Java IDE ○ IntelliJ ○ Repl.it ● Online questions and activities on CollegeBoard Global Perspectives	● Programs and practice code ● Free Response Questions from previous AP Exams ● Teacher provided feedback during programming projects ● Practice multiple choice questions from CollegeBoard ● AP Exam Style Test on Unit 6 (Arrays)

- Throughout the course students will discuss how programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact

Unit 7: ArrayLists

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

Students will understand...

- Advantages and disadvantages of ArrayLists compared to Arrays
- How to choose the best data structure for the application
- How to traverse and manipulate ArrayLists by creating program specific algorithms

Essential Questions

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

- How are ArrayLists different from Arrays?
- How are ArrayLists similar to Arrays?
- How can I choose the appropriate data structure for my application?
- How can we efficiently search through and sort data saved in data structures?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- ArrayList is a class that uses an Array as an instance variable. It contains methods to make it easier for the programmer to use the data saved in the array
- ArrayLists can only hold reference types, but we can use wrapper classes when we want to save primitive values.
- The type of ArrayList is set using a type parameter, ex: `ArrayList<String> = new ArrayList<String>();`
- ArrayLists can be constructed without a type parameter, but it is safer to use the parameter because it can help catch errors while compiling instead of while running
- Traversing ArrayLists can be done with for and for-each loops, but the size of an ArrayList cannot be modified with a for-each or a `ConcurrentModificationException` will be thrown
- Algorithms for Arrays will work with ArrayLists, but we can create more efficient algorithms with ArrayLists if adding or removing elements.

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

Students will:

- Compare data structures to find the most appropriate for the application
- Write algorithm for ArrayLists
- Search and Sort Arrays and ArrayLists efficiently

Suggested Topics/Lessons

- What is an ArrayList?
 - Creation and access
 - Differences from Arrays
- ArrayList Methods
 - Review reference guide

- Create a program that combines all topics learned so far

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

- Determine output given a code segment
- Write code using each method on the reference guide
- Traversing ArrayLists
 - Determine output given a code segment
 - Write for and for-each loops to traverse ArrayLists
- Develop Algorithms using ArrayLists
 - Modify Standard Array Algorithms to work for ArrayLists
 - Find errors in algorithms
 - Standard Algorithms: insert elements, delete elements
- Searching and Sorting Arrays and ArrayLists
 - Sequential Search
 - Binary Search
 - Insertion Sort
 - Selection Sort
- Ethics of Data Collection
 - Discussion groups
- Programming Project(s)
 - Create a StringList class that behaves like an ArrayList but only holds Strings
 - Practice with ArrayList Methods
 - Create a Utilities class containing static methods for searching and sorting Arrays and ArrayLists
 - Theme Park Program: Ride, Park, and Customer classes. Students use whichever data structure is appropriate to accomplish specific tasks

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 ● Partner students based on ability level ● Provide additional assignments to students who are ahead of the course expectations Technology Integration ● Students will program using a Java IDE ○ IntelliJ ○ Repl.it ● Online questions and activities on CollegeBoard Global Perspectives ● Throughout the course students will discuss how	● Programs and practice code ● Free Response Questions from previous AP Exams ● Teacher provided feedback during programming projects ● Practice multiple choice questions from CollegeBoard ● AP Exam Style Test on Units 6 and 7 (Arrays and ArrayLists)

programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact

Unit 8: 2D Arrays

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

Students will understand...

- An Array containing Arrays is referred to as a "Two Dimensional" or "2D" Array
- 2D Arrays can be used to represent pixels in an image, points in a coordinate plane, or other two dimensional objects

Essential Questions

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

- What is a 2D Array?
- How are elements of a 2D Array accessed?
- What can we do with 2D Arrays that would have been difficult or impossible without them?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- A 2D Array is really Arrays stored within an Array
- 2D Arrays can be represented as either row major or column major order
- Elements can be accessed using two sets of square brackets [] []. The first is the index for the outer array, and the second is the index for the inner array (on AP Exam this will always be [row][column] - called "row major order")
- Traversals require nested loops
- Traversals can be either row or column major
- Array algorithms can be applied to each inner array of a 2D Array

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

Students will:

- Create 2D Arrays
- Traverse 2D Arrays
- Write algorithms for 2D Arrays
- Represent digital images with 2D Arrays

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.

Suggested Topics/Lessons

- What is a 2D Array?
 - 2D Array creation
 - Row and Column Major Order (Exam only tests row major)
 - AP Exam only uses rectangular arrays
 - Initializer lists
- Traversing 2D Arrays
 - Row major traversal (of row major order arrays)
 - Column major traversal (of row major order arrays)
- Programming Project(s)
 - Picture Lab from College board

- **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.

- Student and Classroom class where the classroom contains a 2D Array of Students with some null elements, column and row major traversals

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 ● Partner students based on ability level ● Provide additional assignments to students who are ahead of the course expectations Technology Integration ● Students will program using a Java IDE ○ IntelliJ ○ Repl.it ● Online questions and activities on CollegeBoard Global Perspectives ● Throughout the course students will discuss how programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact	● Programs and practice code ● Free Response Questions from previous AP Exams ● Teacher provided feedback during programming projects ● Practice multiple choice questions from CollegeBoard ● AP Exam Style Test on Unit 8 (2D Arrays)

Unit 9: Inheritance

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

Students will understand...

- How one class can inherit properties from another
- Inheritance always happens in Java whenever a class is created
- Objects can be saved to variables of their super type

Essential Questions

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

- What do classes inherit from superclasses?
- Why is it helpful for classes to be created as subclasses?
- Which method gets called when the same method exists in a subclass and its superclass?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Subclasses inherit all methods from superclasses
- Only the default constructor is inherited from the superclass (only if the superclass has a default constructor)
- Private variables and methods exists in a subclass, but cannot be directly accessed by the subclass
- A method can be overridden (replaced) in a subclass by creating a new method with the same signature as the superclass's method
- this keyword super can be used to refer to the superclass. ex: super.toString will call the toString method from the superclass
- If S is a subclass of T then S[] is a subclass of T[] and ArrayList<S> is a subclass of ArrayList<T>
- When code is compiled not all errors are caught, the compiler only verifies that the methods are called on appropriate variables and the parameters are correct
- When the code is run errors may come up where an Object does not have a specific method, because it is saved in a variable of a different type
- Casting can be used to convince the compiler that a specific subtype is saved in a supertype variable

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

Students will:

- Create a class that is a subclass of an existing class
- Create a subclass of a student created class

Suggested Topics/Lessons

- Creating Superclasses and Subclasses
 - Object Class
 - toString and equals method

- Explain how to tell which methods are executed when the method exists in both the super and subclass
- Create a program utilizing subclasses

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.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.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.

- Writing Constructors for Subclasses
 - Default constructor
 - `super()`
 - Describe the behavior of constructors in subclasses
- Overriding Methods
 - Describe the behavior of overridden methods
 - find errors in overriding methods
- `super` Keyword
 - `super` vs. `this`
 - variable and method access
 - when `super` is necessary and when it is optional
- Creating References using Inheritance Hierarchies
 - `instanceof` operator
 - if `String[]` is a subclass of `Object[]`
- Polymorphism
 - Describe the behavior of methods
 - How to know which code is executed
 - run time vs compile time errors
- Programming Project(s)
 -

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 ● Partner students based on ability level ● Provide additional assignments to students who are ahead of the course expectations Technology Integration ● Students will program using a Java IDE ○ IntelliJ ○ Repl.it ● Online questions and activities on CollegeBoard Global Perspectives ● Throughout the course students will discuss how programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact	● Programs and practice code ● Free Response Questions from previous AP Exams ● Teacher provided feedback during programming projects ● Practice multiple choice questions from CollegeBoard ● AP Exam Style Test on Unit 9 (Subclasses)

Unit 10: Recursion

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

Students will understand...

- A recursive method is a method that calls itself
- Recursion is an alternative to a loop as a way to repeat code
- Recursion has advantages and disadvantages compared to loops

Essential Questions

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

- What are the advantages of recursion?
- How can a process be written recursively?
- How can we prevent StackOverflow Errors?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Any process that can be written recursively can be written as a loop, and a process that can be written as a loop can be written recursively
- Recursion is a way to repeat certain code segments by writing a method that calls itself
- A base case is any way to get the recursive method to stop, it is required to prevent infinite recursion
- Some processes are naturally recursive, and may be easier for the programmer to write recursively instead of iteratively; Tower of Hanoi, and Fibonacci numbers are recursive processes
- MergeSort is a process to sort values in a data structure where the values are recursively split into two sections, sorted, then merged together in order. This process is shorter to write recursively than it is to write iteratively.
- Binary Searching can be written recursively, although the code is very similar to writing it as a loop

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

Students will:

- Trace recursive methods to predict the output
- Write recursive algorithms
- Search and Sort data sets recursively

Suggested Topics/Lessons

- Recursion
 - Tracing recursive method class
 - Writing recursive methods as an alternative to loops
 - Methods that are easier to write with recursion. Ex: fibonacci 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.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.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.

- StackOverflow Error
- Recursive Searching and Sorting
 - Binary Search
 - MergeSort
- Programming Project(s)
 - Practice writing recursive methods to accomplish specific tasks
 - Add to previously created Utilities class to implement binary searching and mergesort recursively

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 ● Partner students based on ability level ● Provide additional assignments to students who are ahead of the course expectations Technology Integration ● Students will program using a Java IDE ○ IntelliJ ○ Repl.it ● Online questions and activities on CollegeBoard Global Perspectives ● Throughout the course students will discuss how programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact	● Programs and practice code ● Free Response Questions from previous AP Exams ● Teacher provided feedback during programming projects ● Practice multiple choice questions from CollegeBoard ● AP Exam Style Test on Units 3 and 4 (If Statements and Loops)

Unit 11: AP Exam Review

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

Students will be prepared to take the AP Exam

Essential Questions

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

- What were the important concepts covered this year?
- What do I need to work on before the AP Exam?

Enduring Understandings

What will students understand about the big ideas?

Students will understand...

- The breakdown of topics on the AP Exam
 - Primitive types (2.5-5%)
 - Using Objects (5-7.5%)
 - Boolean Expressions and if Statements (15-17.5%)
 - Iteration (17.5-22.5%)
 - Writing Classes (5-7.5%)
 - Arrays (10-15%)
 - ArrayLists (2.5-7.5%)
 - 2D Arrays (7.5-10%)
 - Inheritance (5-10%)
 - Recursion (5-7.5% Multiple Choice Only)
- The exam format:
 - 40 Multiple choice questions with 5 answer choices (90 minutes)
 - 4 Free response questions (90 minutes)
 - Methods and Control Structures
 - Classes
 - Array or ArrayList
 - 2D Array

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

Students will review material learned during the year to prepare for the AP Exam

NJ Computer Science and Design Thinking Standards

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

Suggested Topics/Lessons

- Practice sets of 4 free response questions
- Practice multiple choice questions with topics weighted like the AP Exam
- Practice timed in class AP Exam from CollegeBoard

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 ● Partner students based on ability level ● Provide additional assignments to students who are ahead of the course expectations Technology Integration ● Students will program using a Java IDE ○ IntelliJ ○ Repl.it ● Online questions and activities on CollegeBoard Global Perspectives ● Throughout the course students will discuss how programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact	● Practice Questions ● Practice AP Exam ○ This should be scored like a real AP Exam so students know how they would have done if this were the actual AP Exam

Unit 12: Final Project

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

Students will understand...

- The design process for a program
- How to choose the appropriate tool to accomplish a specific task
- Multiple tools can be combined to create a new tool
- How to create a Graphical User Interface (GUI) in Java

Essential Questions

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

- How can I create a large program from the beginning?
- How do we decide which class is responsible for what tasks?

Enduring Understandings

What will students understand about the big ideas?

Students will understand that:

- Creating a plan for a program is a key step to creating an effective program
- Large tasks should be broken down into smaller manageable tasks to make code easier to write and read once it is complete

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

Students will:

- Create a program that can draw shapes in different colors
- Write code to allow the user to click on the screen
- Create a program with a GUI in groups

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

Suggested Topics/Lessons

In this unit students will work in small groups to complete one program (Suggested goal: Tic Tac Toe using the mouse as input)

Small lesson on the Graphics class, and MouseListener Interface

As a class students should agree on how to break up the tasks for the program, and what should be accomplished at each stage

Most of the work during this unit should be done by the students

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 ● Partner students based on ability level ● Provide additional assignments to students who are ahead of the course expectations Technology Integration ● Students will program using a Java IDE ○ IntelliJ ○ Repl.it ● Online questions and activities on CollegeBoard Global Perspectives ● Throughout the course students will discuss how programming topics have had an impact on our society and how to use these skills responsibly to have a positive impact	● Project updates ● Teacher provided feedback during programming ● Graded final program