

Unit 5 Array List

Content Area: **21st Century Life and Careers**
Course(s): **Sample Course**
Time Period: **NovDec**
Length: **20 days Grades 10-12**
Status: **Published**

Array List, AP Computer Science A

Department of Curriculum and Instruction



Belleville Public Schools

Curriculum Guide

AP Computer Science A - Java

Array List

Belleville Board of Education

56 Ralph Street

Belleville, NJ 07109

Prepared by: Teacher, Corey Woodring

Dr. Richard Tomko, Ph.D., M.J., Superintendent of Schools

Ms. LucyAnn Demikoff, Director of Curriculum and Instruction K-12

Ms. Nicole Shanklin, Director of Elementary Education K-8,

Mr. Joseph Lepo, Director of Secondary Education

Board Approved:

Unit Overview

As students learned in the previous unit, data structures are helpful when storing multiple related data values. Arrays have a static size, which causes limitations related to the number of elements stored, and it can be challenging to reorder elements stored in arrays. The ArrayList object has a dynamic size, and the class contains methods for insertion and deletion of elements, making reordering and shifting items easier. Deciding which data structure to select becomes increasingly important as the size of the data set grows, such as when using a large real-world data set. In this unit, students will also learn about privacy concerns related to storing large amounts of personal data and about what can happen if such information is compromised.

Students need to consider the impact using ArrayList rather than an array has on the structure of their program code. This includes considering the use of ArrayList methods and the flexibility of a structure with a dynamic size. For instance, the use of an ArrayList will require students to analyze program code that uses method calls. Providing students with practice writing programs for data sets of undetermined sized—or at least larger than they would be able to analyze easily by hand—presents a more relevant and realistic experience with data. Additionally, this requires students to focus more on the algorithm and ensuring that it will work in all situations rather than on an individual result.

With larger data sets, programmers become concerned with the amount of time it will take for their program code to run. Students should have practice determining the number of times a code segment executes; this can help them gain an idea of how long it will take to run a program on a data set of a given size

Essential Questions

BIG IDEA 1 Variables - Why is an ArrayList more appropriate for storing your music playlist, while an array might be more appropriate for storing your class schedule?

BIG IDEA 2 Control - How can we use statement execution counts to choose appropriate algorithms?

BIG IDEA 3 Impact of Computing - What personal data is currently being collected, and how?

Exit Skills

Represent collections of related object reference data using ArrayList objects.

Determine code that would be used to complete code segments.

Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

New Jersey Student Learning Standards (NJSL-S)

CS.9-12.8.1.12.CS.1

Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.

CS.9-12.8.1.12.CS.2

Model interactions between application software, system software, and hardware.

| | |
|---------------------|---|
| CS.9-12.8.1.12.NI.1 | Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. |
| CS.9-12.CS | Computing Systems |
| CS.9-12.NI | Networks and the Internet |
| TECH.9.4.12.CI | Creativity and Innovation |
| TECH.9.4.12.CI.1 | Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a). |
| TECH.9.4.12.CI.2 | Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8). |
| TECH.9.4.12.CI.3 | Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1). |
| TECH.9.4.12.CT | Critical Thinking and Problem-solving |
| TECH.9.4.12.CT.1 | Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3). Collaboration with individuals with diverse experiences can aid in the problem-solving process, particularly for global issues where diverse solutions are needed. The scalability and reliability of the Internet are enabled by the hierarchy and redundancy in networks. Network topology is determined by many characteristics. The usability, dependability, security, and accessibility of devices within integrated systems are important considerations in their design as they evolve. Innovative ideas or innovation can lead to career opportunities. A computing system involves interaction among the user, hardware, application software, and system software. |

Interdisciplinary Connections

| | |
|----------------|---|
| LA.RL.11-12 | Reading Literature Key Ideas and Details |
| LA.RL.11-12.1 | Cite strong and thorough textual evidence and make relevant connections to support analysis of what the text says explicitly as well as inferences drawn from the text, including determining where the text leaves matters uncertain. |
| LA.RL.11-12.2 | Determine two or more themes or central ideas of a text and analyze their development over the course of the text, including how they interact and build on one another to produce a complex account; provide an objective summary of the text. |
| LA.RL.11-12.3 | Analyze the impact of the author's choices regarding how to develop and relate elements of a story or drama (e.g., where a story is set, how the action is ordered, how the characters are introduced and developed). |
| LA.L.11-12.4 | Determine or clarify the meaning of unknown and multiple-meaning words and phrases based on grades 11–12 reading and content, choosing flexibly from a range of strategies. |
| LA.L.11-12.4.A | Use context (e.g., the overall meaning of a sentence, paragraph, or text; a word's position or function in a sentence) as a clue to the meaning of a word or phrase. |
| LA.L.11-12.4.B | Identify and correctly use patterns of word changes that indicate different meanings or parts of speech (e.g., conceive, conception, conceivable). |
| LA.L.11-12.4.C | Consult general and specialized reference materials (e.g., dictionaries, glossaries, |

thesauruses), both print and digital, to find the pronunciation of a word or determine or clarify its precise meaning, its part of speech, its etymology, or its standard usage.

LA.L.11-12.4.D

Verify the preliminary determination of the meaning of a word or phrase (e.g., by checking the inferred meaning in context or in a dictionary).

Learning Objectives

Represent collections of related object reference data using ArrayList objects.

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

The ArrayList class is part of the java. util package. An import statement can be used to make this class available for use in the program.

Suggested Activities & Best Practices

ArrayList Suggested Activities

1. Predict and compare, Have students look at the code they wrote to solve the free-response question in Unit 6 (or other code from Unit 6) on paper, and have them rewrite it using an ArrayList. Have them highlight the parts that need to be changed and determine how to change them. Then, have students type up the changes in an IDE and confirm that the program still works as expected. 2 7.1–7.5
2. Identify a subtask, Have students read through an ArrayList-based free-response question in groups, and have them identify all subtasks. These subtasks could be conditional statements, iteration, or even other methods. Once the subtasks have been identified, divide the subtasks among the group members, and have students implement their given subtask. When all students are finished, have them combine the subtasks into a single solution. 3 7.5
3. Discussion group, Discuss the algorithm necessary to search for the smallest value in an ArrayList. Without explaining what you are doing, change the Boolean expression so that it will find the largest value, and ask students to describe what the resulting algorithm will do. Then, change the algorithm to store and return the location of the largest value, and discuss the change
4. -Student will be able to identify famous African Americans in the field of computer programming
5. Students will also be introduced to the basic energy efficient models in the programming field to help reduce global warming

Assessment Evidence - Checking for Understanding (CFU)

- Complete various written checkpoint exercises that focus on the explanation and description of computer hardware and Java basics.-Alternate Assessment
- Develop a visual representation of the communication processes within a computer using appropriate terminology.
- Properly document a program using correct indentation, spacing, and comment style.
- Debug programs and determine the types of errors in the program.
- Create programs based on programming exercises that display various types of output using string and numeric expressions.-Benchmark Assessment

| Formative Evaluations: | Summative Evaluations: |
|-----------------------------------|---------------------------------------|
| Formative Assessment with polling | Unit Test/ReTest-Summative Assessment |
| codeIt! Nows | |
| Quizzes | |
| Long Programs (LP)/Lab Work | |
| Components of AP approved Lab | |

Self Check Exemplar

BJP4 Self-Check 10.15: ArrayListMystery1

Language/Type: Java [ArrayList Collections mystery](#)
Author: Roy McElmurry (on 2016/09/08)

Consider the following method:

```
public static void mystery1(ArrayList<Integer> list) {  
    for (int i = list.size() - 1; i > 0; i--) {  
        if (list.get(i) < list.get(i - 1)) {  
            int element = list.get(i);  
            list.remove(i);  
            list.add(0, element);  
        }  
    }  
    System.out.println(list);  
}
```

Write the output produced by the method when passed each of the following ArrayLists:

```
[2, 6, 1, 8]  
[30, 20, 10, 60, 50, 40]  
[-4, 16, 9, 1, 64, 25, 36, 4, 49]
```

Upon completion of this section, please remove all remaining descriptions, notes, outlines, examples and/or illustrations that are not needed or used.

By identifying the **Evidence of Student Learning with Checking for Understanding (CFU)** techniques used during the lesson and/or for Closure (Madeline Hunter), please list the variety of means used to assess students' learning (e.g. quizzes, tests,

academic prompts, observations, homework, journals).

- Admit Tickets
- Anticipation Guide
- Common Benchmarks
- Compare & Contrast
- Create a Multimedia Poster
- DBQ's
- Define
- Describe
- Evaluate
- Evaluation rubrics
- Exit Tickets
- Explaining
- Fist- to-Five or Thumb-Ometer
- Illustration
- Journals
- KWL Chart
- Learning Center Activities
- Multimedia Reports
- Newspaper Headline
- Outline
- Question Stems
- Quickwrite
- Quizzes
- Red Light, Green Light
- Self- assessments
- Socratic Seminar
- Study Guide
- Surveys
- Teacher Observation Checklist
- Think, Pair, Share
- Think, Write, Pair, Share
- Top 10 List
- Unit review/Test prep
- Unit tests
- Web-Based Assessments
- Written Reports

Primary Resources & Materials

NJCTL Curriculum

<https://runestone.academy/runestone/books/published/csawesome/Unit7-ArrayList/topic-7-2-arraylist-methods.html?highlight=array%20list>

Curriculum development Resources/Instructional Materials/Equipment Needed Teacher Resources: •
www.gliffy.com • Eclipse IDE • MS DOS Prompt • Computers

Ancillary Resources

Java Resources

- [Java Review for the AP CS A Exam - Great review site with lots of practice questions.](#)
- [Aplus Compter Science Exam Review Material -Slide, Free Response, and more!](#)
- [Introduction to Java - a textbook for a first course in computer science for the next generation of scientists and engineers](#)
- [Guru-99 Introduction Java Material](#)
- [Oracles \(owners of Java\) has their own tutorials](#)
- [Dick Baldwin - ACC - Introduction and Advanced Java Material](#)
- [Introduction to Computer Science using Java - by Bradley Kjell](#)
- [Thinking in Java](#)
- [Blue Pelican Java](#)
- [Java Coding Bat - Lots of good practice problems](#)
- [Code Academy - No Java but good practice.](#)

<https://chortle.ccsu.edu/CS151/cs151java.html>

Technology Infusion

Technology Infusion and/or strategies include chromebooks online materials google/powerpoint slides

Technology Infusion and/or strategies are integrated into this unit to enhance learning

Alignment to 21st Century Skills & Technology

| | |
|-------------------|---|
| WRK.9.2.12.CAP | Career Awareness and Planning |
| WRK.9.2.12.CAP.1 | Analyze unemployment rates for workers with different levels of education and how the economic, social, and political conditions of a time period are affected by a recession. |
| WRK.9.2.12.CAP.2 | Develop college and career readiness skills by participating in opportunities such as structured learning experiences, apprenticeships, and dual enrollment programs. |
| WRK.9.2.12.CAP.3 | Investigate how continuing education contributes to one's career and personal growth. |
| WRK.9.2.12.CAP.13 | Analyze how the economic, social, and political conditions of a time period can affect the labor market. |
| WRK.9.2.12.CAP.14 | Analyze and critique various sources of income and available resources (e.g., financial assets, property, and transfer payments) and how they may substitute for earned income. |
| WRK.9.2.12.CAP.15 | Demonstrate how exemptions, deductions, and deferred income (e.g., retirement or medical) can reduce taxable income. |
| TECH.9.4.12.CI | Creativity and Innovation |
| TECH.9.4.12.CI.1 | Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a). |
| TECH.9.4.12.CI.2 | Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8). |
| TECH.9.4.12.CI.3 | Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1). |
| TECH.9.4.12.CT | Critical Thinking and Problem-solving |
| | With a growth mindset, failure is an important part of success. |
| | Innovative ideas or innovation can lead to career opportunities. |

21st Century Skills/Interdisciplinary Themes

21st century themes: The unit will integrate the 21st Century Life and career standard 9.1 strands A-D. These strands include: critical thinking and problem solving, creativity and innovation, collaboration, teamwork, and leadership, and cross cultural understanding and interpersonal communication

21st Century/Interdisciplinary Themes that will be incorporated into this unit.

- Communication and Collaboration
- Creativity and Innovation
- Critical thinking and Problem Solving
- ICT (Information, Communications and Technology) Literacy
- Information Literacy
- Life and Career Skills

- Media Literacy

9.3.12.BM-OP.3

Apply inventory tracking systems to facilitate operational controls.

12.9.3.IT.12

Demonstrate knowledge of the hardware components associated with information systems.

21st Century Skills

21st Century Skills that will be incorporated into this unit.

- Civic Literacy
- Environmental Literacy
- Financial, Economic, Business and Entrepreneurial Literacy
- Global Awareness
- Health Literacy

CAEP.9.2.12.C

Career Preparation

CAEP.9.2.12.C.1

Review career goals and determine steps necessary for attainment.

CAEP.9.2.12.C.3

Identify transferable career skills and design alternate career plans.

Differentiation

• Technology Resources • Teacher Tutoring • Peer Tutoring • Cooperative Learning Groups • Differentiated Instruction • Follow all IEP Modifications/504 Plan

Provide study guides, worksheets, and notes

Differentiations:

- Small group instruction
- Small group assignments
- Extra time to complete assignments
- Pairing oral instruction with visuals
- Repeat directions
- Use manipulatives

- Center-based instruction
- Token economy
- Study guides
- Teacher reads assessments allowed
- Scheduled breaks
- Rephrase written directions
- Multisensory approaches
- Additional time
- Preview vocabulary
- Preview content & concepts
- Story guides
- Behavior management plan
- Highlight text
- Student(s) work with assigned partner
- Visual presentation
- Assistive technology
- Auditory presentations
- Large print edition
- Dictation to scribe
- Small group setting

Hi-Prep Differentiations:

Exemplar: Students will be given a choice of 3 activities, 2 teacher designed and 1 design your own.

Alternative formative and summative assessments

- Choice boards
- Games and tournaments
- Group investigations
- Guided Reading
- Independent research and projects
- Interest groups
- Learning contracts
- Leveled rubrics
- Literature circles
- Multiple intelligence options
- Multiple texts
- Personal agendas
- Project-based learning
- Problem-based learning
- Stations/centers
- Think-Tac-Toes
- Tiered activities/assignments
- Tiered products
- Varying organizers for instructions

Lo-Prep Differentiations

Exemplar:

- **Mini workshops to re-teach or extend skills**
- Choice of books or activities
- Cubing activities
- Exploration by interest
- Flexible grouping
- Goal setting with students
- Jigsaw
- Mini workshops to re-teach or extend skills
- Open-ended activities
- Think-Pair-Share
- Reading buddies
- Varied journal prompts
- Varied supplemental materials

Special Education Learning (IEP's & 504's)

Please identify the **Special Education Learning** adaptations that will be employed in the unit, using the ones identified below.

Exemplar: Work with a partner

Adjust the method of presentation or content.

- **Develop** supplemental material.

Special Education Learning adaptations that will be employed in the unit, using the ones identified below.

- printed copy of board work/notes provided

- additional time for skill mastery
- assistive technology
- behavior management plan
- Center-Based Instruction
- check work frequently for understanding
- computer or electronic device utilizes
- extended time on tests/ quizzes
- have student repeat directions to check for understanding
- highlighted text visual presentation
- modified assignment format
- modified test content
- modified test format
- modified test length
- multi-sensory presentation
- multiple test sessions
- preferential seating
- preview of content, concepts, and vocabulary
- Provide modifications as dictated in the student's IEP/504 plan
- reduced/shortened reading assignments
- Reduced/shortened written assignments
- secure attention before giving instruction/directions
- shortened assignments
- student working with an assigned partner
- teacher initiated weekly assignment sheet
- Use open book, study guides, test prototypes

English Language Learning (ELL)

Exemplar:

Extended Time

It's obvious that response time for ELLs is significantly greater than it is for students proficient in English. Given this, we know that ELLs may require more [time](#) to process and communicate information on assessments. To support your students in this area, give them additional time on tests to help. Extra time will also help to decrease anxiety, which often has a significant impact on test performance.

English Language Learning adaptations that will be employed in the unit, using the ones identified below.

- teaching key aspects of a topic. Eliminate nonessential information
- using videos, illustrations, pictures, and drawings to explain or clarify

- allowing products (projects, timelines, demonstrations, models, drawings, dioramas, poster boards, charts, graphs, slide shows, videos, etc.) to demonstrate student's learning;
- allowing students to correct errors (looking for understanding)
- allowing the use of note cards or open-book during testing
- decreasing the amount of work presented or required
- having peers take notes or providing a copy of the teacher's notes
- modifying tests to reflect selected objectives
- providing study guides
- reducing or omitting lengthy outside reading assignments
- reducing the number of answer choices on a multiple choice test
- tutoring by peers
- using computer word processing spell check and grammar check features
- using true/false, matching, or fill in the blank tests in lieu of essay tests

At Risk

Exemplar:

Diagnosing the specific learning needs of individual students and using that information to provide targeted [academic support](#) or more [personalized learning experiences](#), is the best way to support students who are at risk

A Strategy Summary

Planned interventions are far superior to remediation approaches. Always plan to address students at risk in your learning tasks, instructions, and directions. Try to anticipate where the needs will be and then address them. Intervene as much as possible to support students at risk. If your intervention strategies are working, continue to use them. If they're not working, plan for new interventions that will help students succeed.

Please identify Intervention Strategies that will be employed in the unit, using the ones identified below.

- allowing students to correct errors (looking for understanding)
- teaching key aspects of a topic. Eliminate nonessential information
- allowing products (projects, timelines, demonstrations, models, drawings, dioramas, poster boards, charts, graphs, slide shows, videos, etc.) to demonstrate student's learning
- allowing students to select from given choices
- allowing the use of note cards or open-book during testing
- collaborating (general education teacher and specialist) to modify vocabulary, omit or modify items to reflect objectives for the student, eliminate sections of the test, and determine how the grade will be determined prior to giving the test.
- decreasing the amount of work presented or required

- having peers take notes or providing a copy of the teacher's notes
- marking students' correct and acceptable work, not the mistakes
- modifying tests to reflect selected objectives
- providing study guides
- reducing or omitting lengthy outside reading assignments
- reducing the number of answer choices on a multiple choice test
- tutoring by peers
- using authentic assessments with real-life problem-solving
- using true/false, matching, or fill in the blank tests in lieu of essay tests
- using videos, illustrations, pictures, and drawings to explain or clarify

Talented and Gifted Learning (T&G)

Exemplar:

Alternate assignments for high-ability students can either be projects related to the modified area of study that extend the curriculum, or they can be independent projects that are chosen based on students' individual interests.

http://www.grandviewlibrary.org/CurriculumAdaptations/General_Gifted.pdf

Grouping • Group gifted students with other gifted students or higher-level learners. • Refrain from grouping gifted students with lower-level students for remediation.

Please identify the **Talented and Gifted** adaptations that will be employed in the unit, using the ones identified below.

- Above grade level placement option for qualified students
- Advanced problem-solving
- Allow students to work at a faster pace
- Cluster grouping
- Complete activities aligned with above grade level text using Benchmark results
- Create a blog or social media page about their unit
- Create a plan to solve an issue presented in the class or in a text
- Debate issues with research to support arguments
- Flexible skill grouping within a class or across grade level for rigor

- Higher order, critical & creative thinking skills, and discovery
- Multi-disciplinary unit and/or project
- Teacher-selected instructional strategies that are focused to provide challenge, engagement, and growth opportunities
- Utilize exploratory connections to higher-grade concepts
- Utilize project-based learning for greater depth of knowledge

Sample Lesson

<https://runestone.academy/runestone/books/published/csawesome/Unit7-ArrayList/topic-7-2-arraylist-methods.html?highlight=array%20list>

Lesson Plan – ArrayLists

| | | | |
|----------|-----------------------|-------------|------------------------|
| Teacher: | Corey Woodring | Time Frame: | 13 days |
| Grade: | 9-12 | School: | Belleville High School |
| Subject: | AP Computer Science A | | |

AP Essential Knowledge

(Referenced from CollegeBoard AP CS A Course & Exam Description)

- Represent collections of related object reference data using ArrayList objects.
- An ArrayList object is mutable and contains object references.
- The ArrayList constructor ArrayList() constructs an empty list.
- Java allows the generic type ArrayList, where the generic type E specifies the type of the elements.
- When ArrayList is specified, the types of the reference parameters and return type when using the methods are type E.
- ArrayList is preferred over Arraylist because it allows the compiler to find errors that would otherwise be found at run-time.
- The ArrayList class is part of the java. util package. An import statement can be used to make this class available for use in the program.
- The following ArrayList methods— including what they do and when they are used—are part of the Java Quick Reference:
 - a. int size() - Returns the number of elements in the list
 - b. boolean add(E obj) - Appends obj to end of list; returns true
 - c. void add(int index, E obj) - Inserts obj at position index (0 <= index <= size), moving elements at position index and higher to the right (adds 1 to their indices) and adds 1 to size

- d. E get(int index) - Returns the element at position index in the list
 - e. E set(int index, E obj) — Replaces the element at position index with obj; returns the element formerly at position index
 - f. E remove(int index) — Removes element from position index, moving elements at position index + 1 and higher to the left (subtracts 1 from their indices) and subtracts 1 from size; returns the element formerly at position index
- For ArrayList objects: Traverse using a for or while loop and Traverse using an enhanced for loop.
 - Iteration statements can be used to access all the elements in an ArrayList. This is called traversing the ArrayList.
 - Deleting elements during a traversal of an ArrayList requires using special techniques to avoid skipping elements.
 - Since the indices for an ArrayList start at 0 and end at the number of elements – 1, accessing an index value outside of this range will result in an ArrayIndexOutOfBoundsException being thrown.
 - Changing the size of an ArrayList while traversing it using an enhanced for loop can result in a ConcurrentModificationException being thrown. Therefore, when using an enhanced for loop to traverse an ArrayList, you should not add or remove elements.
 - For algorithms in the context of a particular specification that requires the use of ArrayList traversals: Identify standard algorithms, Modify standard algorithms, Develop an algorithm.
 - There are standard ArrayList algorithms that utilize traversals to: Insert elements, Delete elements, and apply the same standard algorithms that are used with 1D arrays.
 - Some algorithms require multiple String, array, or ArrayList objects to be traversed simultaneously
 - Apply sequential/linear search algorithms to search for specific information in array or ArrayList objects.
 - There are standard algorithms for searching.
 - Sequential/linear search algorithms check each element in order until the desired value is found or all elements in the array or ArrayList have been checked.

Enduring Understanding & CTP Skills

(Referenced from CollegeBoard AP CS A Course & Exam Description)

VAR-2 To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

CON-2 Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

1.B Determine code that would be used to complete code segments.

2.C Determine the result or output based on the statement execution order in a code segment containing

method calls.

2.D Determine the number of times a code segment will execute.

3.D Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.

4.A Use test-cases to find errors or validate results.

5.C Explain how the result of program code changes, given a change to the initial code.

Essential Questions

(Some referenced from CollegeBoard AP CS A Course & Exam Description)

(What questions will the student be able to answer as a result of the instruction?)

1. Why is an ArrayList more appropriate for storing your shopping list, while an array might be more appropriate for storing your class schedule?
2. How can we use statement execution counts to choose appropriate algorithms?
3. What are the standard algorithms for searching an ArrayList?

Assessment

(What is acceptable evidence to show desired results (rubrics, exam, etc.)? Attach Copy

- Complete various written checkpoint exercises that focus on the explanation and description of computer hardware and Java basics.
- Develop a visual representation of the communication processes within a computer using appropriate terminology.
- Properly document a program using correct indentation, spacing, and comment style.
- Debug programs and determine the types of errors in the program.
- Create programs based on programming exercises that display various types of output using string and numeric expressions.
- Unit 6 Assessment

Formative Evaluations:

Formative Assessment with polling

Summative Evaluations:

Unit 6 Test/ReTest

codeIt! Nows

AP Classroom AP
Computer Science A
Progress Checks

Quizzes

AP Classroom AP Computer Science A Topic Questions

Long Programs (LP)/ Lab work

Sequence and Scope

| Day | Topic/Activities | CW-HW |
|-----|--|--|
| 1 | <ul style="list-style-type: none">• Changing Sizes of Arrays• codeIt! Now | Finish codeIt! Now |
| 2 | <ul style="list-style-type: none">• ArrayLists Basics• codeIt! Now | Finish codeIt! Now Problem #1 |
| 3 | <ul style="list-style-type: none">• More ArrayList Methods• codeIt! Now | Problem #2 |
| 4 | <ul style="list-style-type: none">• Finish More ArrayList Methods• codeIt! Now | Finish codeIt! Now, AP Classroom topic questions 7.1 – 7.4 |
| 5 | <ul style="list-style-type: none">• Quiz Question #1• Lab Work: LP Part 1 | Finish LP Part 1 |
| 6 | <ul style="list-style-type: none">• ArrayLists & Wrapper Classes• codeIt! Now | Problems #3 – 4, AP Classroom topic questions 2.8 |
| 7 | <ul style="list-style-type: none">• Lab Work: CW-HW #5 - 6 | Problems #7 - 8 |
| 8 | <ul style="list-style-type: none">• Quiz Question #2• Iteration | None |
| 9 | <ul style="list-style-type: none">• codeIt! Now | Problems #9 - 11 |
| 10 | <ul style="list-style-type: none">• Lab Work: CW-HW #12 - 14 | Problems #15 - 17 |
| 11 | <ul style="list-style-type: none">• Lab Work: LP Part 2 | Problems #18-20 |
| 12 | <ul style="list-style-type: none">• Lab Work: LP Part 2 | Problems #21-23 Study for Test |
| 13 | <ul style="list-style-type: none">• Unit 6 Assessment | None |

| | |
|---------------------|--|
| CS.9-12.8.1.12.CS.1 | Describe ways in which integrated systems hide underlying implementation details to simplify user experiences. |
| CS.9-12.8.1.12.CS.2 | Model interactions between application software, system software, and hardware. |
| CS.9-12.8.1.12.CS.3 | Compare the functions of application software, system software, and hardware. |
| CS.9-12.8.1.12.NI.2 | Evaluate security measures to address various common security threats. |
| CS.9-12.8.1.12.NI.3 | Explain how the needs of users and the sensitivity of data determine the level of security implemented. |
| CS.9-12.8.1.12.NI.4 | Explain how decisions on methods to protect data are influenced by whether the data is at rest, in transit, or in use. |
| CS.9-12.CS | Computing Systems |
| TECH.9.4.12.CI | Creativity and Innovation |
| TECH.9.4.12.CI.1 | Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a). |
| TECH.9.4.12.CI.2 | Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8). |
| TECH.9.4.12.CI.3 | Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1). |
| TECH.9.4.12.CT | Critical Thinking and Problem-solving |
| TECH.9.4.12.CT.1 | Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3). |
| TECH.9.4.12.CT.2 | Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a). |
| | A computing system involves interaction among the user, hardware, application software, and system software. |
| | With a growth mindset, failure is an important part of success. |
| | Innovative ideas or innovation can lead to career opportunities. |