Unit 4: Iteration

Content Area: **Template**
Course(s):
Time Period: **Full Year**
Length: **Full Year**
Status: **Published**

UNIT RATIONALE

This unit focuses on iteration using while and for loops. As you saw in Unit 3, Boolean expressions are useful when a program needs to perform different operations under different conditions. Boolean expressions are also one of the main components in iteration. This unit introduces several standard algorithms that use iteration. Knowledge of standard algorithms makes solving similar problems easier, as algorithms can be modified or combined to suit new situations. Iteration is used when traversing data structures such as arrays, ArrayLists, and 2D arrays. In addition, it is a necessary component of several standard algorithms, including searching and sorting, which will be covered in later units.

ESSENTIAL QUESTIONS

How does iteration improve programs and reduce the amount of program code necessary to complete a task?

What situations would warrant the use of one type of loop over another?

STANDARDS

NEW JERSEY STUDENT LEARNING STANDARDS: CONTENT AREA

SAVED

New Jersey (NJSLS) - Grades 9-12 - Computer Science and Design Thinking (2020)

8.1.12.AP.1:

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

8.1.12.AP.2:

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

8.1.12.AP.3:

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

8.1.12.AP.4:

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

8.1.12.AP.5:

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

8.1.12.AP.6:

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

CS.9-12.8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
CS.9-12.8.1.12.AP.2	Create generalized computational solutions using collections instead of repeatedly using simple variables.
CS.9-12.8.1.12.AP.3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
CS.9-12.8.1.12.AP.4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
CS.9-12.8.1.12.AP.5	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
CS.9-12.8.1.12.AP.6	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

NEW JERSEY STUDENT LEARNING STANDARDS: CAREER READINESS, LIFE LITERACIES AND KEY SKILLS

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).

NEW JERSEY STUDENT LEARNING STANDARDS: COMPUTER SCIENCE AND DESIGN THINKING

See content area standards.

PRE-ASSESSMENTS

Discussion questions:

How many times does an if statement check a condition? What happens after the condition is checked?

Consider the following code segment:

```
public static void main(String[] args)
{
    int countdown = 3;
    if(countdown > 0)
    {
        System.out.println(countdown);
        countdown--;
```

```
        }

        if(countdown == 0)
        {
            System.out.println("Countdown Complete");
        }
    }
```

How many additional if statements are needed to get the countdown to reach 0 and print “Countdown Complete”?

What if the countdown started at 10 or 100? Could you think of a better way to program something like this?

INSTRUCTIONAL PLAN

MODULE 4

LESSON 4.3

Lesson 4.3: Developing Algorithms Using Strings

In this lesson, students will learn how to develop algorithms using Strings. Students will traverse Strings using a for loop and the `print.length()` command.

Student Learning Intentions (SLI) WALT: (We are learning to…)	• Develop an algorithm using Strings • Find if one or more substrings has a particular property • Determine the number of substrings that meet specific criteria • Create a new string with the characters reversed
Student Learning Strategies	Code tracing Create a plan Error analysis Identify a subtask Look for a pattern Pair programming Predict and compare Simplify the problem Think aloud

Success Criteria	Check for understanding Completion of below activities AP Topic Questions
Formative Assessment (drives instructional decisions)	Teacher Observation Check for Understanding AP Topic Questions
Activities and Resources	• Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding. • Explore the *Traversing Strings* example. • Explore the *Replace Characters* example. • Explore the *Reverse String* example. • Complete the *Replace Letter* exercise. • Complete the *Password Checker* exercise. • Complete the *Finding Palindromes* exercise. • Complete the *Fixing Grammar* exercise. • Complete the *Teen Talk* exercise.
Suggested Modifications	See Lesson 4.1 above.

LESSON 4.2

Lesson 4.2: For Loops

In this lesson. students learn how to use for loops in their programs. The for loop allows students to repeat a specific part of code a fixed number of times.

Student Learning Intentions (SLI) WALT: (We are learning to…)	• Represent iterative processes using a for loop
Student Learning Strategies	Code tracing Create a plan Error analysis Identify a subtask Look for a pattern Pair programming Predict and compare Simplify the problem Think aloud
Success Criteria	Check for understanding Completion of below activities AP Topic Questions

Formative Assessment (drives instructional decisions)	Teacher Observation Check for Understanding AP Topic Questions
Activities and Resources	• Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding. • Explore the *For Loop* example. • Explore the *Countdown* example. • Explore the *Count By Twos* example. • Explore the *Print the Odds* example. • Complete the *Repeat 100 Times* exercise. • Complete the *Replace WHILE with FOR Loop* exercise. • Complete the *Replace FOR Loop with WHILE Loop* exercise • Complete the *Multiplication Table* exercise. • Complete the *Debugging: Working with Loops* handout. • Complete the *Infinite Loops* handout. • Complete the *The Fencepost Problem* handout.
Suggested Modifications	See Lesson 4.1 above.

LESSON 4.1

Lesson 4.1: Iteration

In this lesson, students will learn how and when to use a while loop. Repetitive code can be avoided by using a while loop. While loops are used to repeat a set of statements until a condition is met. This lesson corresponds with AP Computer Science A topic 4.1.

Student Learning Intentions (SLI) WALT: (We are learning to…)	• Represent iterative processes using a while loop • Execute a return or break statement inside an iteration statement to halt the loop and exit the method or constructor • Develop an algorithm • Identify if an integer is or is not evenly divisible by another integer • Determine a minimum or maximum value • Compute a sum, average, or mode
Student Learning Strategies	Code tracing Create a plan Error analysis Identify a subtask Look for a pattern Pair programming Predict and compare Simplify the problem Think aloud

Success Criteria	Check for understanding Completion of below activities AP Topic Questions
Formative Assessment (drives instructional decisions)	Teacher Observation Check for Understanding AP Topic Questions
Activities and Resources	• Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding. • Explore the *While Loop Countdown* example. • Explore the *Get Down to One* example. • Explore the *Running Average* example. • Complete the *Making Taffy* exercise. • Complete the *Guess the Number* exercise. • Complete the *Divisibility* exercise. • Complete the *Max and Min Values* exercise.
Suggested Modifications	**English Language Learners** **Adjusted Speech**: The teacher changes speech patterns to increase student comprehension. This could include facing the students, paraphrasing, clearly indicating the most important ideas, and speaking more slowly. **Visuals**: The teacher uses graphics, pictures, visuals, and manipulatives. This helps ELL students better understand and comprehend the subjects at hand. **Front-Loading Vocabulary**: The teacher front loads vocabulary This means providing students with a list of important vocabulary words they will need to know for a book, lesson, etc. prior to the lesson being taught. Including pictures to go with the vocabulary words is also very beneficial for the students. **Students with Individualized Education Plans/504s** **Chunking**: The teacher presents information in a way that make it easy for students to understand and remember. Chunking is based on the presumption that our working memory is easily overloaded by excessive detail. The best way to deliver information is to organize it into meaningful units. Because students with special needs get overloaded easily, chunking is a effective strategy to use with them. **Checking for Understanding**: It is important to constantly check for understanding, especially for students who have accommodations. Teachers want to make sure students

understand the concepts being covered in a way that makes sense to them.

Extra time: The teacher provides students with special needs extra time to complete work or answer questions. It is important to give students enough time to process their thoughts.

Oral Reading: The teacher will read work orally to students. Class work such as tests and literature circles may need to be read aloud to the student.

Gifted & Talented Strategies

Extensions/Enrichments: Teachers will provide gifted and talented students with extension/enrichment projects. Students will be challenged to further their understanding, to apply acquired knowledge, and/or to produce something in reference t acquired knowledge.

Modify/Change Activities: Teachers will monitor and modify activities to accommodate those students who need to be challenged further. Additional reading, problem-solving, writing, or project work is necessary for those students who are ready to move on at a rate more accelerated than their peers. In this way, G & T students are provided the same opportunity for support as special needs students.

Students at Risk of School Failure

Directions or Instructions: Make sure directions and/or instructions are given in limited numbers. Give directions/instructions verbally and in simple written format. Ask students to repeat the instructions or directions to ensure understanding occurs. Check back with the student to ensure he/she hasn't forgotten.

Peer Support: Peers can help build confidence in other students by assisting in peer learning. Many teachers use the 'ask 3 before me' approach. This is fine, however, a student at risk may have to have a specific student or two to ask. Set this up for the student so he/she knows who to ask for clarification before going to you.

Alternate or Modified Assignments: Always ask yourself, "How can I modify this assignment to ensure the students at risk are able to complete it?" Sometimes you'll simplify the task, reduce

the length of the assignment or allow for a different mode of delivery. For instance, many students may hand something in, the at-risk student may jot notes and give you the information verbally. Or, it just may be that you will need to assign an alternate assignment.

Increase One to One Time: When other students are working, always touch base with your students at risk and find out if they're on track or needing some additional support. A few minutes here and there will go a long way to intervene as the need presents itself.

Contracts: It helps to have a working contract between you and your students at risk. This helps prioritize the tasks that need to be done and ensure completion happens. Each day write down what needs to be completed, as the tasks are done, provide a checkmark or happy face. The goal of using contracts is to eventually have the student come to you for completion sign-offs

Hands On: As much as possible, think in concrete terms and provide hands-on tasks. This means a child doing math may require a calculator or counters. The child may need to tape record comprehension activities instead of writing them. A child may have to listen to a story being read instead of reading it him/herself.

Tests/Assessments: Tests can be done orally if need be. Break tests down in smaller increments by having a portion of the test i the morning, another portion after lunch and the final part the next day.

Seating: Seat students near a helping peer or with quick access to the teacher. Those with hearing or sight issues need to be close to the instruction which often means near the front.

LESSON 4.4

Lesson 4.4: Nested Iteration

In this lesson, students will learn about nested loops. Nested loops are when loops are placed inside other loops to create more complex programs. When a loop is nested inside another loop, the inner loop must complete all its iterations before the outer loop can continue. This lesson corresponds with AP Computer Science A topic 4.4.

Student Learning Intentions (SLI) WALT: (We are learning to…)	• Represent nested iterative processes
Student Learning Strategies	Code tracing Create a plan Error analysis Identify a subtask Look for a pattern Pair programming Predict and compare Simplify the problem Think aloud
Success Criteria	Check for understanding Completion of below activities AP Topic Questions
Formative Assessment (drives instructional decisions)	Teacher Observation Check for Understanding AP Topic Questions
Activities and Resources	• Watch the lesson video and take the corresponding quiz. Thi quiz is a quick check for understanding. • Explore the *Make a Rectangle* example. • Explore the *Nested Loop Iteration Counts* example. • Explore the *Inverted Triangle* example. • Complete the *Upright Number Triangle* exercise. • Complete the *Make a Tree exercise*. ○ This is a culminating activity that can be used as an individual or small group project/lab. • Complete the *Multiplication Table* exercise.
Suggested Modifications	See Lesson 4.1 above.

LESSON 4.5

Lesson 4.5: Informal Code Analysis

In this lesson, students will examine the concept of informal code analysis. This includes an algorithm's correctness, efficiency and the ability to be understood. This lesson corresponds with AP Computer Science A topic 4.5.

Student Learning Intentions (SLI) WALT: (We are learning to…)	• Compute statement execution counts of iterative statements • Compute informal run-time comparison of iterative statement

Student Learning Strategies	Code tracing Create a plan Error analysis Identify a subtask Look for a pattern Pair programming Predict and compare Simplify the problem Think aloud
Success Criteria	Check for understanding Completion of below activities AP Topic Questions
Formative Assessment (drives instructional decisions)	Teacher Observation Check for Understanding AP Topic Questions
Activities and Resources	• Watch the lesson video and take the corresponding quiz. Thi quiz is a quick check for understanding. • Explore the *Loop Execution Count* example. • Explore the *While Loop Time* example. • Explore the *For Loop Time* example. • Complete the *Time Comparisons* free response activity. • Complete the *Improving isChar Speed* exercise. • Explore the *Improving isChar Speed Check* example. • Complete the *isChar Speed Reflection* free response activity
Suggested Modifications	See Lesson 4.1 above.

INTERDISCIPLINARY CONNECTIONS: NEW JERSEY STUDENT LEARNING STANDARDS FOR ELA, SOCIAL STUDIES, SCIENCE AND/OR MATHEMATICS

CCSS - English-Language Arts

Key Ideas and Details:

CCSS.ELA-LITERACY.RL.11-12.1 Cite strong and thorough textual evidence 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.

Integration of Knowledge and Ideas:

CCSS.ELA-LITERACY.W.11-12.1 Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.

Production and Distribution of Writing:

CCSS.ELA-LITERACY.W.11-12.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

Research to Build and Present Knowledge:

CCSS.ELA-LITERACY.W.11-12.7 Conduct short as well as more sustained research projects to answer a question (including a

self-generated question) or solve a problem; narrow or broaden the inquiry when appropriate; synthesize multiple sources on the subject, demonstrating understanding of the subject under investigation.
Range of Writing:
CCSS.ELA-LITERACY.W.11-12.10 Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of tasks, purposes, and audiences

CCSS - Mathematics
Reason quantitatively and use units to solve problems:
CCSS.MATH.CONTENT.HSN-Q.A.2 Define appropriate quantities for the purpose of descriptive modeling.
Create equations that describe numbers or relationships:
CCSS.MATH.CONTENT.HSA-CED.A.1 Create equations and inequalities in one variable and use them to solve problems.
Analyze functions using different representations:
CCSS.MATH.CONTENT.HSF-IF.C.7 Graph functions expressed symbolically and show key features of the graph.
Apply geometric concepts in modeling situations:
CCSS.MATH.CONTENT.HSG-MG.A.1 Use geometric shapes, their measures, and their properties to describe objects
Calculate expected values and use them to solve problems:
CCSS.MATH.CONTENT.HSS-MD.A.1 Define a random variable for a quantity of interest by assigning a numerical value to each event in a sample space;
CCSS.MATH.CONTENT.HSS-MD.A.2 Calculate the expected value of a random variable;

English Language Arts

- Journal writing
- Close reading of industry-related content
- Create a brochure for a specific industry
- Keep a running word wall of industry vocabulary

Social Studies

- Research the history of a given industry/profession
- Research prominent historical individuals in a given industry/profession
- Use historical references to solve problems

World Language

- Translate industry-content
- Create a translated index of industry vocabulary
- Generate a translated list of words and phrases related to information technology

Math

- Compare and contrast use of equations and variables in algebra and programming.
- Program graphics and use the properties of geometric shapes
- Compare the computer graphic coordinate system with the Cartesian coordinate plane in math
- Compare probability and the use of random numbers in computer programming.
- Track and track various data, such as industry's impact on the GDP, career opportunities or among of individuals currently occupying careers

Fine & Performing Arts

- Create a poster recruiting young people to focus their studies on a career in Information Technology

Science

- Research the environmental impact of a given career or industry
- Research latest developments in Information technology
- Investigate applicable-careers in STEM fields

REFLECTIONS