

Unit 3: Boolean Expressions and if Statements

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

UNIT RATIONALE

Algorithms are composed of three building blocks: sequencing, selection, and iteration. This unit focuses on selection, which is represented in a program by using conditional statements. Conditional statements give the program the ability to decide and respond appropriately and are a critical aspect of any nontrivial computer program. In addition to learning the syntax and proper use of conditional statements, students will build on the introduction of Boolean variables by writing Boolean expressions with relational and logical operators. The third building block of all algorithms is iteration, which you will cover in Unit 4. Selection and iteration work together to solve problems.

ESSENTIAL QUESTIONS

How can you use different conditional statements to write a pick-your-own-path interactive story?
Why is selection a necessary part of programming languages?

STANDARDS

NEW JERSEY STUDENT LEARNING STANDARDS: CONTENT AREA

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

Answer the following statements with a True or a False:

- 5 is greater than 4
- 10 is greater than or equal to 11
- 12 is less than or equal to 12
- "hello" is equal to "hello"
- 3.5 is not equal to 3

Write the above statements in code using the comparison operators `<`, `>`, `=`, `≥`, `≤`, and `≠`.

Discussion question:

- Why might it be useful for us to be able to compare things in our programs?

INSTRUCTIONAL PLAN

MODULE 3

LESSON 3.5

Lesson 3.5: Compound Boolean Expressions

In this lesson, students will determine the truth value of compound Boolean expressions. They will also examine short circuit evaluation and nested if statements. This lesson corresponds with AP Computer Science A topic 3.5.

Student Learning Intentions (SLI) WALT: (We are learning to...)	Evaluate compound Boolean expressions in program code
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	<ul style="list-style-type: none"> • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding. • Explore the <i>Light Switch</i> example. • Explore the <i>Number in Range</i> example. • Explore the <i>Pizza Slices</i> example. • Complete the <i>Roller Coaster</i> exercise. • Complete the <i>*Compound Roller Coaster*</i> exercise. • Complete the <i>Divisibility</i> exercise. • Complete the <i>Find the Minimum</i> exercise.
Suggested Modifications	See Lesson 3.1 above.

LESSON 3.4

Lesson 3.4: else if Statements

In this lesson, students will take control structures a step further and learn how to implement a statement with an else if condition.

This lesson corresponds with AP Computer Science A topic 3.4.

Student Learning Intentions (SLI) WALT: (We are learning to...)	Use an else if statement
Student Learning Strategies	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Check for understanding Completion of below activities AP Topic Questions
Formative Assessment (drives instructional decisions)	<ul style="list-style-type: none"> Teacher Observation Check for Understanding AP Topic Questions
Activities and Resources	<ul style="list-style-type: none"> • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding. • Explore the <i>Add Tip</i> example. • Explore the <i>Add Tip - 8 or 4 Customers</i> example.

	<ul style="list-style-type: none"> • Explore the <i>Add Tip - 8, 4, 2 Customers</i> example. • Complete the <i>Positive, Negative, or Zero</i> exercise. • Complete the <i>Salmon Spawn</i> exercise. • Complete the <i>Berries</i> exercise. • Complete the <i>Battleships</i> exercise.
Suggested Modifications	See Lesson 3.1 above.

LESSON 3.3

Lesson 3.3: if-else Statements

In this lesson, students will expand their knowledge of control structures and learn about if/else statements. This lesson corresponds with AP Computer Science A topic 3.3.

Student Learning Intentions (SLI) WALT: (We are learning to...)	Use if/else statements
Student Learning Strategies	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Check for understanding Completion of below activities AP Topic Questions
Formative Assessment (drives instructional decisions)	<ul style="list-style-type: none"> Teacher Observation Check for Understanding AP Topic Questions
Activities and Resources	<ul style="list-style-type: none"> • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding. • Explore the <i>Bill with add tip</i> example. • Explore the <i>Even and Odd</i> example. • Complete the <i>Positive or Negative</i> exercise. • Complete the <i>Battleships</i> exercise. • Complete the <i>Ratings</i> exercise.

	<ul style="list-style-type: none"> • Complete the <i>Player Score</i> exercise.
Suggested Modifications	See Lesson 3.1 above.

LESSON 3.2

Lesson 3.2: if Statements and Control Flow

In this lesson, students will learn about control structures and if statements. If statements are used in programming when there are a set of statements that should be executed only when certain conditions are met. This lesson corresponds with AP Computer Science A topic 3.2.

Student Learning Intentions (SLI) WALT: (We are learning to...)	<ul style="list-style-type: none"> • Represent branching logical processes by using flowcharts • Use if statements
Student Learning Strategies	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Check for understanding Completion of below activities AP Topic Questions
Formative Assessment (drives instructional decisions)	<ul style="list-style-type: none"> Teacher Observation Check for Understanding AP Topic Questions
Activities and Resources	<ul style="list-style-type: none"> • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding. • Explore the <i>Can Vote</i> example. • Explore the <i>Negative Numbers</i> example. • Explore the <i>Rectangle</i> example. • Complete the <i>Discounts</i> exercise. • Complete the <i>Sweet or Unsweet?</i> exercise. • Complete the <i>Cooking</i> exercise. • Complete the <i>Rating</i> exercise.

Suggested Modifications

See Lesson 3.1 above.

LESSON 3.1**Lesson 3.1: Boolean Expressions and if Statements**

In this lesson, students will learn about Boolean expressions and relational operators. Relational Operators are constructs in programming that allow for the comparison of the values of two expressions. This lesson corresponds with AP Computer Science A topic 3.1.

Student Learning Intentions (SLI) WALT: (We are learning to...)	Evaluate Boolean expressions that use relational operators in program code
Student Learning Strategies	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Check for understanding Completion of below activities AP Topic Questions
Formative Assessment (drives instructional decisions)	<ul style="list-style-type: none"> Teacher Observation Check for Understanding AP Topic Questions
Activities and Resources	<ul style="list-style-type: none"> • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding. • Explore the <i>Old Enough to Vote</i> example. • Explore the <i>Grade Range</i> example. • Explore the <i>Equality of Strings</i> example. • Complete the <i>Number Order</i> exercise. • Complete the <i>Sugar Tax</i> exercise. • Complete the <i>Triple Double</i> 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 makes 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 an 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 to 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 in 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 3.6

Lesson 3.6: Equivalent Boolean Expressions

In this lesson, students will examine De Morgan's laws and determine when compound Boolean expressions are equivalent. They will use truth tables to display equivalent Boolean expressions. This lesson corresponds with AP Computer Science A topic 3.6.

Student Learning Intentions (SLI) WALT: (We are learning to...)	<ul style="list-style-type: none">• Compare and contrast equivalent Boolean expressions• Apply De Morgan's Laws to Boolean expressions
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	<ul style="list-style-type: none"> • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding. • Explore the <i>De Morgan AND</i> example. • Explore the <i>De Morgan OR</i> example. • Complete the <i>Amusement Park</i> exercise. • Complete the <i>Odd Numbers</i> exercise. • Complete the <i>Odd and Even</i> exercise.
Suggested Modifications	See Lesson 3.1 above.

LESSON 3.7

Lesson 3.7: Comparing Objects

In this lesson, students will learn the different ways of comparing objects and how it is different than comparing primitive types. With objects, because the variable is pointing to a location, the use of `==` is not possible, as the memory address is being compared between objects, not the actual values. Students will discover and use the `.equals` method. This lesson corresponds with AP Computer Science A topic 3.7.

Student Learning Intentions (SLI) WALT: (We are learning to...)	Compare object references using Boolean expressions in program code
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

<p>Activities and Resources</p>	<ul style="list-style-type: none"> • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding. • Explore the <i>Comparing Strings</i> example. • Explore the <i>Comparing Rectangles</i> example. • Explore the <i>Null Test</i> example. • Explore the <i>Identify Aliases</i> example. • Complete the <i>String Variable Trace</i> exercise. • Complete the <i>String Trace</i> free response activity. • Complete the <i>Three Strings</i> exercise. • Complete the <i>Comparing Circles</i> exercise. • Complete the <i>Comparing Objects</i> handout.
<p>Suggested Modifications</p>	<p>See Lesson 3.1 above.</p>

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
