

# Unit 1: Primitive Types

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

## UNIT RATIONALE

---

This unit introduces students to the Java programming language and the use of classes, providing students with a firm foundation of concepts that will be leveraged and built upon in all future units. Students will focus on writing the main method and will start to call preexisting methods to produce output. Students will start to learn about three built-in data types and learn how to create variables, store values, and interact with those variables using basic operations. The ability to write expressions is essential to representing the variability of the real world in a program and will be used in all future units. Primitive data is one of two categories of variables covered in this course. The other category, reference data, will be covered in Unit 2.

## ESSENTIAL QUESTIONS

---

How can we use programs to solve problems?

In what ways are numbers used in the programs and apps you use most often?

How are mathematical concepts being used in the programs and apps that you use most often?

## STANDARDS

---

### NEW JERSEY STUDENT LEARNING STANDARDS: CONTENT AREA

---

#### New Jersey (NJSL) - 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**

---

CodeHS Java Pretest

## **INSTRUCTIONAL PLAN**

---

# MODULE 1

## LESSON 1.4

### Lesson 1.4: Compound Assignment Operators

In this lesson, students will learn how to use the increment operator (`++`), the decrement operator (`--`) and compound assignment operators (`+=`, `-=`, `*=`, `/=`, `%=`). The increment operator (`++`) and decrement operator (`--`) are used to add 1 or subtract 1 from the stored value of a variable. The new value is assigned to the variable. This lesson corresponds with AP Computer Science A topic 1.4.

<b>Student Learning Intentions (SLI) WALT: (We are learning to...)</b>	<ul style="list-style-type: none"><li>• Evaluate what is stored in a variable as a result of an expression with an assignment statement</li><li>• Use a compound assignment operators (<code>+=</code>, <code>-=</code>, <code>*=</code>, <code>/=</code>, <code>%=</code>), in place of the assignment operator</li><li>• Use an increment operator (<code>++</code>) or decrement operator (<code>--</code>) to add 1 or subtract 1 from the stored value of a variable</li></ul>
<b>Student Learning Strategies</b>	Code tracing Create a plan Error analysis Identify a subtask Look for a pattern Pair programming Predict and compare Simplify the problem Think aloud
<b>Success Criteria</b>	Check for understanding Completion of below activities
<b>Formative Assessment (drives instructional decisions)</b>	Teacher Observation Check for Understanding
<b>Activities and Resources</b>	<ul style="list-style-type: none"><li>• Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.</li><li>• Explore the All Functions Calculator example.</li><li>• Explore the Increase/Decrease by 1 example.</li><li>• Complete the Work Shift exercise.</li><li>• Complete the My Age exercise.</li><li>• Revisit the Computation with Variables handout from the previous lesson. Can the answers be adjusted to now use</li></ul>

	compound assignments? Have students type their answers into their Sandbox to test their results.
<b>Suggested Modifications</b>	See Lesson 1.1

## LESSON 1.3

### Lesson 1.3: Expressions and Assignment Statements

In this lesson, students learn how to use arithmetic expressions in Java. Arithmetic expressions allow the use of mathematical operations within Java. Students will also learn about the modulus operator (%) and how it is evaluated. Such expressions can be used for basic math and even more complex algorithms. This lesson corresponds with AP Computer Science A topic 1.3.

<b>Student Learning Intentions (SLI) WALT: (We are learning to...)</b>	<ul style="list-style-type: none"> <li>• evaluate arithmetic expressions in a program code</li> <li>• use the modulus operator (%)</li> </ul>
<b>Student Learning Strategies</b>	<ul style="list-style-type: none"> <li>Code tracing</li> <li>Create a plan</li> <li>Error analysis</li> <li>Identify a subtask</li> <li>Look for a pattern</li> <li>Pair programming</li> <li>Predict and compare</li> <li>Simplify the problem</li> <li>Think aloud</li> </ul>
<b>Success Criteria</b>	<ul style="list-style-type: none"> <li>Check for understanding</li> <li>Completion of below activities</li> </ul>
<b>Formative Assessment (drives instructional decisions)</b>	<ul style="list-style-type: none"> <li>Teacher Observation</li> <li>Check for Understanding</li> </ul>
<b>Activities and Resources</b>	<ul style="list-style-type: none"> <li>• Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.</li> <li>• Complete the Computation with Variables handout. After completion, discuss with students why having a computer program to evaluate arithmetic expressions is beneficial.</li> <li>• Explore the Calculator example.</li> <li>• Explore the Temperature Conversion example.</li> <li>• Explore the Tricky Java example.</li> <li>• Complete the Weight of a Pyramid exercise.</li> <li>• Complete the Add Fractions exercise.</li> <li>• Complete the Freely Falling Bodies exercise.</li> </ul>

<b>Suggested Modifications &amp; Standards</b>	See Lesson 1.1

## LESSON 1.2

### Lesson 1.2: Variables and Data Types

In this lesson, students will learn about variables. Variables allow information to be stored such as numbers, words, or true/false expressions. A variable can be thought of as a box that stores information inside. In Java, variables are composed of three things: a name, type, and value. This lesson corresponds with AP Computer Science A topic 1.2.

<b>Student Learning Intentions (SLI) WALT: (We are learning to...)</b>	<ul style="list-style-type: none"> <li>• declare, initialize and assign a value to a variable</li> <li>• identify the most appropriate data type category for a particular specification</li> <li>• categorize a data type as either primitive or reference</li> <li>• declare a final variable</li> </ul>
<b>Student Learning Strategies</b>	<ul style="list-style-type: none"> <li>Code tracing</li> <li>Create a plan</li> <li>Error analysis</li> <li>Identify a subtask</li> <li>Look for a pattern</li> <li>Pair programming</li> <li>Predict and compare</li> <li>Simplify the problem</li> <li>Think aloud</li> </ul>
<b>Success Criteria</b>	<ul style="list-style-type: none"> <li>Check for understanding</li> <li>Completion of below activities</li> </ul>
<b>Formative Assessment (drives instructional decisions)</b>	<ul style="list-style-type: none"> <li>Teacher observation</li> <li>Check for understanding</li> </ul>
<b>Activities and Resources</b>	<ul style="list-style-type: none"> <li>• Watch the first lesson video and take the corresponding quiz This quiz is a quick check for understanding.</li> <li>• Complete the Variables handout. Ensure that students understand the differences between declaring, assigning and initializing variables.</li> <li>• Explore the Variables example.</li> <li>• Explore the Using Final example.</li> <li>• Complete the Our First Integer exercise.</li> </ul>

	<ul style="list-style-type: none"> <li>• Watch the second lesson video and take the corresponding quiz. This quiz is a quick check for understanding.</li> <li>• Complete the Syntax and Grammar handout. If time allows, students can create posters or study sheets with their syntax and grammar rules.</li> <li>• Complete the Answering Questions exercise.</li> <li>• Complete the Team Ranking exercise.</li> </ul>
<b>Suggested Modifications</b>	See Lesson 1.1

## LESSON 1.1

### Lesson 1.1: Why Programming? Why Java?

In this lesson, students will be introduced to and practice using the Java main skeleton that includes the class and main method arguments. They will use the `System.out.print` and `System.out.println` commands to print string literals in the editor. This lesson corresponds with AP Computer Science A topic 1.1.

<b>Student Learning Intentions (SLI) WALT: (We are learning to...)</b>	<b>Lesson: 1.1 Why Programming? Why Java?</b> <ul style="list-style-type: none"> <li>• call system class methods to generate output to the console</li> <li>• recognize the difference between display behavior of <code>System.out.print</code> and <code>System.out.println</code></li> <li>• create string literals</li> </ul>
<b>Student Learning Strategies</b>	Error analysis Code tracing Create a plan Error analysis Identify a subtask Look for a pattern Pair programming Predict and compare Simplify the problem Think aloud
<b>Success Criteria</b>	Check for understanding Completion of below activities
<b>Formative Assessment (drives instructional decisions)</b>	Teacher Observation Check for Understanding
<b>Activities and Resources</b>	<ul style="list-style-type: none"> <li>• Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.</li> <li>• Explore the Hello World example.</li> <li>• Explore the Printing Multiple Lines example.</li> <li>• Complete the Welcome Program exercise.</li> </ul>

- Complete the ASCII Art exercise.
- Complete the Fixing a Paragraph exercise.
- Complete the Making Popcorn exercise.
- Complete the Personal Timeline 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 easy for students to understand and remember. Chunking is based on the presumption that our working memory is easily overloaded with 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 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. Classroom 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 instruction 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 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 1.5

### Lesson 1.5: User Input

In this lesson, students will learn how to receive user input with the Scanner class in Java. After creating a new Scanner object, they will learn the commands below that will allow them to take in data from the user.

```
String name = input.nextLine();
int number = input.nextInt();
double decimal = input.nextDouble();
```

**Student Learning Intentions (SLI) WALT: (We are learning to...)**

- import and initialize a new Scanner to take in user input
- create variables that take the assigned value of the user input.
- choose the correct command that will allow the program to receive the input value that corresponds with its desired data type

**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
<b>Formative Assessment (drives instructional decisions)</b>	Teacher Observation Check for Understanding
<b>Activities and Resources</b>	<ul style="list-style-type: none"> <li>• Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.</li> <li>• Explore the int Before String example.</li> <li>• Complete the My Age (User Input) exercise.</li> <li>• Complete the Night Out exercise.</li> <li>• Complete the MLA Citation exercise.</li> </ul>
<b>Suggested Modifications</b>	See Lesson 1.1

## LESSON 1.6

### Lesson 1.6: Casting and Ranges of Variables

In this lesson, students will learn how to change the data type of a value through the process of casting. They will also use casting to help round double values to the closest integer value. Students will explore the range of variables allowed in Java and understand integer overflow when a value outside of this range is attempted. This lesson corresponds with AP Computer Science A topic 1.5.

<b>Student Learning Intentions (SLI) WALT: (We are learning to...)</b>	<ul style="list-style-type: none"> <li>• Evaluate arithmetic expressions that use casting</li> <li>• Understand that integer values in Java are stored using a finite amount (4 bytes) of memory. Therefore, an int value must be in the range from Integer.MIN_VALUE to Integer.MAX_VALUE inclusive</li> <li>• Understand when an expression evaluates to an int value outside of the allowed range, an integer overflow occurs</li> <li>• Recognize and use implicit casting</li> <li>• Use casting to round to the nearest integer</li> </ul>
<b>Student Learning Strategies</b>	Code tracing Create a plan Error analysis Identify a subtask Look for a pattern Pair programming Predict and compare Simplify the problem

	Think aloud
<b>Success Criteria</b>	Check for understanding Completion of below activities
<b>Formative Assessment (drives instructional decisions)</b>	Teacher Observation Check for Understanding
<b>Activities and Resources</b>	<ul style="list-style-type: none"> <li>• Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.</li> <li>• Complete the Casting to an Int exercise.</li> <li>• Complete the Casting to a Double exercise.</li> <li>• Explore the Casting Order of Operations example.</li> <li>• Explore the Rounding Using Casting example.</li> <li>• Complete the Movie Ratings exercise.</li> <li>• Explore the Implicit Casting example.</li> <li>• Explore the Min and Max Values of Integers example.</li> <li>• Complete the Integer Overflow exercise.</li> <li>• Complete the Casting Match handout. This activity is a good recap and review of the new concepts in this lesson. Consider pairing up students or forming small groups.</li> </ul>
<b>Suggested Modifications</b>	See Lesson 1.1

## **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**

---