

Quorum, Visual Track Sample Lab Adjustment Recommendations

- Introduction/Orientation (pgs. 2-3)
- A.1 Introduction to Types and Variables (pgs. 4-9)
- A.2 Operators (pgs. 10-14)
- A.3 Math, Output and User Input (pgs. 15-20)
- A.4 Getting Started with Games (pgs. 21-25)
- A.5 Displaying an Image (pgs. 26-30)
- A.6 Getting Started with the Offline IDE (pgs. 31-32)
- A.7 Type Casting (pgs. 33-36)
- A.8 Ramp Experiment (pgs. 37-41)
- A.9 Commenting Code (pg. 42)
- A.10 Tortoise vs. the Hare (pgs. 43-44)
- A.11 Atmospheric Calculations (pgs. 45-49)
- A.12 Quiz (pg. 50)

Guide Overview: This resource guide **provides teachers** with research-derived recommendations they can use in their CSP class to improve accessibility of the lesson for students with learning and attention disorders. The recommendations stem from what is known about the range of learning and attention disorders typical in classrooms (e.g., reading, written expression, language, math, attention) and the types of adjustments commonly made to support students who learn differently because of these disorders in any discipline.

Adjustment Terminology: This guide includes two types of adjustment recommendations: **adaptations** and **accommodations**. The term “**adaptation**” refers to instructional practices that can be used whole-class (to benefit students with learning and attention disorders, and potentially all learners in the classroom). The term “**accommodation**” refers to differentiation for individual students based on their unique learning needs beyond what the whole-class practices provide. Many of the “adaptations” are akin to [Universal Design for Learning \(UDL\) strategies](#) that emphasize how information is presented, how students demonstrate understanding, and how students interact and engage with materials. However, these recommendations were specifically designed to address the needs of students with diagnosed learning and attention disorders.

Learning and Attention Disorders: The guide identifies the learning and attention disorders that each recommendation addresses. The list of broad disorders and sub-disorders listed on page 3 in **Table 1** are representative of the range of disorders in the general population as identified by the research team learning specialists. The adaptations and accommodations commonly made for these learning differences in any discipline inform the recommendations below.

Learn More about Instruction for Students with Learning & Attention Disorders

Find ways to collaborate with your school’s Special Education personnel (Department Heads, paraprofessionals, school psychologists, etc.) to generally benefit from their expertise and perspective. These specialists may also be able to help you navigate any unfamiliar or unclear information contained in a student’s [Individualized Education Program \(IEP\)](#) or a [504 plan](#) (legal documents that outline supports that schools and teachers must provide for students with disorders that impact their learning, beyond the recommendations here).

Please acknowledge the Outlier Research & Evaluation and Wolcott School team in any use of this resource.

Suggested Citation: The Outlier Research & Evaluation and Wolcott School CSP Team (2018). Quorum Visual Track Lesson Adjustments for Students with Learning and Attention Disorders. Chicago, IL; Outlier Research & Evaluation at UChicago STEM Education, University of Chicago. This project was supported by the National Science Foundation, Grant # 1542963.

Table 1. **Broad Learning and Attention Disorders**

Specific Learning Disabilities and Attention Deficit Disorders	Sub-Disorders
Reading Disorders – All <i>(Adjustment applies to all types of reading disorders)</i>	<ul style="list-style-type: none"> • Reading decoding • Reading fluency • Reading comprehension
Written Expression Disorders – All <i>(Adjustment applies to all types of written expression disorders)</i>	<ul style="list-style-type: none"> • Spelling accuracy • Grammar and punctuation accuracy • Clarity or organization of written expression
Math Disorders – All <i>(Adjustment applies to all types of math disorders)</i>	<ul style="list-style-type: none"> • Number sense • Memorization of arithmetic facts • Accurate or fluent calculation • Accurate math reasoning
Attention Disorders - All <i>(Adjustment applies to all types of attention disorders)</i>	<ul style="list-style-type: none"> • Combined presentation (inattentive & hyperactive/impulsive) • Predominantly inattentive • Predominantly hyperactive/impulsive
Language Disorders – All <i>(Adjustment applies to all types of language disorders)</i>	<ul style="list-style-type: none"> • Reduced vocabulary • Limited sentence structure • Impairments in discourse • Social pragmatic communication

Underlying Processes: In our work, we also consider not only the broader diagnostic information in Table 1, but also the basic, [psychological processes underlying the disorders](#) that can make certain activities common in CS and non-CS classes challenging. This is because two students may both have a disorder in reading that, due to the different underlying processes that contribute to the disorder, call for different kinds of actions to address the disorder. Therefore, it is important to remember that, like students as a whole, those with diagnosed disorders are also very heterogeneous. The recommendations for teachers in this document identifies the **broader disorders** (Table 1, first column).

Quorum A.1 Types and Variables

Note: Throughout most Visual Track lessons/activities, adjustments are written based on on-line use of the Quorum language, as download for offline use was noted as optional. Teachers may not know that instructions exist for installing Quorum offline until mid-way through the Visual Track lessons. Our assumption is that until teachers are more comfortable and familiar with the materials, they will use the online compiler and output window (embedded in each lesson webpage for use) for most lesson activities.

Adjustment Suggestions

Suggestion 1

Lesson section: Student activity instructions (*throughout the lesson*)

Task description: Student read information and instructions on the web page with no verbal input from the teacher. Across lessons, students must carefully read to find markers that indicate where they are expected to perform an activity, and where to find the necessary information to do so (eg., in some lessons, the output windows includes some instruction, while others tell students to look outside the output window for what action to perform).

Some students may have difficulty:

- reading dense and/or complex text on webpages (Reading, Attention)
- recalling or retrieving terminology and concepts presented elsewhere to apply to current work (Language, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- planning and determining an approach towards the independent work time; using a systematic approach while also self-monitoring (Attention/Executive function)

Teacher adjustment recommendations - adaptations:

- At the start of each lesson/set of activities for the day, take time to explicitly review and summarize the main ideas, student activities, and goals for the day verbally, and with the information displayed on a presentation slide or white board.

Teacher adjustment recommendations - accommodations:

- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Offer students the use of text-to-speech software to read and reread information and instructions (as some sections are dense with text).
- Provide explicit guidance for students who may have difficulty identifying the activity instructions, or sequence of steps when interacting with the input window throughout the lesson.

Suggestion 2

Lesson section: Vocabulary (*used throughout lesson*)

Task description: Students encounter and are expected to use new terms while performing programming activities (e.g., variable, integer, Boolean).

Some students may have difficulty:

- reading and understanding new vocabulary (Language)
- comprehending new terms introduced with little information about their meaning (Language)

Teacher adjustment recommendations - adaptations:

- Place key terms and definitions on a presentation slide or white board so that information is presented visually, and remains accessible while students work. As you do, provide students with concrete examples of the new vocabulary words. Show examples of the concepts in the lesson.
- Print out the “Quick Reference: Quorum Overview” sheet for students to reference vocabulary definitions (e.g., integer, Boolean, text). Alternately, create a more comprehensive reference sheet (preferably a shared electronic document), that you, or your students can add to as new operations are encountered and can include additional examples along with the terms.
- Place new vocabulary terms in a running class glossary and assist students in locating them. Alternately, have students bookmark other online glossaries with grade-appropriate definitions, and/or print them out for use (e.g., two examples can be found [here](#) and [here](#)).
- Verbally review vocabulary terms with the whole class at the start of lessons.

Teacher adjustment recommendations - accommodations: None

Suggestion 3

Lesson section: Integer section

Task description: Students begin the first activity/tutorial/lesson in Quorum’s Visual Track, and are instructed to “try the above code blocks” in the Quorum programming environment. There are no directions to indicate how, or why students should use the “Save,” “Load,” “Embed,” “Build,” or “Run Program for Activity” buttons, or what will happen when they do (the output window opens in a separate tab; students must return to the original tab to continue the activity).

Some students may have difficulty:

- initiating work in the programming environment, with little instruction about how the environment works, where to look for the output, when to copy/paste and when to generate their own unique code, etc.
- reading dense and/or complex text on webpages (Reading, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)

Teacher adjustment recommendations - adaptations:

- As a class, orient students to use of the Quorum programming language input and output windows (explain the difference between the “Console” and “Visual” output windows), buttons and functions, and saving work to the “My Projects” tab.
- Create a handout for students to reference that shows the sequence of steps to follow when they encounter each set of red text and the program input window throughout the lessons/activities (e.g., enter code in the input window, test the code for errors by clicking “Build,” check for error messages in the gray box below the input window, click “Run Program” and see your output in the Console window.

Teacher adjustment recommendations - accommodations:

- Some students may need additional, direct support determining the correct sequence of steps to take to complete this activity the first time, and identifying the instructions throughout the lesson.

Suggestion 4

Lesson section: Text section

Task description: Students are introduced to “text” and instructed to copy and paste a text variable into the input window.

Some students may have difficulty:

- comprehending the use of terms in a new/different context (e.g., coding) (Language)
- reading and comprehending directions with little information about why they are performing an activity (Reading, Attention)
- identifying syntax errors (e.g., the difference between “Hello world!” and “Hello world!”) (Reading, Attention)

Teacher adjustment recommendations - adaptations:

- Draw attention to the use of the closed quotation marks, and discuss why they are needed for text variables. Show students what happens when the quotation marks are not included.

Teacher adjustment recommendations - accommodations: None

Suggestion 5

Lesson section: Type Conversion section

Task description: Students are introduced to “type conversion/type casting” and instructed to copy and paste example code into the input window.

Some students may have difficulty:

- recalling or retrieving terminology and procedures (e.g., type casting, errors) presented elsewhere to apply to current work (Language, Attention)
- understanding the process of type casting (or converting types) and how the commands used in the example convert the values (Language, Reading, Attention) [**note**, there is a chunk of text missing above the “Activity: Try Type Conversion” section, which currently reads “We can learn more about errors on theIn these cases, our program alert us to the fact that there is an error on the system.”]

Teacher adjustment recommendations - adaptations:

- Instruct students to keep open the “[Tutorial: Type Casting](#)” page for reference (or, have them print out a hard copy to reference).
- Instruct students to keep open the “[Tutorial: Handling Errors](#)” page for reference while working (or, have them print out a hard copy to reference).
- Project key take aways from the type casting and errors pages on a presentation slide or white board so it’s accessible as students work.
- Explain in detail how the commands convert the value types; walk students through the process and the rationale; and check for understanding.

Teacher adjustment recommendations - accommodations: None

Suggestion 6

Lesson section: Variable names section

Task description: Students read about variable names, view examples of allowable and unallowable naming conventions, and try some out in the input window.

Some students may have difficulty:

- reading dense and/or complex text on webpages (Reading, Attention)
- identifying the key take-aways from the examples (Attention)
- scanning, reading, and reviewing work and text-based code (Reading, Attention)
- thinking flexibly and systematically about command use (Cognitive flexibility: Attention, Written Expression, Math)

Teacher adjustment recommendations - adaptations:

- Point out the difference between the code example and the commentary (e.g., “integer 5sally = 5” isn’t clearly separated from “//a number cannot be the first character” with a different color, nor has the use of “/” to separate notes from codes been introduced, which may cause some student confusion).
- Demonstrate for the class what type of error message will occur when a naming convention is not allowable.
- Create a reference sheet (electronic and hard copy) that lists different types of variables (and descriptions), as well as naming conventions. Encourage students to keep this reference handy while working across all lessons.
- Instruct students to create an electronic document (Word doc, Google doc) to record the variables introduced; students can refer to this document throughout the lessons to: a) compare variables side-by-side to identify errors when debugging, and b) copy the variables in the document to paste in the program in order to ensure they always write them in the same manner.

Teacher adjustment recommendations - accommodations:

- For students who need additional support, provide each unallowable example on a white board or presentation slide, separately, so students can focus on each one separately.

Quorum A.2 Operators

Adjustment Suggestions

Suggestion 1

Lesson section: Student programming instructions (*throughout the lesson*)

Task description: Student read information and instructions on the web page with no verbal input from the teacher. Across lessons, students must carefully read to find markers that indicate where they are expected to perform an activity, and where to find the necessary information to do so (eg., in some lessons, the output windows include some instruction, while others tell students to look outside the output window for what action to perform).

Some students may have difficulty:

- reading dense and/or complex text on webpages (Reading, Attention)
- recalling or retrieving terminology and concepts presented elsewhere to apply to current work (Language, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- planning and determining an approach towards the independent work time; using a systematic approach while also self-monitoring (Attention/Executive function)

Teacher adjustment recommendations - adaptations:

- At the start of each lesson/set of activities for the day, take time to explicitly review and summarize the main ideas, student activities, and goals for the day verbally, and with the information displayed on a presentation slide or white board.

Teacher adjustment recommendations - accommodations:

- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Offer students the use of text-to-speech software to read and reread information and instructions (as some sections are dense with text).
- Provide explicit guidance for students who may have difficulty identifying the activity instructions, or sequence of steps when interacting with the input window throughout the lesson.

Suggestion 2

Lesson section: Vocabulary (*used throughout lesson*)

Task description: Students encounter and are expected to use new terms while performing programming activities (e.g., modulus, concatenate).

Some students may have difficulty:

- reading and understanding new vocabulary (Language)
- comprehending new terms introduced with little information about their meaning (Language)

Teacher adjustment recommendations - adaptations:

- Place key terms and definitions on a presentation slide or white board so that information is presented visually, and remains accessible while students work. As you do, provide students with concrete examples of the new vocabulary words. Show examples of the concepts in the lesson.
- Print out the “Quick Reference: Quorum Overview” sheet for students to reference vocabulary definitions (e.g., integer, Boolean, text). Alternately, create a more comprehensive reference sheet (preferably a shared electronic document), that you, or your students can add to as new operations are encountered and can include additional examples along with the terms.
- Place new vocabulary terms in a running class glossary and assist students in locating them. Alternately, have students bookmark other online glossaries with grade-appropriate definitions, and/or print them out for use (e.g., two examples can be found [here](#) and [here](#)).
- Verbally review vocabulary terms with the whole class at the start of lessons.

Teacher adjustment recommendations - accommodations: None

Suggestion 3

Lesson section: Integer and number types section

Task description: Students read about terms presented in a previous lesson (*integer* and *number types*) and try out mathematical operations using several different operators in the input window.

Some students may have difficulty:

- identifying and using number patterns and mathematical symbols, such as greater than and less than symbols (Math)
- performing mental math calculations (subtracting, adding, multiplying, dividing) or recalling math facts in order to comprehend examples and identify whether programs run accurately (Math)

Teacher adjustment recommendations - adaptations:

- Print out the “Quick Reference: Quorum Overview” for students to reference vocabulary definitions and examples (e.g., integer, number). Alternately, create a more comprehensive reference sheet, that you, or your students can add to as new operations are encountered.
- Project key terms and operators on a presentation slide or white board so they are accessible as students work.

Teacher adjustment recommendations - accommodations:

- Circulate and assist students having difficulty with terminology or distinguishing information to process vs. instructions/to-dos throughout the activity.
- Support students having difficulty visually distinguishing between greater than/lesser than/equal to operators, using them correctly, and accurately evaluating what the output should be (true or false).
- Encourage students to use calculators to perform mental computations or to assist them in recalling “math facts.”

Suggestion 4

Lesson section: Boolean types section

Task description: Students read about Boolean types and then try out code using *equals* and *not equals*.

Some students may have difficulty:

- identifying and using mathematical symbols within Boolean types (Math)
recalling, understanding, and applying Boolean concepts for the different types (Math, Reading)
- understanding, reading and using the Boolean principles that involve patterns (left, right, and, or,) (Reading, Math, Attention)

Teacher adjustment recommendations - adaptations:

- Project the operators on a white board or presentation slide with examples and definitions, so students can reference them while working.

Teacher adjustment recommendations - accommodations:

- If needed, work through additional examples that illustrate variables that are Boolean (only two possible values) and those that are not, with and without the use of “and” and “or.”

Suggestion 5

Lesson section: Text types section

Task description: Students read about text types and try out code that combines a text value with an integer.

Some students may have difficulty:

- writing code and identifying coding errors related to spelling, capitalizations, and punctuation (e.g., the difference between “Hello,” vs “Hello” and “world!” vs “world”) (Reading, Attention)

Teacher adjustment recommendations - adaptations:

- Draw attention to the use of the closed quotation marks, and discuss why they are needed for text variables. Show students what happens when the quotation marks are not included.

Teacher adjustment recommendations - accommodations: None

- Circulate and assist students who may have difficulty consistently spelling words correctly each time when writing code or identifying errors (e.g., “true” and “false”). Suggest these students create an electronic document (Word, Google) to record often-used terms (“true,” “false,” “text,” “output”). Students can refer to the document to copy and paste the words from the document into the program in order to ensure they always spell them correctly.

Quorum A.3. Math Output and User Input

Adjustment Suggestions

Suggestion 1

Lesson section: Student programming instructions (*throughout the lesson*)

Task description: Student read information and instructions on the web page with no verbal input from the teacher. Across lessons, students must carefully read to find markers that indicate where they are expected to perform an activity, and where to find the necessary information to do so (eg., in some lessons, the output windows includes some instruction, while others tell students to look outside the output window for what action to perform).

Some students may have difficulty:

- reading dense and/or complex text on webpages (Reading, Attention)
- recalling or retrieving terminology and concepts presented elsewhere to apply to current work (Language, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- planning and determining an approach towards the independent work time; using a systematic approach while also self-monitoring (Attention/Executive function)

Teacher adjustment recommendations - adaptations:

- At the start of each lesson/set of activities for the day, take time to explicitly review and summarize the main ideas, student activities, and goals for the day verbally, and with the information displayed on a presentation slide or white board.

Teacher adjustment recommendations - accommodations:

- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Offer students the use of text-to-speech software to read and reread information and instructions (as some sections are dense with text).
- Provide explicit guidance for students who may have difficulty identifying the activity instructions, or sequence of steps when interacting with the input window throughout the lesson.

Suggestion 2

Lesson section: Vocabulary (*used throughout lesson*)

Task description: Students learn and use new terms while performing programming activities.

Some students may have difficulty:

- reading and understanding new vocabulary (Language)
- comprehending new terms introduced with little information about their meaning (Language)

Teacher adjustment recommendations - adaptations:

- Place key terms and definitions on a presentation slide or white board so that information is presented visually, and remains accessible while students work. As you do, provide students with concrete examples of the new vocabulary words. Show examples of the concepts in the lesson.
- Place new vocabulary terms in a running class glossary and assist students in locating them.
- Verbally review vocabulary terms with the whole class at the start of lessons.
- Refer students to the descriptions and examples in the earlier lesson “Tutorial: Types and Variables.”

Teacher adjustment recommendations - accommodations: None.

Suggestion 3

Lesson section: Goal 1, Using basic mathematical operations and solving problem statements in Quorum

Task description: Students read instructions to “declare and initialize” integer and number variables in the input window.

Some students may have difficulty:

- recalling or retrieving terminology and procedures (e.g., integer, variable, number) presented previously, but used throughout the lesson (Language, Attention)
- comprehending the use of terms in a new/different context (e.g., coding) (Language)

Teacher adjustment recommendations - adaptations:

- Print out the “Quick Reference: Quorum Overview” for students to reference vocabulary definitions and examples (e.g., integer, number). Alternately, create a more comprehensive reference sheet, that you, or your students can add to as new operations are encountered.

Teacher adjustment recommendations - accommodations:

- For students who need additional support, provide instructions as a sequence of steps to clearly indicate the order in which students should enter variables.

Suggestion 4

Lesson section: Goal 1, Using basic mathematical operations and solving problem statements in Quorum

Task description: Students perform different mathematical operations by entering and then changing information entered in the input window to tell Quorum to subtract, multiply, and divide.

Some students may have difficulty:

- reading and comprehending directions, particularly when they are required to read dense text on webpages (Reading, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)

Teacher adjustment recommendations - adaptations:

- Remind students what information needs to be carried forward from the first “Activity” window to each following window (e.g., paste the integer and number variables in each input window).
- Read activity questions aloud as a group and clarify vocabulary and phrasing by rephrasing information and referring students to a running glossary (that you or the students create for the class at the beginning of the school year and add to throughout the year).

Teacher adjustment recommendations - accommodations:

- For students who need additional support, provide the sequence of steps required for each programming exercise/activity on a white board or presentation slide so students can see them more clearly, without the surrounding text.

Suggestion 5

Lesson section: Goal 1, Using basic mathematical operations and solving problem statements in Quorum

Task description: Students create and name a variable that includes addition, subtraction, multiplication, and division.

Some students may have difficulty:

- developing names for functions because the entity is difficult to describe (Language)
- thinking flexibly and systematically about command use (Cognitive flexibility: Attention, Reading, Written Expression, Math)
- retaining and applying new information to programming activities (Attention)

Teacher adjustment recommendations - adaptations:

- Remind students that for every set of operations that they do, they have to define a, b, c, and d anew.
- Refer students to the “Tutorial: Types and Variables” lesson, for reference (or, have them print out a hard copy to reference while working).

Teacher adjustment recommendations - accommodations:

- For students who seem to have difficulty naming variables, provide suggestions to help them get started.
- Some students may need additional support determining the correct sequence of steps to take to complete this activity.

Suggestion 6

Lesson section: Goal 2: Getting input from the user

Task description: Students get input from users to perform calculations using the “input keyword” (which involves using the *concatenate* function, introduced in a previous lesson). Students are not told they are using the concatenate function until the very end of the lesson.

Some students may have difficulty:

- recalling the use of each command and the type of variable that should be entered in the command (Language)
- reading webpages dense with text (Reading, Attention)

Teacher adjustment recommendations – adaptation:

- Project activity instructions on a screen and read any complex activity directions aloud as a group. Provide explanations, definitions, and examples.
- Work through additional examples that illustrate the logic in the expressions and statements, and provide clarification when necessary.
- Create a reference sheet listing new code; provide a short descriptor of the command and the type of variable that should be entered in the command. Encourage students to keep this reference handy while working.

Teacher adjustment recommendations – accommodation:

- Some students may need additional one-on-one time with the instructor to understand the expressions and how to program with them.

Quorum A.4 Getting Started with Games

Adjustment Suggestions

Suggestion 1

Lesson section: Student programming instructions (*throughout the lesson*)

Task description: Student read information and instructions on the web page with no verbal input from the teacher. Across lessons, students must carefully read to find markers that indicate where they are expected to perform an activity, and where to find the necessary information to do so (eg., in some lessons, the output windows include some instruction, while others tell students to look outside the output window for what action to perform).

Some students may have difficulty:

- reading dense and/or complex text on webpages (Reading, Attention)
- recalling or retrieving terminology and concepts presented elsewhere to apply to current work (Language, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- planning and determining an approach towards the independent work time; using a systematic approach while also self-monitoring (Attention/Executive function)

Teacher adjustment recommendations - adaptations:

- At the start of each lesson/set of activities for the day, take time to explicitly review and summarize the main ideas, student activities, and goals for the day verbally, and with the information displayed on a presentation slide or white board.

Teacher adjustment recommendations - accommodations:

- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Offer students the use of text-to-speech software to read and reread information and instructions (as some sections are dense with text).
- Provide explicit guidance for students who may have difficulty identifying the activity instructions, or sequence of steps when interacting with the input window throughout the lesson.

Suggestion 2

Lesson section: Vocabulary (*used throughout lesson*)

Task description: Students encounter new terms while reading about aspects of video game creation (e.g., frame, main loop, IDE).

Some students may have difficulty:

- reading and understanding new vocabulary (Language)
- comprehending new terms introduced with little information about their meaning (Language)

Teacher adjustment recommendations - adaptations:

- Place key terms and definitions on a presentation slide or white board so that information is presented visually, and remains accessible while students work. As you do, provide students with concrete examples of the new vocabulary words. Show examples of the concepts in the lesson.
- Print out the “Quick Reference: Quorum Overview” sheet for students to reference vocabulary definitions (e.g., integer, Boolean, text). Alternately, create a more comprehensive reference sheet (preferably a shared electronic document), that you, or your students can add to as new operations are encountered and can include additional examples along with the terms.
- Place new vocabulary terms in a running class glossary and assist students in locating them. Alternately, have students bookmark other online glossaries with grade-appropriate definitions, and/or print them out for use (e.g., two examples can be found [here](#) and [here](#)).
- Verbally review vocabulary terms with the whole class at the start of lessons.

Teacher adjustment recommendations - accommodations: None

Suggestion 3

Lesson section: Understanding Frames and Understanding the Main Loop sections

Task description: Students learn about the role of frames and main loops in video games.

Some students may have difficulty:

- reading and comprehending dense text (Reading, Attention)
- retaining and applying new information in programming contexts (Attention)
- identifying key take-aways (Attention)

Teacher adjustment recommendations - adaptations:

- As a class, watch the first 2.5 minutes of this [YouTube video](#) that describes movie frames, with the closed captioning enabled. Find further videos that highlight the role of loops in video games.
- Create a visual that summarizes key aspects related to how main loops work in games.
- As you encounter new vocabulary words, place them and their definitions on a presentation slide or white board, and offer examples.

Teacher adjustment recommendations - accommodations:

- For students who need assistance with reading the material, read it aloud to them, and check for understanding.
- Offer students the use of text-to-speech software to read and reread the information about frames and the Main Loop (as some sections are dense with text).

Suggestion 4

Lesson section: Anatomy of a Game Class and Breakdown sections

Task description: Students are instructed to create a new Quorum project in “NetBeans” development environment (which requires them to read a separate tutorial to learn how to install the offline Integrated Development Environment, or IDE). Then, students read lines of code for a game application template and descriptive notes that indicate what each part of the code does.

Some students may have difficulty:

- reading and comprehending dense text (Reading, Attention)
- retaining and applying new information in programming contexts (Attention)
- recognizing subtle differences in elements of code (when writing and debugging) related to spelling, capitalization, spacing, punctuation, and symbols (e.g., the difference between “action CreateGame” and “action Create Game” (Reading, Attention)
- returning to the lesson/tutorial after exploring hyperlinks (Attention)

Teacher adjustment recommendations - adaptations:

- Consider whether you want to have students complete this lesson/activity with the online environment/compiler and output window (embedded in the lesson website page for use) or, if you want them to download Quorum 5.0 with the offline NetBeans environment now vs. later when you complete the “Getting Started with the Offline IDE” lesson.
- If you choose to use the offline IDE to create a new Quorum project, walk students through how to install NetBeans on their computers, as this is the first time that they will be using that environment. Direct students to the appropriate download option for their operating system on the [“Quorum Download”](#) page (*Terminology note: the only difference between “Netbeans” and “Sodbeans” is that the Quorum and Accessibility plugins are loaded by default in Sodbeans*). See additional suggestions in the lesson adjustment examples for “Getting Started with the Offline IDE.”
- Project the “Anatomy of a Game Class” template code on a screen and read through the information as a group while also describing the additional information from the “Breakdown” section. Draw students’ attention to the use of punctuation, parentheses, spacing, and capitalizations that are needed for the Quorum program to run correctly.
- Create a reference sheet with the template code so that students can take notes on it as you discuss the code as a group and can reference when writing code. Encourage students to keep this reference handy in the future while working on other projects in the offline NetBeans environment.

Teacher adjustment recommendations - accommodations:

- Offer students the use of text-to-speech software to read and reread the template code description.
- For students who need text-to-voice accessibility when using the offline IDE, remind them to turn on the “Self-Voicing” option in the NetBeans environment.
- Some students may need additional one-on-one time with the instructor to understand the different actions.

- Some students may need to be redirected to the tutorial after viewing the hyperlinks within the lesson and help transitioning back to the tutorial activities.

Quorum A.5 Displaying an Image

Note: adjustment suggestions only reflect the website text/experience and not the actual offline IDE.

Adjustment Suggestions

Suggestion 1

Lesson section: Student programming instructions (*throughout the lesson*)

Task description: Student read information and instructions on the web page with no verbal input from the teacher. Across lessons, students must carefully read to find markers that indicate where they are expected to perform an activity, and where to find the necessary information to do so (eg., in some lessons, the output windows include some instruction, while others tell students to look outside the output window for what action to perform).

Some students may have difficulty:

- reading dense and/or complex text on webpages (Reading, Attention)
- recalling or retrieving terminology and concepts presented elsewhere to apply to current work (Language, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- planning and determining an approach towards the independent work time; using a systematic approach while also self-monitoring (Attention/Executive function)

Teacher adjustment recommendations - adaptations:

- At the start of each lesson/set of activities for the day, take time to explicitly review and summarize the main ideas, student activities, and goals for the day verbally, and with the information displayed on a presentation slide or white board.

Teacher adjustment recommendations - accommodations:

- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Offer students the use of text-to-speech software to read and reread information and instructions (as some sections are dense with text).
- Provide explicit guidance for students who may have difficulty identifying the activity instructions, or sequence of steps when interacting with the input window throughout the lesson.

Suggestion 2

Lesson section: Vocabulary (*used throughout lesson*)

Task description: Students encounter and are expected to use new terms while performing programming activities (e.g., instantiate, parameter) while learning how to perform programming actions (e.g., creating objects, calling actions on objects).

Some students may have difficulty:

- reading and understanding new vocabulary (Language, Reading)
- comprehending new terms introduced with no information about their meaning (e.g., instantiate, parameter) (Language)

Teacher adjustment recommendations - adaptations:

- Place key terms and their definitions on a presentation slide or white board so that information is presented visually, and remains accessible while students work. As you do, provide students with concrete examples of the new vocabulary words. Show examples of the concepts in the lesson.
- Print out the “[Quick Reference: Quorum Overview](#)” sheet for students to reference any vocabulary definitions introduced in earlier activities, and used in the current lesson. Alternately, create a more comprehensive reference sheet (preferably a shared electronic document), that you, or your students can add to as new operations are encountered and can include additional examples along with the terms.
- Place new vocabulary terms in a running class glossary and assist students in locating them. Alternately, have students bookmark other online glossaries with grade-appropriate definitions, and/or print them out for use (e.g., two examples can be found [here](#) and [here](#)).
- Verbally review vocabulary terms with the whole class at the start of lessons.

Teacher adjustment recommendations - accommodations: None

Suggestion 3

Lesson section: Overview section

Task description: Students are instructed to write a program that displays an image on the screen, completing the work in the offline Sodbeans/NetBeans environment or in the online environment (embedded in the lesson website page). To complete work in the offline environment, students are instructed to follow provided steps to download a zip file to access a project template.

(Terminology note: the only difference between “Sodbeans” and “NetBeans” is that the Quorum and Accessibility plugins are loaded by default in Sodbeans).

Some students may have difficulty:

- using a systematic approach to complete the template download steps and return to the lesson; storing, naming, and accessing files at a later date (Attention)
- differentiating the instructions for the compiler (e.g., the action, “use Libraries.Game.Game”) from the notes about what each action does (Reading, Attention)

Teacher adjustment recommendations - adaptations:

- If you choose to use the offline IDE to create a new Quorum project, walk students through how to install NetBeans on their computers as well as any other supporting program such as JDK or Java, as this is the first time that they will be using that environment, if they don’t already have it installed. Direct students to the appropriate download option for their operating system on the [“Quorum Download”](#) page (*Terminology note: the only difference between “Netbeans” and “Sodbeans” is that the Quorum and Accessibility plugins are loaded by default in Sodbeans*). See additional suggestions in the lesson adjustment examples for “Getting Started with the Offline IDE.”
- Project the “Displaying an Image” code and coding notes on a screen and read through the information as a group. Draw students’ attention to the use of punctuation, parentheses, and capitalizations that are needed for the program to run correctly.

Teacher adjustment recommendations – accommodations: none

Suggestion 4

Lesson section: Goal 1: Accessing the Quorum Standard Library

Task description: Students read instructions about how to use the Quorum Game Graphics and System File libraries, and type code.

Some students may have difficulty:

- reading dense text (Reading, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- writing commands with consistent spelling and capitalizations (Written expression, Reading)
- typing, writing, or organizing code (Written expression, Attention)

Teacher adjustment recommendations - adaptations:

- If you choose to use the online IDE, print the instructions located below the compiler window on a single sheet of paper for students to reference as they complete the activity (and to minimize the amount of scrolling up and down on the screen needed to review instructions and complete the activity).
- If students created a code reference sheet in the “Getting Started with Games” lesson, encourage them to add any new code to the reference sheet as you review the information as a group. Students can help the sheet handy while working on future projects.
- If students have difficulty with consistent spelling and capitalization, encourage them to use an electronic document where they write commands and codes and later copy and paste the commands into the programming environment rather than retyping commands and codes each time.
- Provide the sequence of steps required for the activity on a white board or presentation slide so students can see them more clearly, without the surrounding text.

Teacher adjustment recommendations - accommodations:

- Offer students the use of text-to-speech software to read and reread the instructions.
- For students who need text-to-voice accessibility when using the offline IDE, remind them to turn on the “Self-Voicing” option in the NetBeans environment.
- Some students may need additional one-on-one time with the instructor to understand the different actions, and to differentiate the activity instructions from the commentary.

Suggestion 5

Lesson section: Goal 2: Creating Objects; Goal 3: Calling Actions on Objects

Task description: Students read instructions about how to display an image, and then compile and run a program with an image.

Some students may have difficulty:

- Differentiating between different types of functions (e.g., end, Add, Load, File, number, SetPosition) (Language, Attention)
- reading dense text (Reading, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- recalling or retrieving information about programming presented previously (Attention)
- retaining and applying new information to programming activities (Attention)
- writing commands with consistent spelling and capitalizations (Written expression, Reading)
- typing, writing, or organizing code (Written expression, Attention)

Teacher adjustment recommendations - adaptations:

- Provide the sequence of steps required for the activity on a white board or presentation slide so students can see them more clearly, without the surrounding text.
- Assist students in differentiating aspects of the written text by highlighting the differences between the instructions given regarding the steps to programing and the instructions given that provide general background information.
- Provide explicit guidance for students who may have difficulty initiating and finishing the multi-step activity.
- Circulate and assist students as they write code. Draw their attention to the use of punctuation, parentheses, spacing, and capitalizations that are needed for the finished program to run correctly.

Teacher adjustment recommendations - accommodations:

- Offer students the use of text-to-speech software to read and reread the instructions.
- For students who need assistance with reading the material, read it aloud to them, and check for understanding.
- Some students may need additional one-on-one time with the instructor to understand the different actions, and to differentiate the activity instructions from the commentary.

Quorum A.6 Offline IDE

Note to teachers about Getting Started with the Offline IDE

- Adjustment suggestions only reflect the website text/experience and not the actual offline IDE.
- This activity serves as a tutorial to show students how to use the offline Integrated Development Environment (IDE). It can be completed in a shorter amount of time than a typical lesson, so consider giving students other lessons to work on.
- Terminology note: the only difference between “NetBeans” and “Sodbeans” is that the Quorum and Accessibility plugins are loaded by default in Sodbeans. Direct students to download the Sodbeans installation package, as that will reduce the number of steps that students have to complete (if students install NetBeans, they will additionally have to install the Java platform as well as the Quorum and Accessibility plugins).
- Consider completing this activity/tutorial prior to using A.4: Getting Started with Games or A.5: Displaying an Image.

Adjustment Suggestion

Suggestion 1

Lesson section: Entire tutorial/activity

Task description: Students read information and instructions about how to install the Quorum offline IDE, create a new Quorum project, and run or debug a program. Students are also introduced to debugging controls in Quorum and their keyboard shortcuts.

Some students may have difficulty:

- reading dense and/or complex text on webpages (Reading, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- comprehending and executing the instructions (Language, Reading, Attention)

Teacher adjustment recommendations - adaptations:

- Complete the installation process and running/debugging experimentation as a whole class (modeling the process on your own computer, projected for the class to see), walking students through each step. Demonstrate how the environment works before students work independently in Quorum.
- To extend the demonstration and help students make the transition from the online IDE/compiler to the offline IDE environment, copy and paste code from an earlier lesson into the offline IDE, and ask students to identify similarities and differences.
- Provide the sequence of steps required for installation on a white board or presentation slide.
- Create a reference sheet with the offline IDE controls and their corresponding keyboard shortcuts for students to use in future lessons. Demonstrate what each control does, and talk about the types of scenarios where you might use each control.
- Refer students to reference materials previous lessons they completed using the offline environment (e.g., A.4: Getting Started with Games and A.5: Displaying an Image)

Teacher adjustment recommendations - accommodations:

- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Offer students the use of text-to-speech software to read and reread information and instructions (as some sections are dense with text).
- For students who need text-to-voice accessibility in the offline IDE, ask them to turn on the “Self-Voicing” option.
- Circulate and assist students having difficulty with terminology or distinguishing information to process vs. instructions/to-dos throughout the activity.

Quorum A.7 Type Casting

Adjustment Suggestions

Suggestion 1

Lesson section: Student activity instructions (*throughout the lesson*)

Task description: Student read information and instructions on the web page with no verbal input from the teacher. Across lessons, there are no consistent markers that indicate where students should look for clear “to do” instructions (eg., in some lessons, the output windows includes some instruction, while others tell students to look outside the output window for what action to perform).

Some students may have difficulty:

- reading dense and/or complex text on webpages (Reading, Attention)
- recalling or retrieving terminology and concepts presented elsewhere to apply to current work (Language, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- planning and determining an approach towards the independent work time; using a systematic approach while also self-monitoring (Attention/Executive function)

Teacher adjustment recommendations - adaptations:

- At the start of each lesson/set of activities for the day, take time to explicitly review and summarize the main ideas, student activities, and goals for the day verbally, and with the information displayed on a presentation slide or white board.

Teacher adjustment recommendations - accommodations:

- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Offer students the use of text-to-speech software to read and reread information and instructions (as some sections are dense with text).
- Provide explicit guidance for students who may have difficulty identifying the activity instructions, or sequence of steps when interacting with the input window throughout the lesson.

Suggestion 2

Lesson section: Vocabulary (*used throughout lesson*)

Task description: Students encounter new terms while performing programming activities (e.g., compiler error, cast object) and revisit terms introduced in previous lessons (e.g., cast/type casting).

Some students may have difficulty:

- reading and understanding new vocabulary (Language, Reading)
- comprehending the use of terms in a new/different context (e.g., coding) (Language)
- comprehending new terms introduced with little information about their meaning (Language)

Teacher adjustment recommendations - adaptations:

- Place key terms and definitions on a presentation slide or white board so that information is presented visually, and remains accessible while students work. As you do, provide students with concrete examples of the new vocabulary words. Show examples of the concepts in the lesson.
- Print out the “[Quick Reference: Quorum Overview](#)” sheet for students to reference any vocabulary definitions introduced in earlier activities, and used in the current lesson. Alternately, create a more comprehensive reference sheet (preferably a shared electronic document), that you, or your students can add to as new operations are encountered and can include additional examples along with the terms.
- Place new vocabulary terms in a running class glossary and assist students in locating them. Alternately, have students bookmark other online glossaries with grade-appropriate definitions, and/or print them out for use (e.g., two examples can be found [here](#) and [here](#)).
- Verbally review vocabulary terms with the whole class at the start of lessons.

Teacher adjustment recommendations - accommodations: None

Suggestion 3

Lesson section: Converting One Type to Another section

Task description: Students read about using “cast” instructions to convert types (e.g., converting a number or integer to text, or vice versa), and the difference between “cast errors” and “compiler errors.”

Some students may have difficulty:

- reading and comprehending information presented briefly (Attention, Reading)
- identifying the key take-aways from the text and examples (Attention)
- thinking flexibly and systematically about command use (Cognitive flexibility: Attention, Reading, Written Expression, Math)
- understanding the process of type casting (or converting types) and how the commands used in the examples convert the values (Language, Reading, Attention)

Teacher adjustment recommendations - adaptations:

- Demonstrate for the class the differences between cast errors and compiler errors by projecting examples on a presentation slide or white board. Check for understanding and work through additional examples, as needed.
- Instruct students to have open the “[Tutorial: Handling Errors](#)” page for reference while working (or, have them print out a hard copy to reference). Remind students of the earlier class discussion you had when completing the “[Introduction to Types and Variables](#)” lesson (if you completed that lesson earlier in the Visual Track lesson sequence).
- Explain in detail how the commands convert the value types; walk students through the process and the rationale; and check for understanding.

Teacher adjustment recommendations - accommodations: None

Suggestion 4

Lesson section: Casting Objects section

Task description: Students read a short section about casting objects in Quorum (using the Math class package as an example) with no description about why such an action would be performed.

Some students may have difficulty:

- comprehending mathematical terms introduced with little information about their meaning (Reading, Math)
- reading and comprehending information presented briefly (Attention, Reading)

Teacher adjustment recommendations - adaptations:

- Direct students to the "[Libraries.Compute.Math](#)" page to review the math actions typically found in scientific calculators. Project a few examples of code that will perform mathematical actions, and draw attention to the details around operators, punctuation, capitalization, etc. required for writing instructions for a computer. Direct students' attention to details that distinguish aspects of programming from each other. Start a discussion about why you might want to cast an object.
- Alternatively, have students skip over this section of the lesson (if there is not time to review sufficiently).

Teacher adjustment recommendations - accommodations: None

Quorum A.8 Ramp Experiment

Adjustment Suggestions

Suggestion 1

Lesson section: Student programming instructions (*throughout the lesson*)

Task description: Student read information and instructions on the web page with no verbal input from the teacher. Across lessons, students must carefully read to find markers that indicate where they are expected to perform an activity, and where to find the necessary information to do so (eg., in some lessons, the output windows include some instruction, while others tell students to look outside the output window for what action to perform).

Some students may have difficulty:

- reading dense and/or complex text on webpages (Reading, Attention)
- recalling or retrieving terminology and concepts presented elsewhere to apply to current work (Language, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- planning and determining an approach towards the independent work time; using a systematic approach while also self-monitoring (Attention/Executive function)

Teacher adjustment recommendations - adaptations:

- At the start of each lesson/set of activities for the day, take time to explicitly review and summarize the main ideas, student activities, and goals for the day verbally, and with the information displayed on a presentation slide or white board.

Teacher adjustment recommendations - accommodations:

- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Offer students the use of text-to-speech software to read and reread information and instructions (as some sections are dense with text).
- Provide explicit guidance for students who may have difficulty identifying the activity instructions, or sequence of steps when interacting with the input window throughout the lesson.

Suggestion 2

Lesson section: Goal 1: Have a face-to-face meeting

Task description: Students move into groups to discuss types and implicit casting, review the symbols associated with types, and work with test calculations to look at implicit casting rules (using the division operator).

Some students may have difficulty:

- comprehending mathematical terms introduced with little information about their meaning (Reading, Math)
- identifying number patterns and mathematical symbols (Math)
- performing mental math calculations (dividing) or recalling math facts in order to comprehend examples and identify whether programs run accurately (math)
- recalling or retrieving terminology and procedures (e.g., type casting, errors) presented elsewhere to apply to current work (Language, Attention)

Teacher adjustment recommendations - adaptations:

- Instruct students to keep open the “[Tutorial: Type Casting](#)” and “[Tutorial: Types and Variables](#)” pages for reference (or, have them print out a hard copy to reference). Project the key points on a slide or whiteboard, and review as a whole class.
- Print out the “Quick Reference: Quorum Overview” for students to reference vocabulary definitions and examples (e.g., integer, number). Alternately, create a more comprehensive reference sheet, that you, or your students can add to as new operations are encountered.
- Project key take aways from the type casting and [errors pages](#) on a presentation slide or white board so it’s accessible as students work.
- Project the test calculations on a whiteboard or slide, and draw attention to the details around the operators and punctuation, and the cause of errors.
- Review difference between integers and numbers. Ask students to share examples of each.

Teacher adjustment recommendations - accommodations:

- Encourage students to use calculators to assist them in reviewing the “test calculations” listed in the lesson, before building a program for Quorum to compute division using integers.
- Support students having difficulty visually distinguishing between integers and numbers, using them correctly, and accurately evaluating what the output should be.

Suggestion 3

Lesson section: Goals 2, 3 and 4: Building two ramps, creating a hypothesis, and running the experiment

Task description: Students work in group to: build ramps for the experiment, generate a hypothesis about which ramp will cause a ball to roll further, and then test their hypothesis.

Some students may have difficulty:

- working effectively and efficiently in groups (Language, Attention/Social Skills)
- wording the instructions in manner that can be understood by others; retrieving and using phrasing to relay their intended meaning (Language, Written Expression, Attention)
- sustaining attention while classmate express their thoughts and comprehending the information shared (Language, Attention)
- reading dense and/or complex text on webpages (Reading, Attention)
- putting their thoughts into words (hypothesis generation) (Language, Written Expression)
- typing or hand-writing responses (Written Expression)
- writing down ideas quickly and accurately (Written Expression, Language)

Teacher adjustment recommendations - adaptations:

- Assign student groupings that will be supportive of learning differences and minimize difficulties that could arise because of social skills or other factors. For example, some students may struggle with positive language with others when they think they “work better alone,” recognizing when they are talking over someone else or saying abrasive things, sustaining attention while others provide ideas.
- Provide guidelines for how students should work together. Guidelines may include: one person speaking at a time; everyone accepts feedback; everyone actively listens, etc.
- Review and model guidelines for students on how to generate a shared hypothesis; suggest ways to split up activity roles during the activities.
- Display project steps on a discussion slide or white board (with additional instruction) for students to reference as they work. Additionally provide instruction about how to measure the distance of each roll.
- As a class, provide several examples of strong hypothesis statements and review the type of information that should be included.
- Model how to set up a spreadsheet to capture all needed data accurately for each ramp.

Teacher adjustment recommendations - accommodations:

- Circulate and model phrasing for students who may have difficulty translating their thoughts to language and who have difficulty retrieving the words they would like to use.
- Work directly with groups that need extra support to make progress with the activity.
- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Offer students the use of text-to-speech software to read and reread information and instructions (as some sections are dense with text).
- Provide opportunities for students to use dictation software to add their ideas to program documentation.

Suggestion 4

Lesson section: Goal 5: Calculating the average.

Task description: Students write a program to calculate the averages of their experiment distance data.

Some students may have difficulty:

- recalling math facts (averages) in order to comprehend examples and identify whether programs run accurately (math)
- performing mental math calculations (subtracting, adding, multiplying, dividing) or recalling math facts in order to comprehend examples and identify whether programs run accurately (math)
- recognizing subtle differences in elements of code (when writing and debugging) related to spelling, capitalization, spacing, punctuation, and symbols (e.g., the difference between “ramp2a” and “ramp2 a” (Reading, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)

Teacher adjustment recommendations - adaptations:

- Project the instructions for manually calculating an average.
- Call attention to the “Activity: Program Averages” input window and point out the instructions to be completed.
- Highlight subtle difference in elements of code for the students.
- Check for understanding of the examples and clarify information when needed

Teacher adjustment recommendations - accommodations:

- Encourage students to use calculators to perform averages, as needed.

Suggestion 5

Lesson section: Goal 6: Conducting an online meeting

Task description: Student groups discuss their experiment results through an “online collaborative meeting.”

Some students may have difficulty:

- expressing/writing thoughts with the correct words and phrases to articulate them, and in a way comprehensible to classmates (Language, Written Expression)
- writing down ideas quickly and accurately (Written expression, Language)
- typing or hand-writing responses (Written Expression)
- sustaining attention while classmates explain their thoughts (Attention)
- using socially appropriate language in communication and collaboration (Language, Attention/Social Skills) expressing their thinking in a way comprehensible to others (Language)

Teacher adjustment recommendations - adaptations:

- Display the meeting guidelines and requirements on a slide or white board for students to reference as they prepare to share with the class.
- Provide support for students to understand in which circumstances a hypothesis would be supported, inconclusive, or not supported by the data.
- Allow enough wait time before calling on any students to give them adequate time to process information and come up with their responses.
- Rephrase and recap student responses to emphasize essential points and to clarify information shared by students that may not be clear to their peers.
- Throughout the discussion, help students by modeling phrasing.
- Near the end of the discussion, explicitly summarize the main points students shared.
- Display the key points from the discussion as a Google Slide.
- If an online format is too challenging to organize, do this last activity component as in-class presentation and discussion.

Teacher adjustment recommendations - accommodations:

- As needed, provide sentence starter templates (verbally, or in writing) to help students formulate their thoughts for the presentation.
- Assist students who are having difficulty putting their thoughts into words.
- Honor requests from students who do not want to present, particularly if the student has language challenges. Offer these students another way to engage in the group presentation to demonstrate their contributions and understanding.
- Provide opportunities for students to use dictation software to add their ideas to an electronic document in place of describing them quickly in front of others.
- Provide opportunities for students to use dictation software to add their ideas to program documentation. Set aside space in a secluded corner of the room, hallway, or somewhere else outside of the general classroom for students to use dictation software if needed.

Quorum A.9 Commenting Code

Note about “Commenting Code”

- Code comments are used starting in Visual Track, Section A lesson 4A. [Getting Started with Games](#)” and as such, it will be helpful to share this activity with students prior to moving to 4A.
- This is a very short activity, so consider ways to combine it with other content to fill your entire class period (e.g., you may choose to introduce it as part of “Getting Started with Games”).
- Look for additional examples to share with students so they can see how comments are typically used in code, and why.

Adjustment Suggestions

Suggestion 1

Activity section: Entire tutorial

Activity description: Students learn how two different methods for commenting code in Quorum

Some students may have difficulty:

- determining if/when to use one comment method over the other (Language)
- comprehending the use of terms (comments) in a new/different context (e.g., coding) (Language)
- understanding the use of general phrases in place of code in the examples such as “class main,” “action main,” and “output” (Language, Reading)
- recognizing subtle differences in elements of code/code comments when writing related to symbols and spacing (e.g., the difference between “//integer j=25” and “/integer j=25” or “/*integer k=26” and “/*integer k=26/” (Reading, Attention)

Teacher adjustment recommendations - adaptations:

- Provide real-world example situations that show single line and multiple line comments in use, explicitly pointing out why a student would pick one comment format over the other (and the kinds of notes that are included), and places where either will work.
- Engage students in discussion about why it’s important to use comments (e.g., to remember why you wrote certain elements of a program/what happens at various points, and to communicate what was done to another human coder)
- Draw attention to the use of the double forward slashes in a single line/end-of-line comment, and the use of a single forward slash and asterisk for comments that require multiple lines. Demonstrate what happens when the slashes are not included, or if a single or double slash is used mistakenly (e.g., using “/* on the first line and closing on the second line with “*/” will generate an error).

Teacher adjustment recommendations - accommodations: none

Quorum A.10 Tortoise Vs. Hare

Note: adjustment suggestions only reflect the website text/experience and not the actual offline IDE (due to technical challenges).

Adjustment Suggestions

Suggestion 1

Lesson section: Student programming instructions (*throughout the lesson*)

Task description: Student read information and instructions on the web page with no verbal input from the teacher. Across lessons, students must carefully read to find markers that indicate where they are expected to perform an activity, and where to find the necessary information to do so (eg., in some lessons, the output windows include some instruction, while others tell students to look outside the output window for what action to perform).

Some students may have difficulty:

- reading and comprehending dense and/or complex text providing information and instructions for programming in the offline IDE (Reading, Language, Attention)
- recalling or retrieving terminology and concepts presented elsewhere to apply to current work (Language, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- planning and determining an approach towards the independent work time; using a systematic approach while also self-monitoring (Attention/Executive function)
- moving between two windows on the computer: one with the instructions and the other in the offline IDE where the instructions need to be executed (Language, Attention)

Teacher adjustment recommendations - adaptations:

- At the start of each section, project the instructional page and take time to explicitly review, read aloud, and summarize main objectives, vocabulary, concepts and student programming activities as a group.
- Identify previously learned terminology, concepts and functions that students will need to use in the present activity and review these as a class.
- Create and provide reference documents that contain guidelines regarding how to call actions or share the information in the [Quorum overview page](#).
- Offer the option for students to work in pairs so that one student can display the instructions on his/her screen while the other has the offline IDE open. Alternately, project the instructions, so they can be viewed by the whole class while students work.
- Circulate and assist students experiencing difficulty determining how to approach their independent programming time; explicitly provide guidance regarding systematic strategies for working through the tasks and assist students in monitoring their work.

Teacher adjustment recommendations - accommodations:

- Read the instructions aloud to students or allow those who need text-to-speech software to use it.

Suggestion 2

Lesson section: Understanding the Provided Template and Goal 1: Calling Actions on Objects

Task description: Students read instructions for writing a program to play a race.

Some students may have difficulty:

- reading and comprehending dense text (Reading, Attention)
- recognizing subtle differences in elements of code (when writing and debugging) related to spelling, capitalization, spacing, punctuation, and symbols (Reading, Attention)
- using and understanding x- and y- coordinates (Mathematics)
- locating the lines of code from which they must write additional code and expand (Reading, Attention)

Teacher adjustment recommendations - adaptations:

- Project the assignment template code on a screen and read through the information as a group, while also talking through the additional information presented on the instructions page. Draw students' attention to the use of punctuation, parentheses, spacing, and capitalizations that are needed for the program to run correctly.
- Direct students to the reference sheet they used in A.4 "Getting Started with Games" (if you implemented that recommendation) and add any new code from this assignment template to that reference sheet. Encourage students to keep the reference handy while working on future projects in the offline NetBeans/Sodbeans environment.
- Offer the option for students to work in pairs so that one student can display the instructions on his/her screen while the other has the offline IDE open. Alternately, project the instructions, so they can be viewed by the whole class while students work on their own computers.
- Circulate and assist students in debugging for errors by modeling the necessary steps and providing example approaches.

Teacher adjustment recommendations - accommodations:

- For students who need additional support, provide instructions as a sequence of steps to clearly indicate the order in which students should write their program.
- For students having difficulty identifying the lines of code where the program needs to be written or expanded, assist them in locating lines of code.
- Circulate and assist students having difficulty with terminology or distinguishing information to process vs. instructions/to-dos on the instructional page.
- Read the instructions aloud to students or allow those who need text-to-speech software to use it.
- Provide additional review opportunities for students who need more time on visual spatial constructs like proportion and x- and y- coordinates before engaging in the lesson's problem-solving task.

Quorum A.11 Atmospheric Calculations

Adjustment Suggestions

Suggestion 1

Lesson section: Student programming instructions (*throughout the lesson*)

Task description: Student read information and instructions on the web page with no verbal input from the teacher. Across lessons, students must carefully read to find markers that indicate where they are expected to perform an activity, and where to find the necessary information to do so (eg., in some lessons, the output windows include some instruction, while others tell students to look outside the output window for what action to perform).

Some students may have difficulty:

- reading dense and/or complex text on webpages (Reading, Attention)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- planning and determining an approach towards the independent work time; using a systematic approach while also self-monitoring (Attention/Executive function)

Teacher adjustment recommendations - adaptations:

- At the start of each lesson/set of activities for the day, take time to explicitly review and summarize the main ideas, student activities, and goals for the day verbally, and with the information displayed on a presentation slide or white board.

Teacher adjustment recommendations - accommodations:

- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Offer students the use of text-to-speech software to read and reread information and instructions (as some sections are dense with text).
- Provide explicit guidance for students who may have difficulty identifying the activity instructions, or sequence of steps.

Suggestion 2

Lesson section: Vocabulary (*used throughout lesson*)

Task description: Students encounter and are expected to remember mathematic terminology (quotient, average, difference, and decade) and terms from previous lessons while performing programming activities (e.g., instantiation, explicit casting)

Some students may have difficulty:

- reading, recalling or retrieving terminology and concepts presented elsewhere to apply to current work (Language, Attention)

Teacher adjustment recommendations - adaptations:

- Place key terms and definitions on a presentation slide or white board so that information is presented visually, and remains accessible while students work. As you do, provide students with concrete examples of the new vocabulary words. Show examples of the concepts in the lesson.
- Verbally review vocabulary terms, including mathematical terms, with the whole class at the start of lessons.

Teacher adjustment recommendations - accommodations: None

Suggestion 3

Lesson section: Goal 2: Create a customized calculator to analyze large data sets

Task description: Students review dense raw data and learn how averages are calculated traditionally (the pen and paper method).

Some students may have difficulty:

- reading directions, particularly when they are required to read dense text on webpages (Reading, Attention)
- interpreting and understanding data represented in different tables (Reading, Attention, Math)
- comprehending and using mathematical principles (e.g., averages); thinking flexibly and systematically (Math)
- recalling or identifying background information regarding CO₂ in the air and understanding the unit of measure of “parts per million (ppm)” (Reading, Attention, Math)

Teacher adjustment recommendations - adaptations:

- Pull [the data needed for the activity](#), and present it in a more readable format. For example, remove variables that aren’t directly relevant to the lesson activity (the only variables needed are year, month, and average CO₂ levels for each month and year).
- As a class, review the traditional methods for “finding a trend” in a large data set and demonstrate the steps on a white board and check for student recall mathematical terms and concepts.
- Provide the sequence of steps required for the activity on a white board or presentation slide so students can see them more clearly, without the surrounding text.
- As a class review the general concept regarding CO₂ in the air to assist students in recalling information learned previously and in comprehending the background knowledge needed to help interpret the data

Teacher adjustment recommendations - accommodations:

- Allow students to use calculators.
- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Circulate and assist students in interpreting charts and graphs; provide guidance and instruction when needed.

Suggestion 4

Lesson section: Goal 2: Create a customized calculator to analyze large data sets

Task description: Students write a program for a customized calculator.

Some students may have difficulty:

- comprehending and executing the instructions (Language, Reading, Attention)
- performing mental math calculations (subtracting, adding, multiplying, dividing) or recalling math facts in order to comprehend examples and identify whether programs run accurately (Math)
- recalling math facts (averages) in order to comprehend examples and identify whether programs run accurately (Math)
- distinguishing instructions/to-dos from other information when embedded in with general information (Reading, Attention)
- recalling the use of each command and the type of variable that should be entered in the command (Language)
- identifying spelling or syntax errors in coding/debugging (e.g., the difference between “Square1” and “squar1”) (Reading, Written Expression, Attention)

Teacher adjustment recommendations - adaptations:

- Project the activity instructions and input statement examples on a screen and read though the information as a group. Draw students’ attention to the use of punctuation, parentheses, and capitalizations that are needed for the program to run correctly.
- If you choose to use the online IDE, print the instructions located below the compiler window on a single sheet of paper for students to reference as they complete the activity (and to minimize the amount of scrolling up and down on the screen needed to review instructions and complete the activity).
- If students have difficulty with consistent spelling and capitalization, encourage them to use an electronic document where they write commands and codes and later copy and paste the commands into the programming environment rather than retyping commands and codes each time.
- Present previously-introduced commands visually (project on a screen or slide, or provide in a handout) to refresh students’ memory from previous lessons. Check for understanding, and clarify and rephrase when necessary.

Teacher adjustment recommendations - accommodations:

- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, and clarify when needed.
- Circulate and assist students in writing code by modeling the necessary steps and providing examples.

Suggestion 5

Lesson section: Goal 3: Data analysis – Interpreting and analyzing the processed data

Task description: Students consider questions about data patterns and engage in class discussion about the causes of atmospheric CO₂ increases.

Some students may have difficulty:

- reading and comprehending questions (Reading, Attention)
- expressing thoughts with the correct words and phrases to articulate them, and in a way comprehensible to classmates (Language, Written Expression)
- sustaining attention while classmates explain their thoughts (Attention)
- using socially appropriate language in communication and collaboration (Language, Attention/Social Skills) expressing their thinking in a way comprehensible to others (Language)

Teacher adjustment recommendations - adaptations:

- Provide support for students to understand in which circumstances a hypothesis would be supported, inconclusive, or not supported by the data.
- Check for understanding of discussion prompts; clarify and rephrase when needed.
- Review group discussion guidelines for students (e.g., only one person talks at a time; listen actively; people can share differences of opinion) and provide feedback. Guidelines can be posted in the classroom as a visual reference and appropriate language can be modeled.
- Circulate and rephrase and recap student responses to emphasize essential points and to clarify information shared by students that may not be clear to their peers.
- Near the end of the discussion, explicitly summarize the main points students shared.
- Display the key points from the discussion as a Google Slide.

Teacher adjustment recommendations - accommodations:

- Assist students who are having difficulty putting their thoughts into words.
- Provide opportunities for students to use dictation software to add their ideas to an electronic document in place of describing them quickly in front of others.

Quorum A.12 Quiz for Variables and Types

Note to teachers about “Quiz for Variables and Types”

- If you have only used lessons/activities in Visual Track, Section A, there will be some content in this quiz, and phrasing used that was not covered in Visual Track, Section A, so review carefully before asking students to complete.

Adjustment Suggestions

Suggestion 1

Task description: Students are provided questions to respond to; both short-answer and essay.

Some students may have difficulty:

- reading and comprehending questions (Reading, Language, Attention)
- reading dense text (Reading, Attention)
- recalling vocabulary and concepts (Language)
- concentrating, particularly if they need additional time to process information and complete work (Attention)
- putting their thoughts into words (Written Expression, Language)
- typing or hand-writing responses (Written Expression)
- writing down responses quickly and accurately (Writing, Attention, Language)
- expressing their thoughts/ideas in writing in a way comprehensible to others (Writing, Language)

Teacher adjustment recommendations - adaptations:

- Provide instructions about how and where students should respond to the questions.
- Check for understanding of question prompts; clarify as needed.
- Review and select questions before administering to students to ensure content is covered in the Visual Track, Section A lessons, and that terminology used matches with that content.
- Provide guidance about length and richness of content for each response type (short-answer, essay format).

Teacher adjustment recommendations - accommodations:

- Offer students the use of text-to-speech software to read and reread questions, and to listen to and edit their own written responses.
- For students who need assistance with reading and comprehending the material, read it aloud to them, check for understanding, clarify when needed.
- Provide opportunities for students to use dictation software to add their ideas to electronic documents (like Word or Google documents) or to write assessment responses. Set aside space in a secluded corner of the room, hallway, or somewhere else outside of the general classroom for students to use dictation software if needed.