



blackbird

EXPRESSIONS & EQUATIONS TEACHER GUIDE

PACING GUIDE

'This unit is designed to take about a week of class time to complete. As with all lessons on Blackbird, this sequence has been designed to allow students to work at their own pace, so students will move through the content differently. We have provided suggestions for support and extension in the Stage Guides that follow. The sequence has been completed when students have successfully finished the 'Guided Project' activity.

USING STAGE GUIDES

'An overview of major concepts introduced in the stage is listed below. Use the 'Guiding Question' at the start of the stage and 'Probing Questions' once students have finished the activities as a wrap-up. Probing questions can facilitate conversations about what students learned to program, provide opportunities for questions, important concept review, and connection back to the math concepts they're learning and reviewing. Pick your favorite 2 or 3 to use in class.

- **Stage 1 - Evaluating Expressions**

- **Guiding Question:** "How can I program a computer to evaluate a mathematical expression?"
- **Probing Questions**
 - How do expressions in math class and programming compare?
 - Students might describe how variables need to be declared in a computer program before they can be used in a program. They might also mention the line-by-line nature of programs and the different operators used in programming, / for division for example.
 - What does it mean to 'evaluate' an expression? How can you see how a computer program evaluates an expression?
 - 'Evaluate' in a math or programming setting means to carry out the operations in an expression to find a single value for the expression. Students can 'see' how a computer will evaluate an expression in Blackbird by using the 'Debug' mode. This allows them to step down a program line-by-line. When they move their mouse over the operators in an expression they'll see how the computer is evaluating those operations and the resulting value of the expression.

- What other area formulas do you know that you would want to add to a computer program? How would you do it?
 - Students might know rate formulas ($v = d/t$), the quadratic formula, or any number of area formulas. The key to incorporating each into a computer program is being able to break them down into pieces that can be added on a line of code. In lesson 1.4 students are introduced to this idea when building the formula for the area of a circle. You can challenge students to create a program in the Workshop that solves a complex formula.

- **Extension Ideas**

- At the end of each lesson there is an 'Extra Challenge' for students to attempt. These challenges take students into the Workshop where they tinker with the code they created within a lesson.
- The information displayed on the canvas in these lessons is simply the result of the expressions calculated by the program. Ask students to explore 'concatenation' in the Docs. Challenge them to make the information displayed on the canvas in their programs more informative.
- After students have completed the Guided Project for the stage, challenge them to think of another 'solver' program they could build that evaluates another equation or formula.

- **Support Ideas**

- The bottom of most tasks have a 'Deep Dive' button that students can press to see more information and explanations of the code they are typing. Reminding students to slow down and read these sections can help them make sense of the work they are doing.
- The 'Debug' button can be used at any time. There are places within lessons that step students through their programs, but making routine use of the Debug button can help students understand how their expressions are being evaluated, and how lines of code are being executed by the program.

- **Lesson 1.1 - Sum Machine**

- **What students have created at the end of the lesson:** A very short program that adds two numbers and displays the sum on the canvas.
- **Important programming concepts introduced:**
 - Assigning and declaring variables
 - Using operators like + and =
 - Using the write() function
- **Important math concepts:**
 - Expressions
 - Using variables in place of numbers in an expression
 - Evaluating expressions

- **Lesson 1.2 - My Dear Aunt Sally**

- **What students have created at the end of the lesson:** A program that evaluates several similar expressions that are evaluated differently based on the order of operations.
- **Important programming concepts introduced:**
 - Mathematical operators like * and /
- **Important math concepts:**
 - Order of operations
 - Factoring expressions using ()

- **Lesson 1.3 - Area Calculator**

- **What students have created at the end of the lesson:** A program that calculates the area of a Rectangle object displayed on the canvas.
- **Important programming concepts introduced:**
 - Shape objects
 - Object properties
 - Using properties in expressions
- **Important math concepts:**
 - Rectangle area formula

- **Lesson 1.4 - Area Calculator, too**

- **What students have created at the end of the lesson:** A program that calculates the area of a Circle object displayed on the canvas.
- **Important programming concepts introduced:**
 - Built-in constants (PI)
- **Important math concepts:**
 - Breaking complex formulas into smaller parts

- **Lesson 1.5 - Let's hear from the crowd**

- **What students have created at the end of the lesson:** A program that collects data from a user and incorporates it into a mathematical expression.
- **Important programming concepts introduced:**
 - Using the prompt() function
 - Using the Number() function
 - Data types (numbers and strings)
- **Important math concepts:**
 - Review of using variables in expressions