Lesson Plan

Description

Students roll two dice to determine the effort and the load on a lever. Students will then use an algorithm to solve for the missing value that would balance the lever.

Learning Outcomes

- A lever is a simple machine that uses a beam and a fulcrum
- A lever is balanced when the effort and the load is the same
- An algorithm is a series of steps used by a computer program to perform a task
- Conditional statements help computers make decisions
- Pseudocode is when code is written out

Specific Expectations

Science, Strand A: STEM Skills and Connections
A2.1 write and execute code in investigations and when modelling concepts, with a focus on decomposing problems into smaller steps

Science, Strand D: Structures and Mechanisms
D2.3 identify the six basic types of simple machines: lever, inclined plane, wedge, pulley, wheel and axle, and screw

Math, Strand C: Algebra
C2.2 determine what needs to be added to or subtracted from addition and subtraction expressions to make them equivalent
C3.1 solve problems and create computational representations of mathematical situations by writing and executing

Introduction

A lever is a simple machine made of a beam and a fulcrum. The fulcrum is the point on which the beam pivots. When an effort is applied on one end of the lever, a load is applied at the other end of the lever that will move a mass up. The location of the fulcrum will determine how much effort is required to lift a mass.

For this exercise, we will consider a first-class lever that has the fulcrum located half-way between the load and the effort. Regardless of the configuration, the mass will be lifted once the effort is greater than the load but be balanced if they are the same. We will be using this concept of equilibrium to help conceptualize math equations and how computer programs can do those calculations for us.

Coding at its simplest is about following an algorithm, or a series of step-by-step instructions. In this activity, students will follow a sequence of steps to balance a math equation built into a lever. The code includes conditional statements, starting with an If statement which us used to help computers make a decision. For example, IF it's raining, then get an umbrella. The action of getting the umbrella is conditional on the IF statement is satisfied.

This lesson can be done in multiple ways, individually on the worksheet, in partners, in groups or as a repetitive exercise using laminated copies as white boards, repeating the exercise multiple times.

Action

To do this activity, students will need to use two dice.

First the student will roll two dice to determine the effort. Next, they will roll two dice to determine the load. Based on the results, they will have to determine what needs to be added (effort or load) and how much to balance the lever.

To determine this, they can use a coding algorithm to follow the steps to find the missing number. First, a computer would have to figure out if the effort or the load needs more to balance the lever. Here is the algorithm it would use:

```
If Effort > Load
    Add to Load
If Effort < Load
    Add to Effort
```

Since there are two possible outcomes, this can be simplified by using an Else statement. An Else statement provides an alternative. If it's raining, get an umbrella, else get sunscreen. This would let us rewrite the statement as follows:

```
If Effort > Load
    Add to Load
Else
    Add to Effort
```

Once it's determined what side to add to, a computer program would have to determine how much to add to balance the lever. Since it is dependent on the side, it would be nested into the algorithm depending on what condition is met.

```
If Effort > Load
```

Missing Number = Effort - Load
Else
Missing Number = Load – Effort

Using this algorithm, students can solve for the missing number needed to balance the lever.

For example, if a student rolls a 7 for the effort, and a 3 for the load:

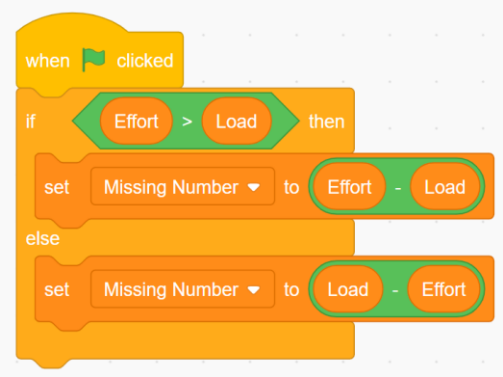
If Effort > Load (true)
Missing Number = Effort – Load
Missing Number = 7 – 3
Missing Number = 4

Since an algorithm goes from top to bottom and the first condition is met, the second equation would not be used. In the reverse instance, if the Load is larger than the Effort, the first condition is not met so the computer (or student) would use the second equation. For example, if a student rolls a 12 for the effort and a 6 for the load:

If Effort > Load
Missing Number = Effort – Load (False)
Else

Missing Number = Load – Effort (true)
Missing Number = 12 – 6
Missing Number = 6

An algorithm like this could be put into a program such as Scratch and work the same way. It would look like this:



We created a full scratch program that uses this concept for students to test their work. To see this program and look at how the coding works, follow this link: https://scratch.mit.edu/projects/708841094

For students to practice this, there is an accompanying handout, that also includes the code as reference. They'll roll the dice, fill in the load and effort, then solve for the missing number. To reinforce the coding, students can write out the accompanying code. Code that is written out is called pseudocode.

Consolidation/Extension

- You can challenge students by changing the number of dice used. For example, have students roll three or four dice for the effort and the load.
- You can challenge students by changing the problem, for example provide students with the missing number, have them roll a die for effort, then have them solve for the load.
- Have students write an algorithm that has a third option, if the effort and the load are the same

Accommodations/Modifications	Assessment
• To use this as an ongoing resource for practice or group work, laminate the sheets to be used as white boards • Allow students to use dry-erase markers and complete the sheet multiple times for practice rather than one time in pencil	• Assess your student's ability to balance expressions using addition and subtraction • Assess your student's ability to write pseudocode

Additional Resources

Materials Needed:

- 2 dice per student
- 1 sheet per student
- Laminating sheets (if using as on-going resource)
- Dry-erase markers (is using as on-going resource)