|  |  |
| --- | --- |
| http://compsci.appstate.edu/sites/compsci.appstate.edu/files/imagecache/slideshow/slideshow/ASU_compsci_logo.png  **The CS4ALL NSF Supported Program** | https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcQGzOU-XT8XZWIBUwiPs2jjgixLO3CvrEyNq90lu1dbXJ0BQume  [**https://cs.appstate.edu/cs4all/**](https://cs.appstate.edu/cs4all/) |

**Activity Title:** SNAP Calculator

**Introduction/Motivation:**

This module is designed to introduce some of the more advanced applications of SNAP. The goal is to become familiar with handling input/output functions as well as the mathematical functions in SNAP while improving general programming skills.

**Background:**

To be successful, students should have a basic understanding of the mathematics they are trying to have their calculator perform. A rudimentary understanding of programming and previous experience with SNAP will provide added benefits.

**Materials List:**

* Access SNAP from: <https://snap.berkeley.edu/snap/snap.html>

**Assessments:**

**Pre-Assessment:**

* Understand Order of Operations.
* Familiarity of the mathematics involved with any advanced functions.

**Preparation:**

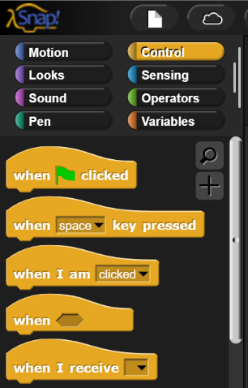
* Access SNAP from: <https://snap.berkeley.edu/snap/snap.html>

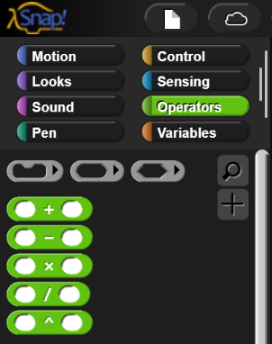
**Basic Lab Activity:**

1. Open a new project in SNAP.
2. Create a program that does the following things:

* Accepts inputs from the user.
* Performs the desired simple operation: +, -, x, or /.
* Outputs the result to the user.

1. Getting the input is going to be the most difficult part of this lab. There are several ways to implement the input portion and some will be easier than others. Below are some tips and things to consider as you begin writing your program.

* Are you going to have your user type in the number, click a button, or use a slider?
* Can you handle decimals and negative numbers?
* If you choose to create buttons on the screen there is an operation under the Control tab that detects when a sprite is clicked on. You may find this useful.
* How will you handle inputs that are non-numeric?
* How will you store user inputs?

1. Performing the operations themselves is fairly straightforward. Look under the Operators tab for the four required operators. Below are some things you may want to consider for this step.

* What will happen if a user attempts to divide by 0? You may want to consider adding a check for this.
* How will you know when the user is ready to perform the operation?
* What if a user makes a mistake? Do they have to restart the program? Can they backspace or would they have to clear the whole operation?

1. Displaying output is the simplest part of this lab. Use a variable to store the output of the operation and display it on the screen.

**Follow Up Questions for Basic Lab:**

1. Describe how you would implement a user interface other than the one you chose to implement. For example, if you used on-screen clicking, describe how you would use the keyboard for input.
2. Implement the ability of a user to continue performing operations on the result. For example, if a user wants to multiply 9.8 by 3 and then add 4, make it so that the user can do the multiplication and then press or type “+” followed by a “4” and get the correct answer.

**Advanced Lab Activities:**

1. Implement a square operation.
2. Implement a square root operation.
3. Implement an “n” power operation.
4. Implement an “nth” root operation.
5. Implement the three basic trigonometric functions.

**Follow Up Questions for Advanced Lab:**

1. I’m feeling a little lazy. How would you create a secant, cosecant, and cotangent function?
2. Suppose I wanted a factorial function. Explain how you would have the program do this.
3. Explain how you could create a calculator for matrices.

**Programming Tips:**

1. Save. Often.
2. Watch out for infinite loops. You may want to include a step that looks for a key to stop the program.
3. Be wary of weird inputs by the user. What will happen if the user inputs an unexpected character?