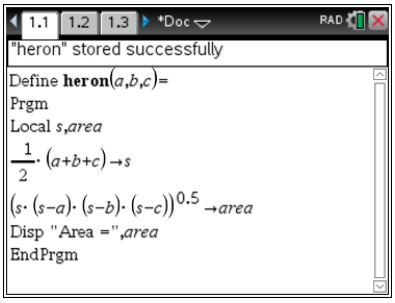
Unit 2: Assigning Values to Variables | **Skill Builder 3: Functions and global variables**

In this third lesson for Unit 2, you will learn about a function's impact on global variables.

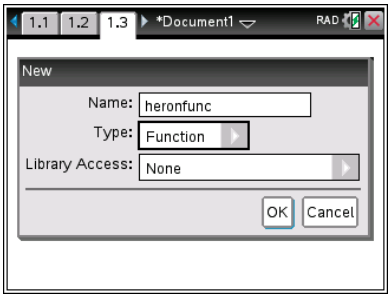
Objectives:

- Compare the impact of a function on global variables with that of programs
- Copy and Paste code from one program to another or within a program

1. Open the document that contains the **heron** program seen at the right.

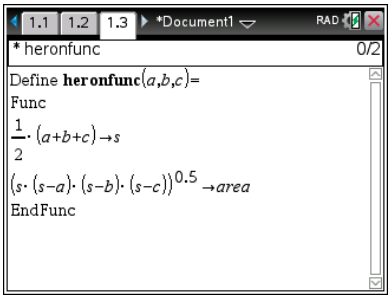


2. Add another Program Editor to the document by pressing **ctrl+doc** and selecting **Add Program Editor**. Create a new program using a unique name, and change the Type to **Function**.



3. Add the arguments *a, b, c* in the parentheses.

4. Copy the two assignment statements from the program **heron** to the new function.



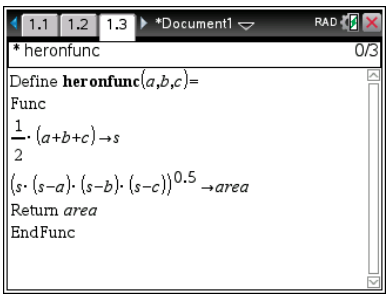
To *Copy* and *Paste* on a handheld:

- Hold the **SHIFT** key down while moving the text cursor over the desired text.
- Press **ctrl+C** to copy the selection to the clipboard.
- Move from the program to the function, and place the cursor in the desired position.
- Press **ctrl+V** to paste the selection.

Recall from Unit 1 that a function must Return a value.

5. Add the statement
   Return area
   at the end of the function.

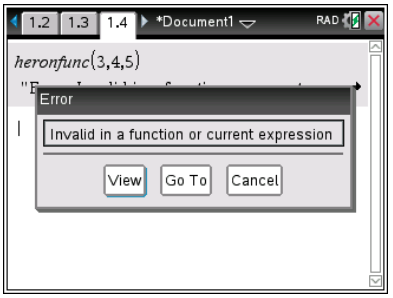
Do not use the **Disp** statement in the function.



Tech Tip: On the TI-Nspire™ App for iPad®, tap and hold to select text and drag ends to highlight all desired text. Select *Copy,* and then tap and hold and select *Paste*.

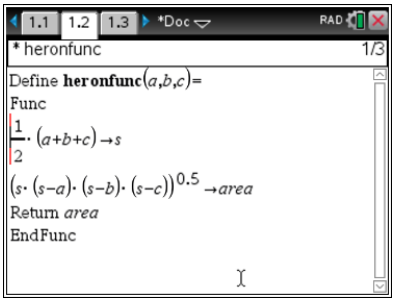
6. Prepare to run the function by selecting **ctrl+R**. Be sure to supply the three argument values. Press **enter**.

An error message is displayed.

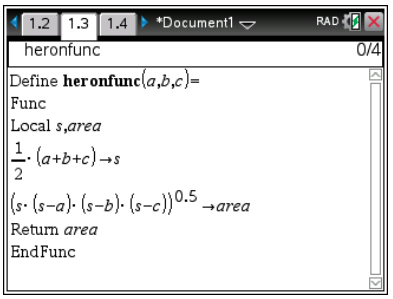


What is wrong?

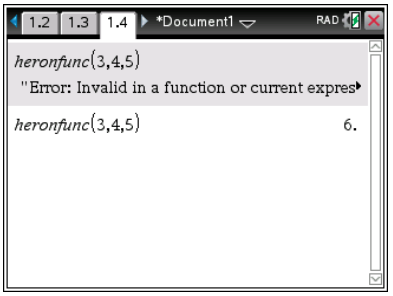
The variables **s** and **area** are global variables since they are not *declared* **Local**. Global (problem) variables are not permitted in a function though global variables are permitted in a program. Recall that the purpose of a function is to return a value. To protect global variables from inadvertent changes ('side effects'), functions cannot modify global variables.



7. To fix this error, add the statement **Local s, area** at the top of the function.



8. Store, and run the function again.



Teacher Tip: The **Return** statement can be placed anywhere in a function but once it is executed the function terminates and no further statements are processed. This becomes significant in later units that involve branching (**If** statements) and looping.