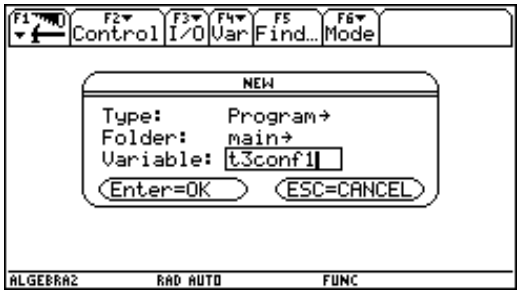
Basic Programming for the TI-92 Plus

By: Joshua L. Springman
Math Teacher Northmont High School
4916 West National Rd.
Clayton, Ohio 45315
jspringman@northmont.k12.oh.us

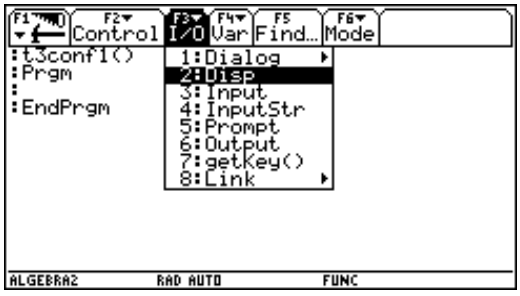
Starting a new program:

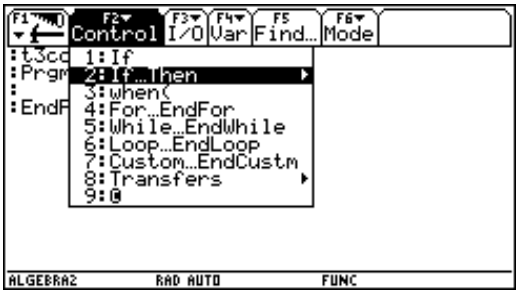
- Press the **APPS** button.
- Arrow down to the **Program Editor** or press 7.
- Arrow down to **New** or press 3.
- Select
  - **Type: Program**
  - **Folder:** The name of the folder where you want the program to be saved.
  - **Variable:** Type a name for your program that is up to 8 characters long, beginning with a letter. It may contain numbers, but it may not contain any other symbols.
- Press enter twice.

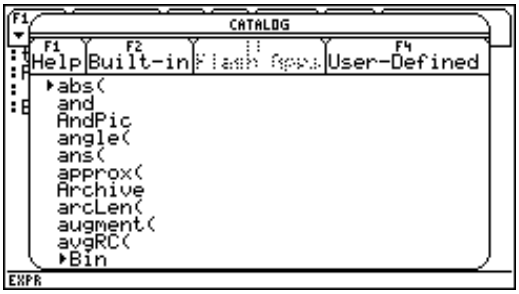


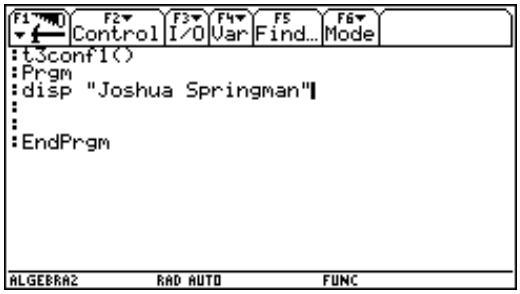
General Information:

- The name of the program always appears in the first line of the program followed by ( ).
- All lines of programming must be contained between the **Prgm** and **EndPrgm** lines.
- You can select commands from the **F1-F6** menus, from the **Catalog**, or just type them in.
- Commands are not case sensitive. (Upper case/ Lower case, no matter)









Display Command: Application 1 (Quotation Marks)

- Can be used to print letters, phrases, or numbers on the input/output screen.
- Type **Disp** or select the command from the **F3** menu.
- Following the command, place a quotation mark, the text you want to appear on the screen, and finish the line with another quotation mark.

Example: **:**Disp "Name" (When the user runs the program, Name will appear on the screen)

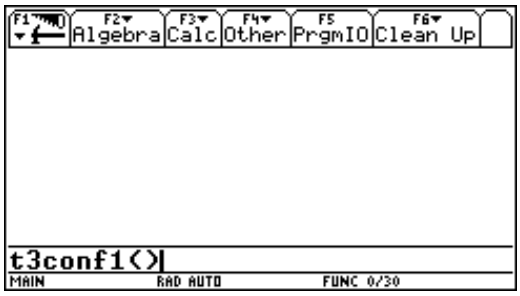
Example Program:

```
t3conf1()
Prgm
Disp "Joshua Springman"
EndPrgm
```

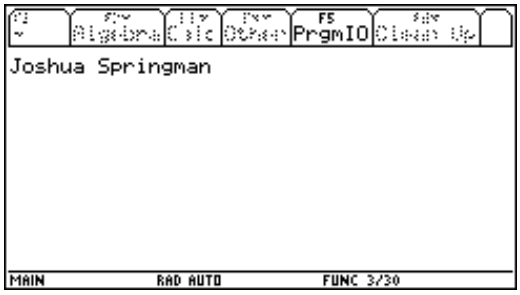
Running a Program:

- In the entry line of the home screen, type the name of the program followed by ( ).
- The program will run on the I/O screen (input/output).

Before running



After running



Storing a Value to a Variable:

- **STO▸** is the "Store" command. It is the button located to the left of the spacebar.
- Enter a value, press **STO▸**, enter the name of the variable

Examples: **:**5 → g (Whenever g is used in a calculation, 5 will be substituted for g.)
:l*w → a (Will calculate l*w and store that value as a.)

Localizing Variables:

- You will receive an error message if you try to have a program store a value to a variable, that is used by another program, defined as a function, used for the name of a Cabri drawing, etc....
- You can localize variables so that their values are only stored while the program is running.
- Type **Local** or select the command from the **F4** menu.
- List the variables you are going to use in the program, separated by commas.

Example: **:**Local a,b,c (While the program is running, values can be stored for a,b,c; however, when the program finishes the values for a, b, and c will return to their original value.

Display Command: Application 2 (No Quotation Marks)

- Can be used to print the values of variables and answers to calculations on the I/O screen.
- **Disp** followed by a variable or a calculation on a variable(s) without quotation marks will print the value of the variable or the answer to the calculation.

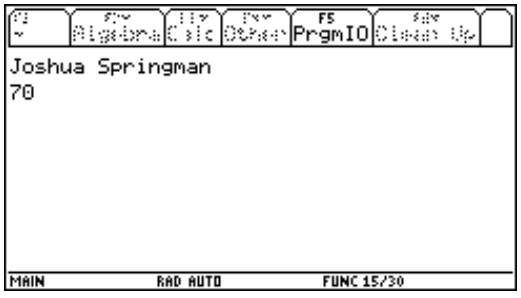
Examples: **:**Disp 5*8 (Will print 40 on the I/O screen.)
:Disp h (Will print the value of h on the I/O screen.)
:Disp a*x (Will print the answer to a*x. The answer depends on the values that were previously stored for a and x.)

Clearing the Input / Output Screen:

- **ClrIO** this command clears the programming screen.

Example Program:

```
t3conf1()
Prgm
ClrIO
Local g,f
Disp "Joshua Springman"
5 →g
14 →f
Disp g*f
EndPrgm
```



This program will clear the I/O screen, localize variables g and f, print my name, store 5 for g, store 14 for f, and print 70 on the I/O screen.

Input Command:

- The input command allows the user to enter a new value for a variable every time the user runs the program.
- Type **Input** or select the command from the **F3** menu.
- Following the input command, enter the name of a variable. The user's value will be stored as this variable.

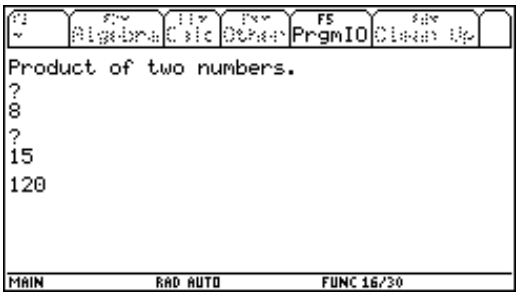
Example: **:**Input h (The program will pause until the user enters a value and presses enter. Whatever value the user enters will be stored as h. The user will only see a ? on the screen.)

- The input command can also function as a display command.
- Type **Input,** a phrase in quotation marks (usually a description of the variable), followed by a comma and the name of a variable.

Example: **:**Input "Height =",h (In this example, the user will see Height =, rather than just a ? .)

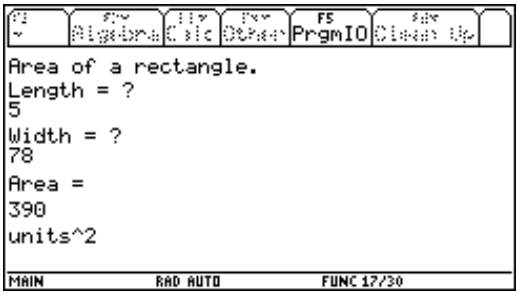
Example Program:

```
t3conf1()
Prgm
ClrIO
Local g,f
Disp "Product of two numbers."
Input g
Input f
Disp g*f
EndPrgm
```



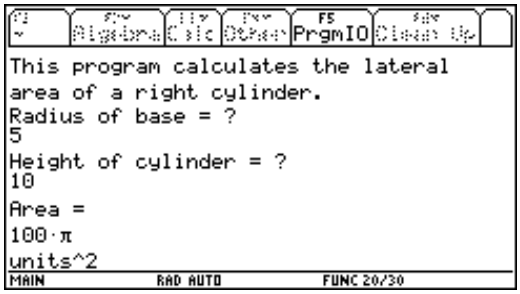
Example Program: Area of a rectangle

```
t3conf1()
Prgm
ClrIO
Local  g,f
Disp  "Area of a rectangle."
Input  "Length = ?",g
Input  "Width = ?",f
Disp  "Area = "
Disp  g*f
Disp  "units^2"
EndPrgm
```



Example Program: Lateral Area of a Right Cylinder

```
lartcyl()
Prgm
ClrIO
Local r,c,h
Disp "This program calculates the lateral"
Disp "area of a right cylinder."
Input "Radius of base = ?",r
Input "Height of cylinder = ?",h
2*π*r ➜c
Disp "Area = "
Disp c*h
Disp "units^2"
EndPrgm
```



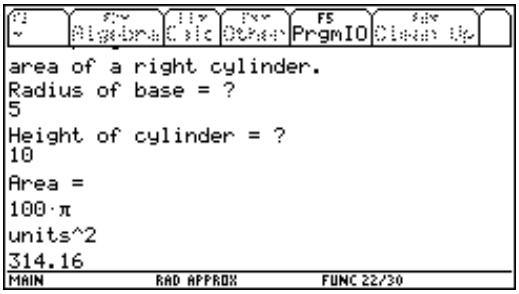
Setting Modes in Programs: Exact / Approx and Display Digits:

- If you want the program to operate with specific mode settings, you can change them before you run the program, or you can have the program set them every time it runs.
- While editing the program, the **F6** menu lists the different mode settings that can be changed during the operation of a program.
- Move the cursor to a blank line in the program (prior to the needed change), and press **F6** and select the mode setting of your choice.

Example: setMode("Display Digits","FIX 2") (This will set the calculator to two decimal places of accuracy.)

Example Program:

```
lartcyl()
Prgm
ClrIO
Local r,c,h
Disp "This program calculates the lateral"
Disp "area of a right cylinder."
Input "Radius of base = ?",r
Input "Height of cylinder = ?",h
setMode("Exact/Approx","EXACT")
2*π*r ➜c
Disp "Area = "
Disp c*h
Disp "units^2"
setMode("Display Digits","FIX 2")
setMode("Exact/Approx","APPROXIMATE")
Disp c*h
EndPrgm
```



Output Command:

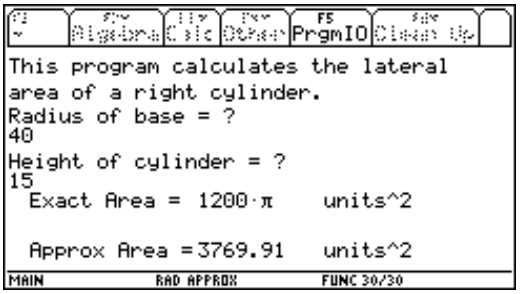
- This command allows you to display information on the I/O screen at specific coordinates.
- Type **Output** or select the command from the **F3** menu.
- Type a value for how far down from the top of the screen you want the information displayed, followed by a comma.
- Type a value for how far to the right from the left side of the screen you want the information displayed, followed by a comma.
- The final area after the comma can be used like a display command (Quotes or No Quotes).

Example: **:**Output 10,80,"TI-92's rule!!" (TI-92's rule!! will appear 10 units down and 80 units to the right.)

:Output 30,120,8*90 (720 will appear 30 units down and 120 units to the right.)

Example Program:

```
lartcyl()
Prgm
ClrIO
Local r,c,h
Disp  "This program calculates the lateral"
Disp  "area of a right cylinder."
Input  "Radius of base = ?",r
Input  "Height of cylinder = ?",h
setMode("Exact/Approx","EXACT")
2*π*r ➜c
Output  62,10,"Exact Area = "
Output  62,90,c*h
Output  62,150,"units^2"
setMode("Display Digits","FIX 2")
setMode("Exact/Approx","APPROXIMATE")
Output  85,10,"Approx Area ="
Output  85,90,c*h
Output  85,150,"units^2"
EndPrgm
```



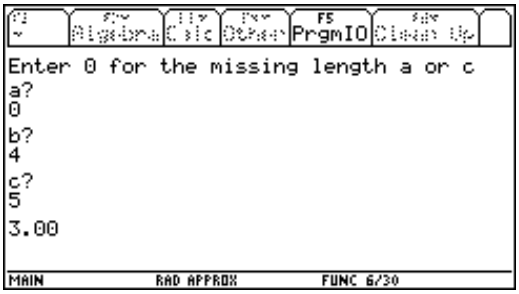
If - Then Statements:

- These statements are often used when you need to perform different operations depending on specific criteria.
- Type **If**, followed by a condition that you want meet, finish that line with **Then**.
- You may to continue programming on the next lines. These lines will only be executed when the **If** part of the above statement is true.
- Finish this statement with **EndIf** on its own line.
- If the **If** part of the statement is false, the program will ignore all lines of programming between the **If** and the **EndIf**.

Example:
```
:If a=5 Then
:Disp "Still With Us?"
:EndIf
```
(*Still With Us* will only be displayed if *a* has a value of 5.)

Example Program: Pythagorean Theorem

```
pythag()
Prgm
ClrIO
Local a,b,c
Disp "Enter 0 for the missing length a or c"
Prompt a
Prompt b
Prompt c
If a=0 Then
√(c^2-b^2)➞a
Disp a
EndIf
If c=0 Then
√(a^2+b^2)➞c
Disp c
EndIf
EndPrgm
```



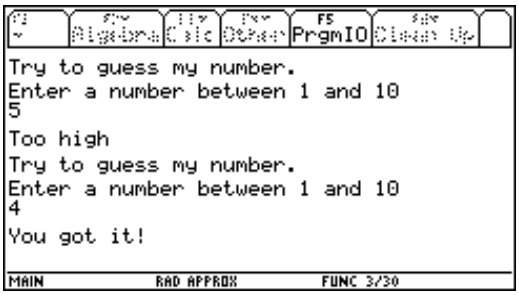
Labels and the Goto Command:

- **Lbl** and **Goto** can be used to redirect the operation of a program.
- **Goto** redirects the program to a specific label.
- **Lbl** marks an specific location in a program.
- You can place **Lbl** followed by a variable some where in the program. Some where else in the program, you can place **Goto** followed by the same variable and the program will redirect its current operation to the line of programming below the **Lbl**.

Example: **:**Lbl a (This program will print T3 over and over again.)
:Disp "T3"
:Goto a

Example Program: Guess My Number

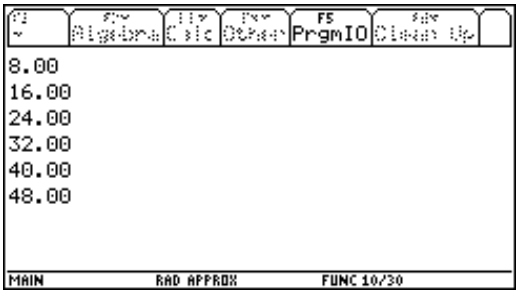
```
guess()
Prgm
ClrIO
Local  a,b,c
rand(10)➜a
Lbl  c
Disp  "Try to guess my number."
Input  "Enter a number between 1 and 10",b
If b=a Then
Goto  d
EndIf
If b>a Then
Disp  "Too high"
Goto  c
EndIf
If b<a Then
Disp  "Too low"
Goto  c
EndIf
Lbl  d
Disp  "You got it!"
EndPrgm
```



For - Next Loops:

- For - Next commands repeat a certain set of programming lines and specific number of times.
- Type **For**, a variable name for the loop, a comma, the beginning count, a comma, the ending count, a comma, and the amount the count steps "up" every time the loop repeats.
- The value of the variable for the loop "increases" by the step amount every time the loop repeats.
- You may continue programming on the next lines. These lines will be repeated until the value of the loop variable "exceeds" the ending count.
- You must finish the loop with an **EndFor** statement.

Example:
```
:For n,1,6,1
:Disp 8*n
:EndFor
```
(This program will display 6 terms of a linear sequence.)



Example Program: Sum of Interior Angles

```
anglsum()
Prgm
ClrIO
Local n,k
Disp "Will display the sum of interior angles"
Disp "for polygons with 3 through n sides."
Input "Enter n",n
For k,3,n,1
Disp k
Disp (k-2)*180
EndFor
EndPrgm
```