



#### Junior Coding for Good, Badge 1 requirements:

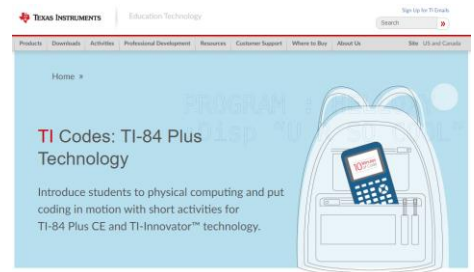
1. Create algorithms for a computer that follow a sequence.
2. Use loops to improve your algorithm.
3. Keep your code interesting with conditionals.
4. Create your own set of commands that use conditionals.
5. Learn about women in computer science.

Be sure to review the “Junior Coding for Good” guide for the badge requirements and badge steps provided by the Girl Scouts and your leader. It also includes relevant vocabulary and interesting background information to spark your interest in coding. This lesson will allow you to earn **Badge 1: Coding Basics** using your TI-84 Plus family graphing calculator.

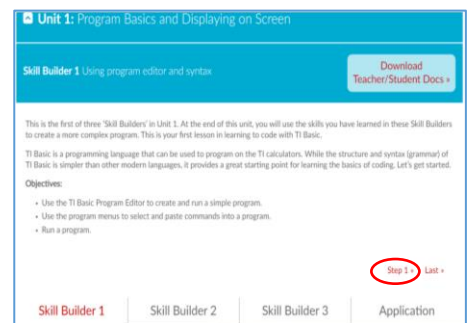
#### Introduction

The programming language **BASIC** (stands for **B**eginner’s **A**ll-purpose **S**ymbolic **I**nstruction **C**ode) was developed in the 1960s as an easy system for teaching computer programming. TI-Basic is similar to other flavors of BASIC, but you must select the programming words and commands from the onboard menus, as you will soon see.

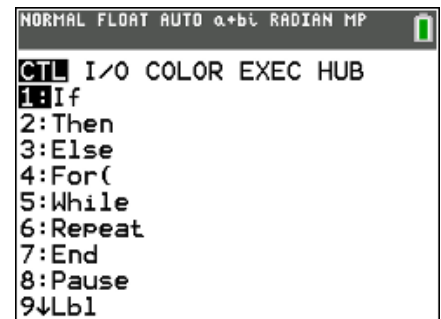
1. For an introduction to programming on a TI-84 Plus family graphing calculator, see the TI Codes lessons at [education.ti.com](http://education.ti.com) > **Activities** > **TI Codes** > **TI-Basic**. Units 1 through 4 should be enough to get started, including coverage of algorithms and conditionals as the Badge requirements state (1–3 above). After completing those lessons, you will then have a basic understanding and can return to this lesson for the **Junior Coding, Badge 1** project. Go to [10 Minutes of Code \(for TI-84 Plus technology\)](#).



**How to navigate TI Codes:** Be sure to notice that each unit has three Skill Builders (SB) and one Application activity, as you can see here. You will navigate through each Skill Builder by clicking on the “Step 1” in the right bottom corner, which will take you through the content. This screenshot shows you are currently in Skill Builder 1 > Step 1. After you complete all the steps in SB 1, you will then move to SB 2, and so on, until you complete all of Unit 1. You will then move on to Units 2, 3, 4 ... and do the same thing.



2. After you have completed the TI Codes Units 1–4, you know about:
  - The programming process (planning, coding, testing and debugging)
  - Using the TI-84 Program Editor and its menus
  - Running a program
  - Some important pieces of TI-Basic code: Input, Disp, variables and assignment statements, **if...Then...Else** structures and loops, like **For** and **While** (some of these statements are shown to the right)





# Girl Scouts: Coding for Good

JUNIOR LEVEL: BADGE 1

## TI-84 PLUS / TI-84 PLUS CE TECHNOLOGY

## STUDENT ACTIVITY

3. Your Badge 1 project will be a **Number Guessing Game**. The calculator will select a random number between one and 100, and the player must guess the number. And:
- The game continues until the player guesses the number
  - If the guess is wrong the computer will tell you whether to guess HIGHER or LOWER to get closer to the number
  - After the player guesses the number, the computer will report the number of guesses that the player took

A sample run of the program is shown to the right. The player entered the numbers 45, 34 and 12. The program responded with HIGHER and LOWER. Think about the plan. You will begin writing the program next.

```

NORMAL FLOAT AUTO a+b| RADIAN MP
----- Done -----
PrgmHILOW
GUESS?45
LOWER
GUESS?34
LOWER
GUESS?12
LOWER
GUESS?

```

4. Our plan is:
- Create some variables
  - Use a **While** loop to keep guessing as long as we have not guessed the number
  - In the loop
    - Count the guesses
    - **Input** a guess
    - Check to see whether the guess is too high or too low
  - After the loop
    - Display a congratulations message
    - Display the number of guesses



5. Start a new program: Press **[prgm]**, and select the **NEW** menu at the top of the screen. Name the program **HILOW** and press **[enter]**. You are now using the Program Editor. As you learned in the TI Codes lessons, this Editor changes many of the keys on the keypad.

```

NORMAL FLOAT AUTO a+b| RADIAN MP
EDIT MENU: [alpha] [f5]
PROGRAM:HILOW
:

```

6. Start by making a random number from one to 100 using `randInt()` found on **[math] > Prob**. You will see a dialog like the one on the right. Enter 1 for lower:, 100 for upper: and leave n: blank. Highlight Paste and press **[enter]**.

```

NORMAL FLOAT AUTO a+b| RADIAN MP
----- randInt -----
lower:1
upper:100
n:
Paste

```



### TI-84 PLUS / TI-84 PLUS CE TECHNOLOGY

### STUDENT ACTIVITY

- Complete the statement by pressing **[sto]** (→) and the letter **N**. This variable is the computer's Number. After the **randInt(1, 100)** function, press the **sto** key (it displays →) and type the variable **N** (press **[alpha]** N). Press **[enter]** to go to the next line.

The other two variables, **G** for **G**uess and **C** for **C**ount, are given the value 0, as shown. Remember to use the **sto** key and then the correct letters. Your screen should match the one here.

```
NORMAL FLOAT AUTO a+bi RADIAN MP
EDIT MENU: [alpha][f5]
PROGRAM:HILOW
:randInt(1,100)→N
:0→G
:0→C
:
:
:
:
:
:
```

- Now make a **While...End** loop to continue to enter guesses until the guess equals the computer's number (as long as *G does not equal N*).

Press **[prgm]** **Ctl** to find the **While** statement, type the condition, **G ≠ N**, then press **[enter]** a few times and add the **End** statement from the same menu.

The does not equal symbol (≠) is on the **[test]** menu: press **[2<sup>nd</sup>]** **[math]** to see the test menu.

```
NORMAL FLOAT AUTO a+bi RADIAN MP
EDIT MENU: [alpha][f5]
PROGRAM:HILOW
:randInt(1,100)→N
:0→G
:0→C
:While G≠N
:
:
:
:End
:
```

- Inside the **While** loop:
  - Add 1 to the guess-counter variable **C**
  - Add an **Input** statement to accept the player's guess (**G**) by getting the word **Input** from the **[prgm]** **I/O** menu and add the variable **G**
  - If guess **G** is *less than* (<) the number (**N**) — display "HIGHER"
  - If guess **G** is *greater than* (>) the number (**N**) — display "LOWER"

Note: Be sure to write *two If...* statements even though *only one is shown here*.

```
NORMAL FLOAT AUTO a+bi RADIAN MP
EDIT MENU: [alpha][f5]
PROGRAM:HILOW
:randInt(1,100)→N
:0→G
:0→C
:While G≠N
:C+1→C
:Input "GUESS?",G
:If G<N
:Disp "HIGHER"
:
:
:
:
:
:
```

- After the **End** of the **While** loop:
  - Display "YOU GOT IT."
  - Display the number of guesses (**C**) that the player took to guess the number

```
NORMAL FLOAT AUTO a+bi RADIAN MP
EDIT MENU: [alpha][f5]
PROGRAM:HILOW
:Input "GUESS?",G
:If G<N
:Disp "HIGHER"
:If G>N
:Disp "LOWER"
:End
:Disp "YOU GOT IT."
:Disp C," GUESSES"
:
:
```



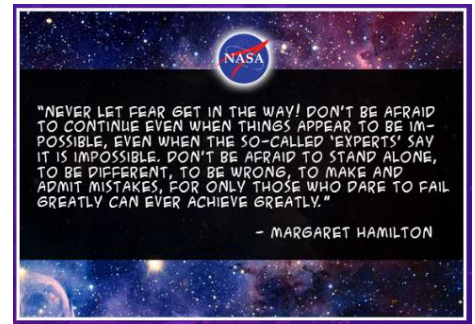
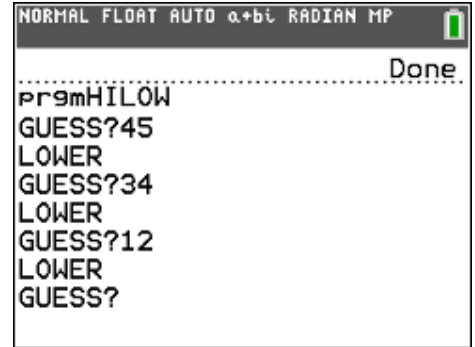
11. Now test your program, and make changes as needed.

To run the program:

Press [quit] ([2nd] [MODE]) to return to the HOME screen.

- Press [PRGM].
- Under the EXEC ('execute') menu, select your program.
- Press [ENTER] to paste the program name on the HOME screen.
- Press [ENTER] again to begin the run.

12. For the final part of the badge requirement, remember to research "women in computer science," such as Margaret Hamilton, who helped humans land on the moon. Use the internet for your searching.



### Congratulations!

You have completed the requirements for earning your **Junior Coding for Good, Badge 1: Coding Basics**. Now that you have completed this requirement, you are challenged to *give service* by *sharing* what you have learned about coding with others. Refer to the "Coding for Good" Girl Scout guide for suggestions on how to do so.

### What's next? (Optional extensions)

Ready to try some additional practices with your new skills?

1. Try these to edit your game from above:
  - Before the game begins, let the player enter the upper number (100 in the lessons) to play with. **Input** T and use **randInt(1,T)** instead of **randInt(1,100)**.
  - **"Trickster"**: Want to play a trick on the player? After each guess, have the program pick a new random number. Do you think the player will ever get the number?
  - **"Warm or Cold"**: Guess the number from one to 100 again, but this time, the computer responds with **"Freezing-Cold-Warm-Hot-Boiling"** depending on how close the guess is to the answer. You can decide just what each of those words means in your code, but do not tell the player.



### TI-84 PLUS / TI-84 PLUS CE TECHNOLOGY

### STUDENT ACTIVITY

2. **Whitney Wolfe Herd** is an American entrepreneur. She is founder and CEO of Bumble, a popular social and dating app. Bumble launched in 2014 and now the company is now valued at more than *1 billion* dollars.
  - a. How much is 1 billion (1,000,000,000)? How long is 1 billion seconds in days and years? Use the calculator. How long will it take to spend 1 billion dollars? Write a program to let the user enter the amount she spends in one day and display 1) how many days and 2) how many years it will take to spend 1 billion dollars.
  
3. Did you know you can create other “graphical” games on your calculator? Take a peek at the projects “SNAKE” or the “MAZE” in the [Beyond Basics](#) projects to try your hand at a few simple games.

*Warning: These are more complex, but would be an excellent challenge!*