Ambassador Coding for Good, Badge 1 performance:
Song lyrics; sound using the TI-Innovator™ Hub

1. Learn about functions through song lyrics.
2. Learn about loops through song patterns.
3. Write an algorithm duet.
4. Code a performance routine.
5. Share your coded routine with others.

Be sure to review the "Ambassador Coding for Good" guide for the badge requirements and badge steps provided by your leader and the Girl Scouts. 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 > Activities > TI Codes > TI-Basic**. Units 1 through 4 should be enough to get started. Then return to this lesson for the **Ambassador 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.
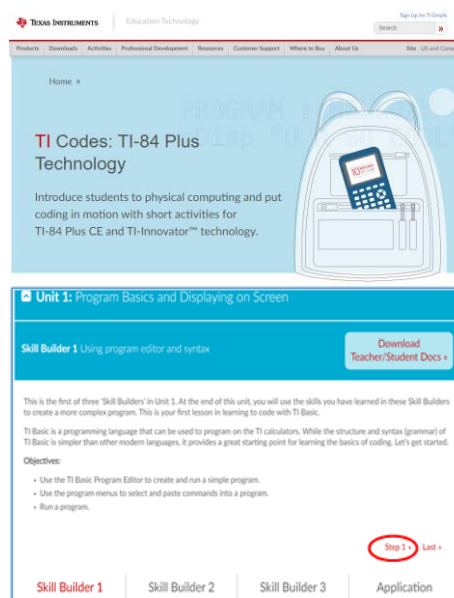
2. If you are using a TI-84 Plus CE graphing calculator and plan to use a TI-Innovator™ Hub for sound (optional) then also check out the TI-Innovator™ lessons at 10 Minutes of Code (for TI-84 Plus and TI-Innovator™ technology).

3. After you have completed the TI Codes units, you know about:
   - The programming process (planning, coding, testing and debugging)
   - Using the TI-84 Plus Program Editor and its menus

- Running a program
- Using the TI-Innovator™ Hub (optional)
- 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)

4. The basic structure of songs is the verse, chorus and refrain structure. When a portion of a song is repeated (like the chorus or refrain), then it is a perfect opportunity to incorporate a subroutine in computer programming. Your main program (**PERFORM1** in the image to the right) can "call" another program (**prgmPREFRAIN** or **prgmPCHORUS**) to display the refrain or chorus at the proper times. This sample program uses the lyrics from the Beatles song "She Loves You."*

   Your program will be written in all UPPERCASE letters, except for the programming keywords which are picked from the menus. The program displayed here was written with the free [TI Connect™ CE software application](#).

5. To start a program, press **[prgm]**, arrow right to **NEW**, press **[enter]**, type the name of the program (we use **PERFORM1**), and press **[enter]** again.

   This example uses three separate programs.

6. The main program **PERFORM1**, and the two subroutines **PREFRAIN** and **PCHORUS**, are listed as three separate programs on the **[prgm]** menu to the right (along with other, unrelated programs).

   You must write *separate* programs. We are using **PERFORM1**, **PCHORUS** and **PREFRAIN**. They all start with the letter **P** so that they appear in the same area on the alphabetical list of programs shown to the right.

7. In your programs, use **Disp** statements (found on **[prgm] I/O**) to display each line of the song, one at a time. Be sure to put the line in quotes.

   To type a long string of characters you can turn on alpha-lock (2ⁿᵈ [alpha]). Be sure to turn it off later by pressing [alpha] before pressing other keys like the up/down arrow keys because alpha-up and alpha-down are PgUp and PgDn keys. Pay attention to the cursor.

8. In your main program, you can "call" or use the subroutine by entering its name on a line of its own. Note the statement containing **prgmPREFRAIN** in the image to the right. When the processor encounters this statement, it will run the code in that program (and bookmark the location in this program where it left off).

   The statement **:"[Verse 2]** *is a comment and is not displayed when the program runs. This is a good way to add personal notes to your programs.*

   ### EDITING TIPS
   To insert lines in your code:
   - *Place the cursor at the insertion point (the beginning of a line)*
   - *Press **[2ⁿᵈ] [del]** for the [insert] mode*
   - *Press **[enter]***
   - *Press **[uparrow]** to place the cursor on the new blank line*

   *If you accidentally press the **[clear]** key (which erases the entire line of code!) you can restore the lost line by pressing **[alpha][graph]** (the F5 key) and choose "Undo clear" from the menu. There are other useful tools on that menu as well.*

9. To add the subroutine program name into your main program (to "call" the subroutine) on a blank line in the main program, press **[prgm],** arrow to the **EXEC** list, and locate the name of your subroutine in the list of all your programs. The subroutine must be created *before* it can be found on this list. It does not have to be complete or even correct, but the program name must exist.

   *Note that the screen to the right is showing the **EXEC** menu.*

10. Your subroutine displays the refrain or chorus (shown) and then the returns control to the *bookmarked* spot in your main program to continue the rest of the song.

    Note the use of the **Wait** *statement to control the display of the lyrics.* **Wait** *is found on* **[prgm] CTL**. **Wait** *is followed by the number of seconds to wait.* **Wait 1** *means "wait one second."*

    EDITING TIP
    Use the **[F5]** *key (***[alpha][graph]***) to copy and paste a line of code.*

```
NORMAL FLOAT AUTO a+bi RADIAN MP
EDIT MENU:[alpha][f5]
PROGRAM:PCHORUS
:Disp "She loves you, yeah
, yeah, yeah"
:Disp "She loves you, yeah
, yeah, yeah"
:Disp "She loves you, yeah
, yeah, yeah, yeah"
:Wait 1
:
```

11. Your main program (**PERFORM1**) displays the verses with calls to the **prgmPCHORUS** and **prgmPREFRAIN** subroutines at the proper times.

```
NORMAL FLOAT AUTO a+bi RADIAN MP
EDIT MENU:[alpha][f5]
PROGRAM:PERFORM1
:Disp "It's you she's thin
king of "
:Disp "And she told me wha
t to say"
:prgmPREFRAIN
:"[Verse 2]
:Disp "She said you hurt h
er so"
:■isp "She almost lost her
```

12. For an audio challenge using a TI-84 Plus CE and TI-Innovator™ Hub, write a song and include the **SOUND** commands with the proper notes (or frequencies) and timings (**Wait**). Perhaps you can even sync the sounds with the lyrics to make a karaoke machine!

    The **Send** *and* **Wait** *statements in this screen are incomplete, and the actual code is more involved. For more information, see the TI-Innovator™ lessons at* [10 Minutes of Code](#).

    If you are sharing on social media, remember to tag it @**TICalculators**.

```
NORMAL FLOAT AUTO a+bi RADIAN MP
EDIT MENU:[alpha][f5]
PROGRAM:PCHORUS
:Disp "She loves you, yeah
, yeah, yeah"
:Send("SET SOUND
:
:Wait
:
:
:Disp "She loves you, yeah
, yeah, yeah"
```

Congratulations!

You have completed the requirements for earning your **Ambassador 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 other suggestions on how to do so.

What's next? (Optional extensions)

Ready to try some additional challenges with your new skills?

1. The **Beyond Basics** section of TI Codes provides several interesting programming projects. In keeping with the theme of this badge, try the **Piano** project or the **Musical Scale** project. Both make use of the TI-Innovator™ Hub for making sounds.

2. Make your programs more efficient using subroutines/subprograms as described in this lesson to play the different sections of the songs.