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

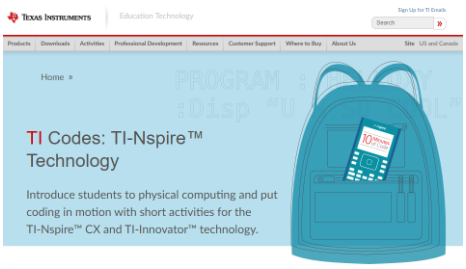
1. Learn about functions through song lyrics.
2. Learn about loops through song patterns.
3. Write an algorithm duet.
4. Code a performance routine.
5. Share your coded routine with others.

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-Nspire™ CX or TI-Nspire™ CX II graphing calculator.

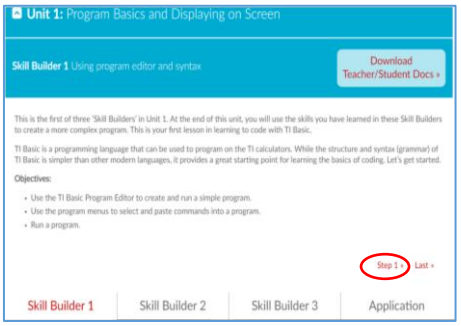
Introduction

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.

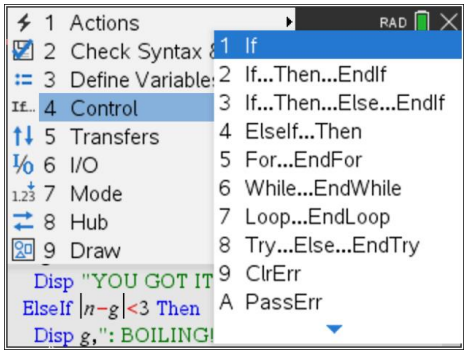
1. For an introduction to programming on the TI-Nspire™ CX family graphing calculators, see the TI Codes lessons at **education.ti.com > Activities > TI Codes > TI-Basic**. Units 1 through 4 should be enough to get started. Then return to this lesson for the **Ambassador Coding, Badge 1** project. Go to TI Codes (for TI-Nspire™ 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.

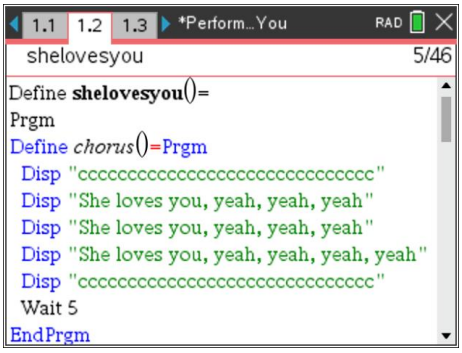


2. After you have completed the TI Codes units, you know about:
   - The programming process (planning, coding, testing and debugging)
   - Using the TI-Nspire™ Program Editor and its menus
   - Running a program
   - Using the TI-Innovator™ Hub (optional)
   - Some important pieces of TI-Basic code: 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)



3. The sample program for this lesson uses the lyrics of the song “She Loves You” by John Lennon and Paul McCartney (of the Beatles). *Do not begin coding yet.*

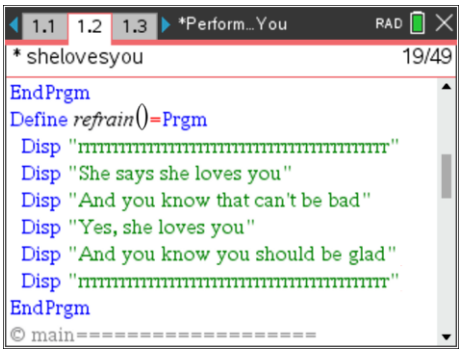
   The basic structure of songs is the verse, chorus and refrain structure. When a portion of a song is repeated (like the chorus or refrain), then it is a perfect opportunity to incorporate a subroutine in computer programming. The main program in the image to the right *contains* and “calls” the subroutine **chorus()** to *repeatedly* display the refrain or chorus at the proper times *without having to rewrite the lyrics.*



4. This lesson explains how to write and use subroutines on a **TI-Nspire™ CX family graphing calculator.**

   Notice that in this image, the main program **shelovesyou()** contains the subroutine **chorus()** along with another subroutine called **refrain().**

   To write a program with subroutines, follow the next steps.

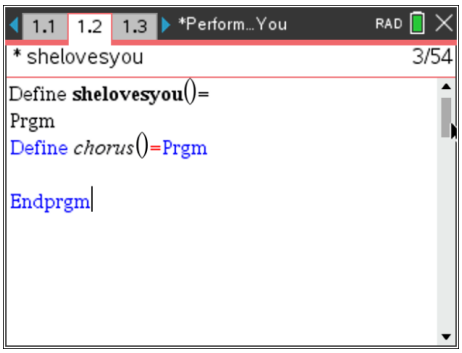


5. Start the main program by making a new document (**[home] New**). Select **Add Program Editor** from the next apps menu. We named our program **shelovesyou**.

   The subroutines are created first:

   a. Press **[menu] > Define variables**, and select the word **Define** from the menu.
   b. Right after the word **Define**, write the name of the subprogram, **chorus( )=** including the parentheses and the equals sign. Do not press **[enter]** yet.
   c. Press **[menu]> Define Variables** and select the structure **Prgm…EndPrgm**.

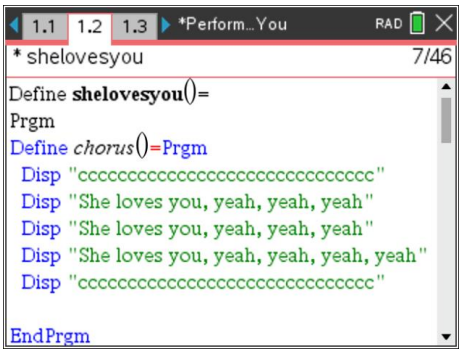
   Tip: When defining a subroutine within a program it is important that the keyword **Prgm** *be on the* ***same line*** *as* **Define** *chorus()=***Prgm**. *An error occurs if it appears on the next line as it does for the main program.*



6. In the chorus subroutine write the **Disp** statements (found on **[menu] I/O**) that show the lines of the chorus portion of the song now. Or you can save that for later.

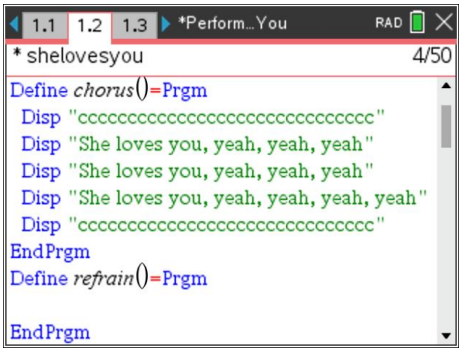
   The two lines that display "ccccccccccccccccccccccc" are for programmer information only, indicating that this section of text is the chorus. These can be removed (or commented) later, but for coding purposes it makes the program execution clear.

   Tip: To speed up the typing, you can select, copy and paste code just like on a computer: Use **shift-arrows** *to select,* **ctrl-C** *to copy and* **ctrl-V** *to paste.*



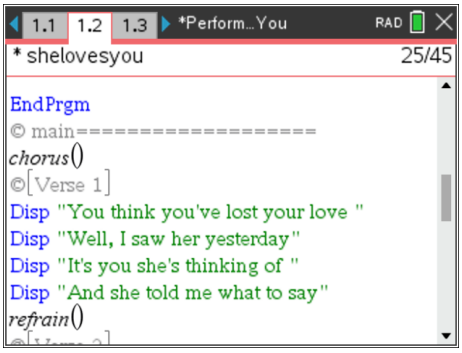
7. This sample song also contains a refrain. Below the chorus subroutine (right after the **EndPrgm** of the chorus section), start another subprogram called **refrain()**.

   In this subroutine, **Displ**ay the lyrics of the refrain (not shown).



8. After the **refrain()** subroutine (and any other subroutines your song may use), you can begin writing the *main* program consisting of the verses. This sample song begins with the **chorus()**, then verse 1 (note the © comment), then the **refrain()**.

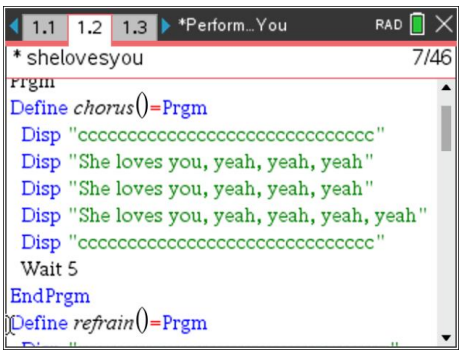
   Comments (the lines beginning with the © symbol) are used a lot in programming to keep the code organized and clear to the reader of the code. These comments are not displayed when the program is run.



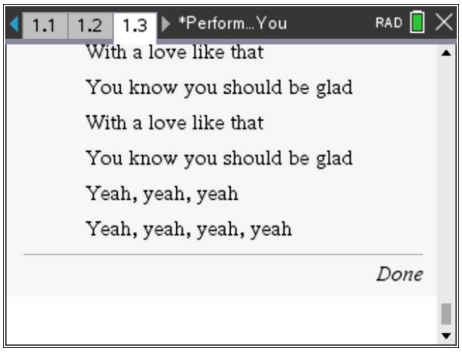
9. You may want to add some **Wait** statements to your program in appropriate places to control the display so that the lyrics don't scroll by too fast.

   Wait is found on **[menu] > Control** (at the bottom) *and* on **[menu] > Hub**. It needs a number of seconds to wait after the command. **Wait 5** means "wait 5 seconds."

   *Share your project on Instagram, or other social media, and be sure to tag it @**TICalculators***



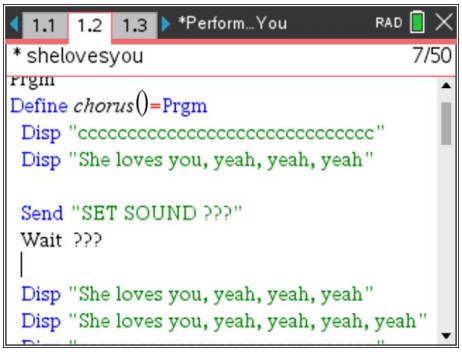
Congratulations!



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 suggestions on how to do so.

What's next? (Optional extensions)

Ready to try some additional practices with your new skills?

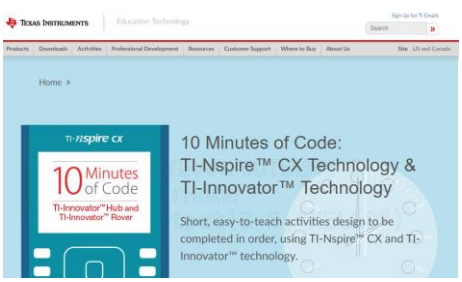


1. Want to try and include some audio? We can use the speaker on the TI-Innovator™ Hub, write a song that includes using the **Send "SET SOUND…"** command with the proper notes (or frequencies) and timings (**Wait**).

   Perhaps you can even sync the sounds with the lyrics to make a karaoke machine!

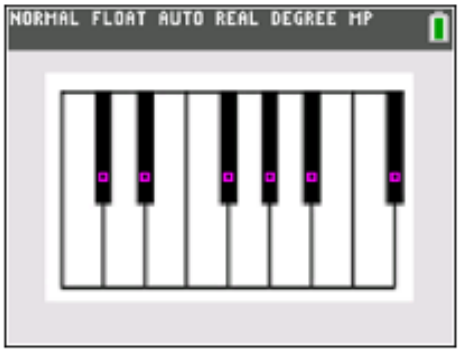
   The **Send** and **Wait** statements (shown) are incomplete, and the code for producing notes is more involved than this.

   If you plan to use the TI-Innovator™ Hub for sound, then also check out the TI-Innovator™ lessons at 10 Minutes of Code.



   If you are using a TI-Nspire™ CX II graphing calculator, consider using the graphics canvas and **DrawText** statements to display your lyrics instead of the **Disp** statements used in this lesson. See Unit 6 of the TI Codes materials for more information on the **Draw** commands.

2. The **Beyond Basics** section of **TI Codes** for the **TI-84 Plus family graphing calculator** 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. You should be able to adapt the projects from the TI-84 Plus family graphing calculator to the TI-Nspire™ CX II graphing calculator, since the TI-Basic languages are the same but the implementation (keywords, punctuation and syntax) are slightly different.



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