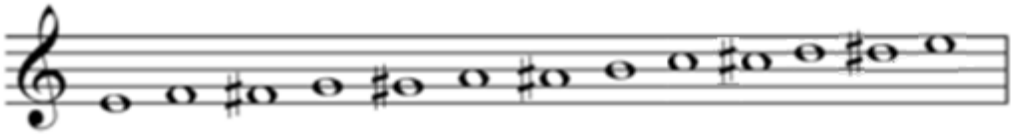
## Play a Musical Scale

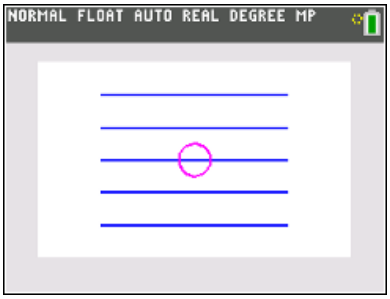
After completing this project, you'll have a program that draws a musical note on a staff that corresponds to the tone played. The brightness readings from your TI-Innovator™ brightness sensor will determine the note to play.

**Objectives:**

- draw a music staff
- use the brightness sensor on the TI-Innovator™ Hub
- use IF statements to make decisions
- use a WHILE statement to repeat code
- use lists to store patterns

### Musical Scale Project Overview






In this project, you'll use the **Line()** command to draw a music staff. You'll use the **Circle()** feature to draw music notes. You'll use the reading from the TI-Innovator Hub brightness sensor to determine which note to draw and which frequency to play.

**Teacher Tip:**
Make sure the calculator is at least 50% charged in order to power the TI-Innovator Hub.
Make sure students plug the 'B' end of the cable into the TI-Innovator hub port.

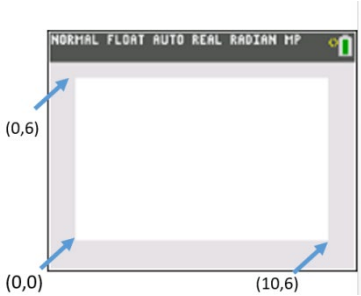
1. Create a new program named SCALES

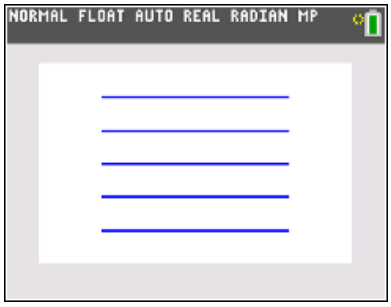
   Change the orientation of the screen so (0, 0) is in the lower left and the value increase as you go up the screen.

   Put ClrDraw as your first line of code ( 2nd prgm )
   This will clear any old drawing off the screen.

   Use the [vars] menu to set the window using the following 4 statements.
   :0→Xmin
   :10 →Xmax
   :0 →Ymin
   :6 →Ymin

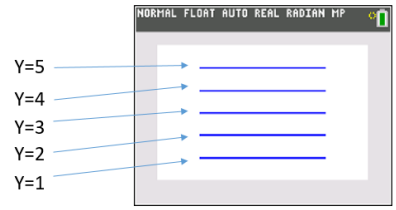
   To draw a line, the basic command is: Line(X1, Y1, X2, Y2)
   Write a For loop to draw the 5 lines of the musical staff on the board.



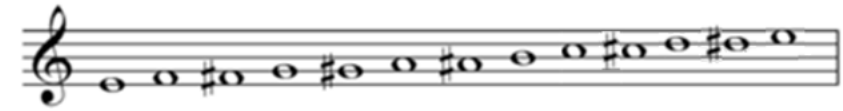
2. Does your code look something similar to

```
:ClrDraw
:0→Xmin
:10 →Xmax
:0 →Ymin
:6 →Ymin
:For(Y,1,5)
:Line(2,Y,8,Y)
:End
```

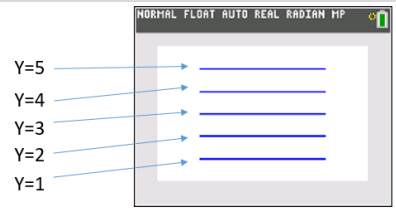



**Teacher Tip:**
Xmin, Xmax, Ymin, Ymax are located in Vars – Windows

3. This particular scale starts at E.
   Each note has a y-value

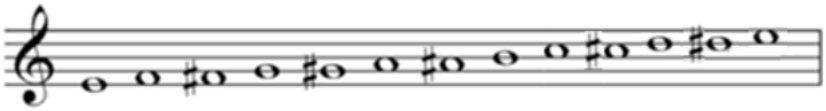


Y-Value: 1 1.5 1.5 2 2 2.5 2.5 3 3.5 3.5 4 4 4.5

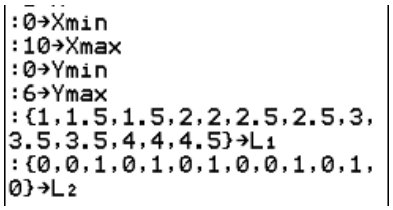



Store each value in a list so you can use the heights later.
:{1, 1.5, 1.5, 2, 2, 2.5, 2.5, 3, 3.5, 3.5 , 4 , 4 , 4.5} →L1

4. You will also use a list to determine IF you need to draw a sharp.
   You will use 0 to represent a natural note and a 1 to represent a sharp note.



0 0 1 0 1 0 1 0 0 1 0 1 0




Store these values in a second list
:{0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0} →L2

```
PROGRAM:SCALES
:10→Xmax
:0→Ymin
:6→Ymax
:{1,1.5,1.5,2,2,2.5,2.5,3,
3.5,3.5,4,4,4.5}→L1
:{0,0,1,0,1,0,1,0,0,1,0,1,
0}→L2
:0→K
:0→T
```

5. Create a variable **K** to store keypress values. If the user pressed the clear key (45), the While loop repeating your code will terminate.

   You need to create a variable **T** to store the last note played. You'll change the frequency of the note only if it is a new frequency.

   Store 0 as the first value for K.

   Store 0 as the first value for T.

6. A.) Write a loop that repeats while K doesn't equal the value of the clear key (45).

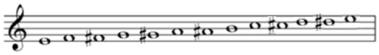
   B.) Inside the loop, read the value of the brightness sensor.

   C.) Store value as variable B.

   D.) You need to scale the brightness value B. For this example, divide B by 8 then store only the integer portion back in B. (Based on your lighting you might have to adjust this scaled value later.)

7. Does your code match the cold on the right?

```
:While K≠45
:getKey→K
:
:Send("READ BRIGHTNESS ")
:Get(B)
:
:iPart(B/8)→B
```



There are 13 values stored in L1 and 13 in L2.

If the brightness level is between 1 and 13 and it isn't the note already playing, you should
1. Clear the old picture
2. Draw the scale
3. Draw the new note
4. Draw the sharp, #, if it is a sharp note
5. Play the new frequency.

Take a moment to brainstorm how to accomplish the 5 tasks above.

**Teacher Tip:**
Based on the lighting, students might need to divide B by a different value.
Many cell phones have flashlight apps that can be used to generate different brightness levels.

8. Can you code the following:
   If the brightness level is between 1 and 13 and it isn't the note already playing, you should
   1. Clear the old picture
   2. Draw the scale.

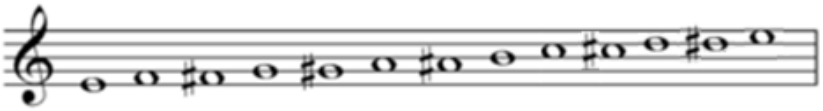
9. Does your code match the code on the right?

```
:iPart(B/8)→B
:If B>0 and B≤14 and T≠B
:Then
:ClrDraw
:
:For(Y,1,5)
:Line(2,Y,8,Y)
:End
:
```

Remember L1 stores the y-values for all the notes.
Draw the new note as a circle with a center at (5, L1(B)) and a radius of 0.5.

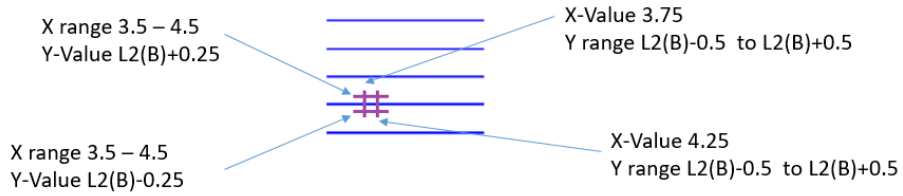
```
:Circle(5,L1(B),.5,MAGENTA
)
```

10. Remember L2 has 1s and 0s stored representing the sharp note.



0 0 1 0 1 0 1 0 0 1 0 1 0

If the value in L2(B) is a 1, draw a sharp sign

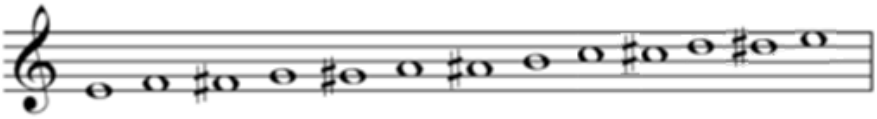



Write the IF statement and the lines of code to draw the sharp symbol.
The sharp should only be drawn IF L2(B) = 1.

11. Does your code match the code on the right?

```
:If L2(B)=1
:Then
:Line(3.5,L1(B)-.25,4.5,L1
(B)-.25)
:Line(3.5,L1(B)+.25,4.5,L1
(B)+.25)
:Line(3.75,L1(B)+0.5,3.75,
L1(B)-0.5)
:Line(4.25,L1(B)+0.5,4.25,
```

12.



You may recall from [10 Minutes of Code with TI-Innovator](): Unit 2, SB 3 that "Musical notes are determined by the frequency of a vibrating object such as a speaker, drum head, or as in a guitar or a piano. The notes of the musical scale have a special mathematical relationship. If a note has a frequency F, then the very next note has a frequency F * $2^{1/12}$. "

Multiplying a note's frequency by $2^{1/12}$ gives the next note in the scale.

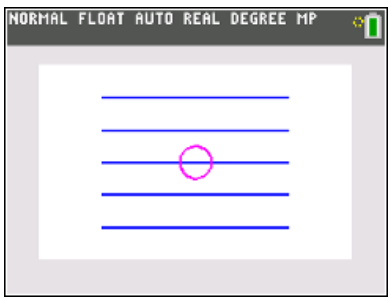
```
:Send("SET SOUND eval(164.
82*2^((B-1)/12))
```

The scale starts at E, so 164.82Hz will be the base frequency. In order to get E, the exponent needs to be 0. Subtract one from the B value for the exponent

13. Store B as the last brightness value T.
    :B→T

End the While Loop
    :End

Execute your code.




Remember the line iPart(B/8)→B? Try dividing by values other than 8. Optimize this value for the brightness level in your setting.

**Teacher Tip:** Sample Code

```
NORMAL FLOAT AUTO REAL DEGREE MP
EDIT MENU: [alpha] [f5]
PROGRAM:SCALES
:ClrDraw
:0→T
:1→K
:0→Xmin
:10→Xmax
:0→Ymin
:6→Ymax
:{1,1.5,1.5,2,2,2.5,2.5,3,
3.5,3.5,4,4,4.5}→L1
:{0,0,1,0,1,0,1,0,0,1,0,1,
0}→L2
:While K≠45
:getKey→K
:For(A,1,5)
:Line(2,A,8,A)
:End
:Send("READ BRIGHTNESS ")
:Get(B)
:
:iPart(B/8)→B
:If B>0 and B≤14 and T≠B
:Then
:ClrDraw
:Circle(5,L1(B),.5)
:If L2(B)=1
:Then
:Line(3.5,L1(B)-.25,4.5,L1
(B)-.25)
:Line(3.5,L1(B)+.25,4.5,L1
(B)+.25)
:Line(3.75,L1(B)+0.5,3.75,
L1(B)-0.5)
:Line(4.25,L1(B)+0.5,4.25,
L1(B)-0.5)
:End
:End
:
:B→T
:If B>0 and B≤14
:Then
:Send("SET SOUND eval(164.
82*2^((B-1)/12))
:End
:Wait .2
:End
```