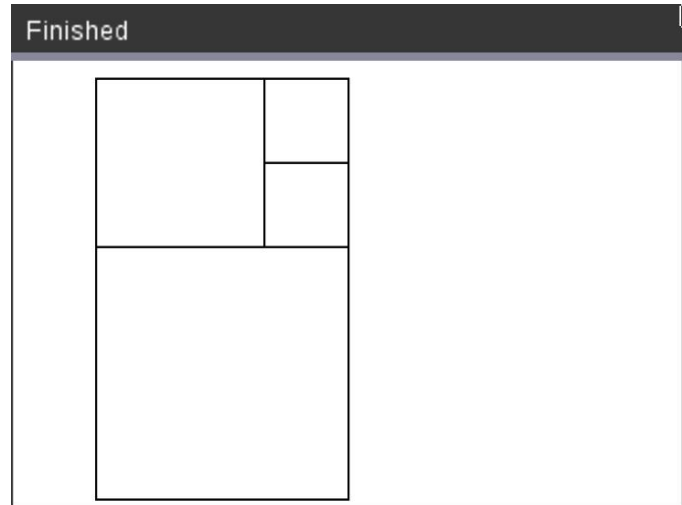
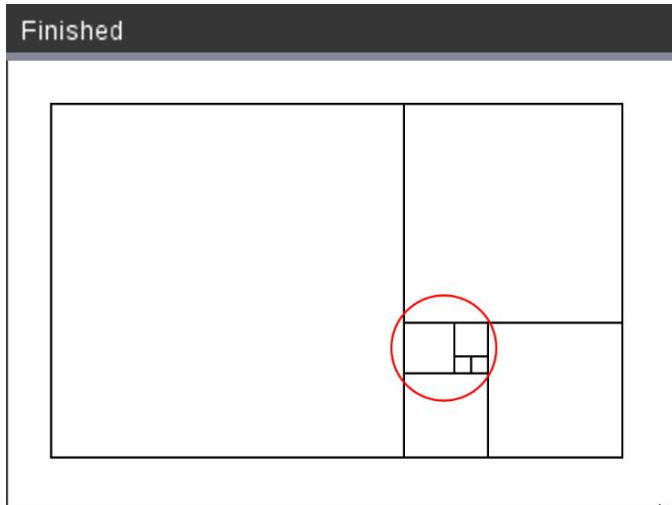
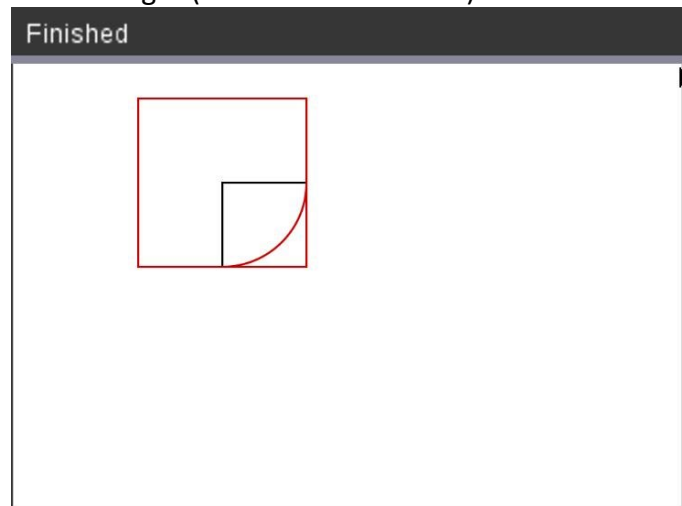
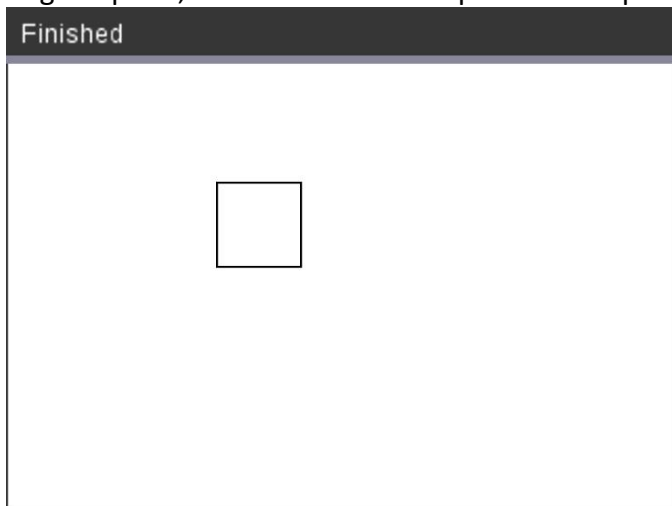


## How to draw the “Fibo” Heart

The Fibonacci heart is drawn using both the **fill\_arc** and **draw\_arc** commands in the draw menu on the TI-Nspire™ CX-II. The arcs are drawn in successive squares of the Fibonacci sequence. There was one section of the well know arrangement that I focused on. This section below I then rotated and created it as large as possible on the 318 x 212 pixel screen. The other half was just a reflection of this image.

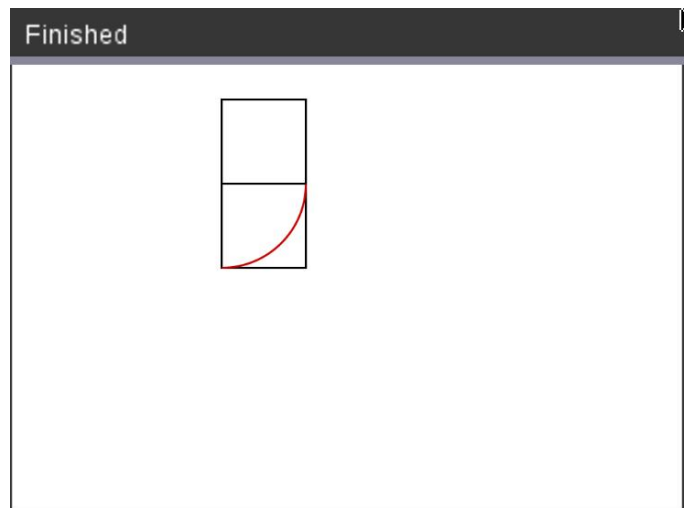
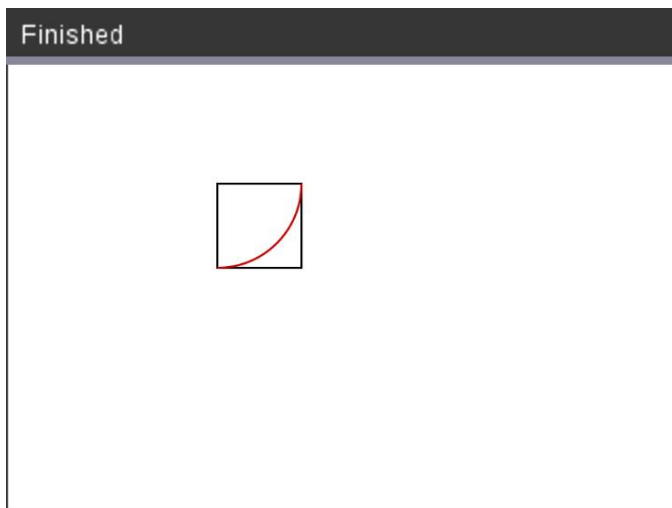


After this was done, it was time to create the arcs (or quarter circles). So for the first square, this required a larger square, where the smaller square was a quarter of the larger (as seen in red below).

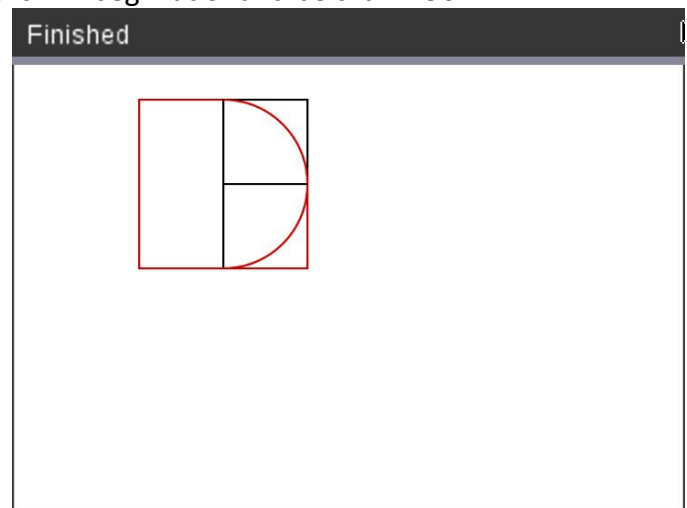
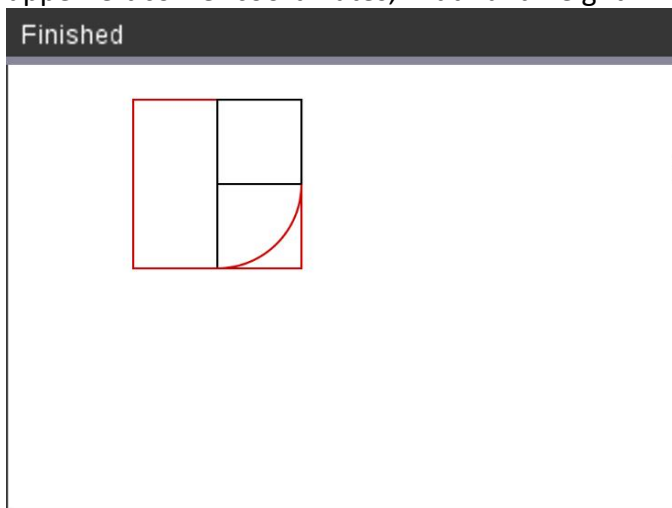


The black square has a draw command of **draw\_rect(119,48,40,40)** [**draw(x,y,width, height)** (x,y) is the upper left corner of the rectangle] In order to draw the  $\frac{1}{4}$  arc (or  $90^\circ$  arc) I then created the larger square, in red, which was **draw\_rect(79,8,80,80)** or **draw\_rect(119-40,48-40,80,80)**. I use the **draw\_rect(** command a lot to see where it is that the arc will be drawn. So for this red square we want to start our arc at  $0^\circ$  and draw it to  $-90^\circ$  or we could start it at  $270^\circ$  and draw it a positive  $90^\circ$ . This is a great way to intergrate some coding in your trigonometry class.

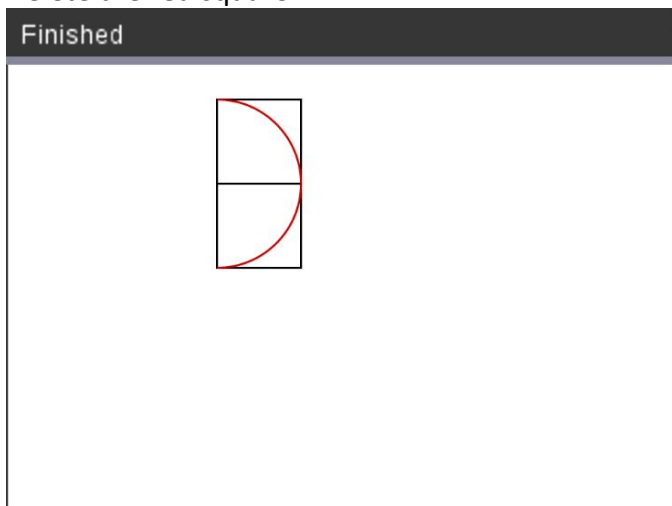
Deleting the red square will leave you with the below, drawing on the left. Next step is to create the next square, which is still 40 x 40 and has the same x-coordinate as the first square, but it's y-coordinate decreases by 40.



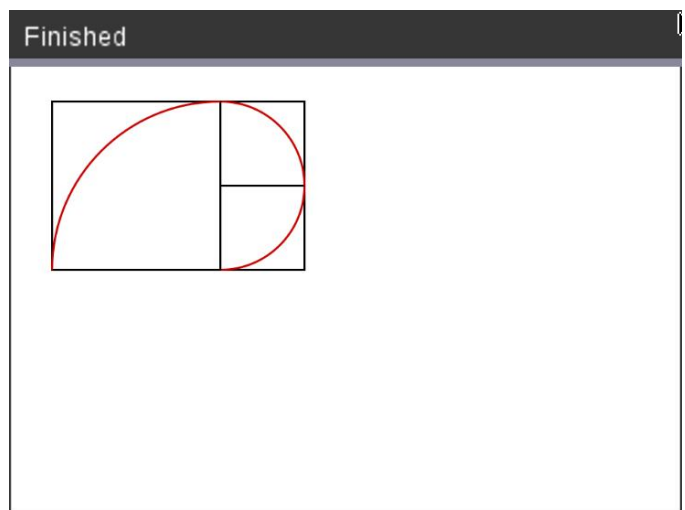
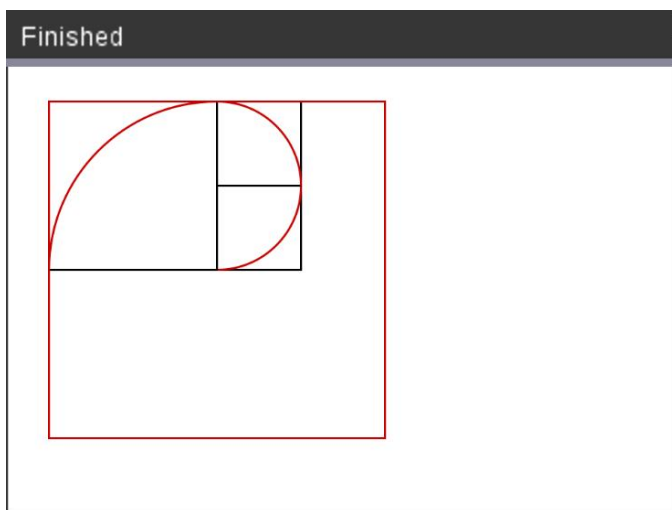
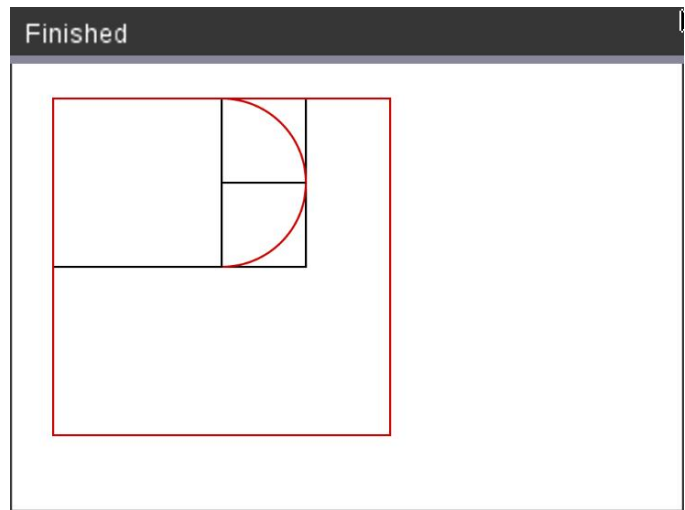
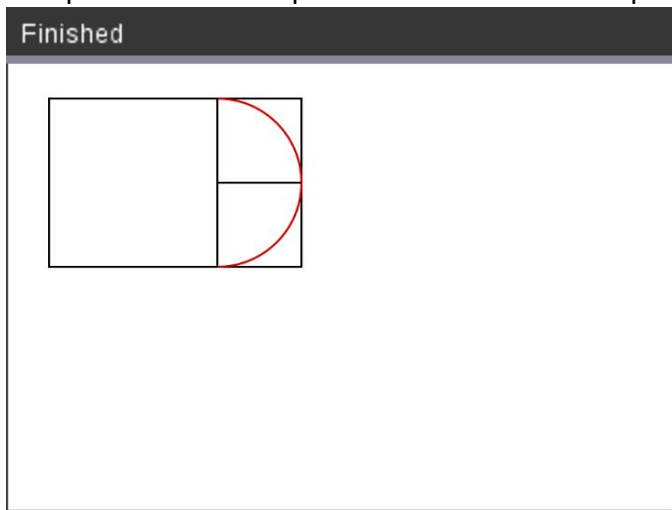
The second arc is fairly easy to create since it is part of that red square that we previously had. Keep the same upper left corner coordinates, width and height. The arc will begin at  $0^\circ$  and be drawn  $90^\circ$ .



Delete the red square:



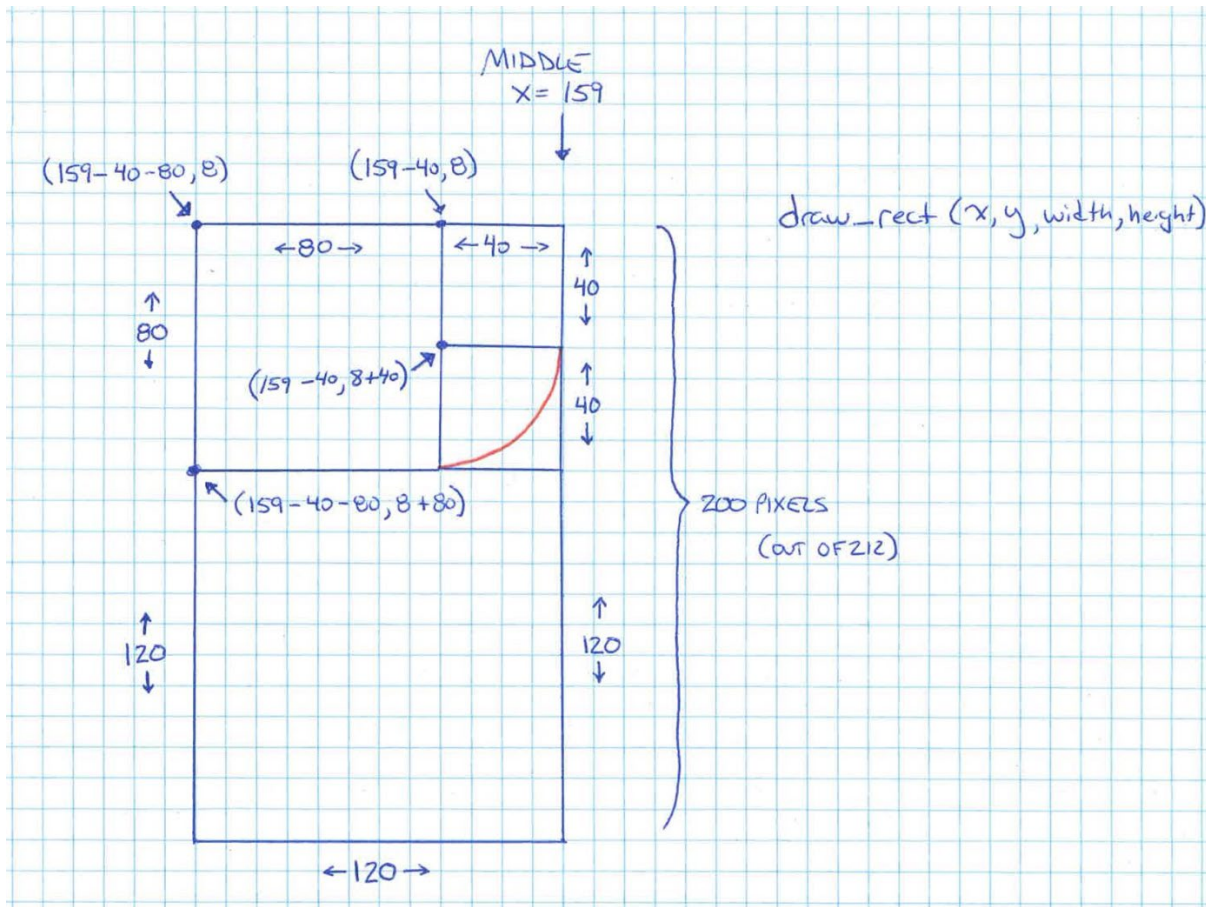
This process will be repeated for the next two squares.



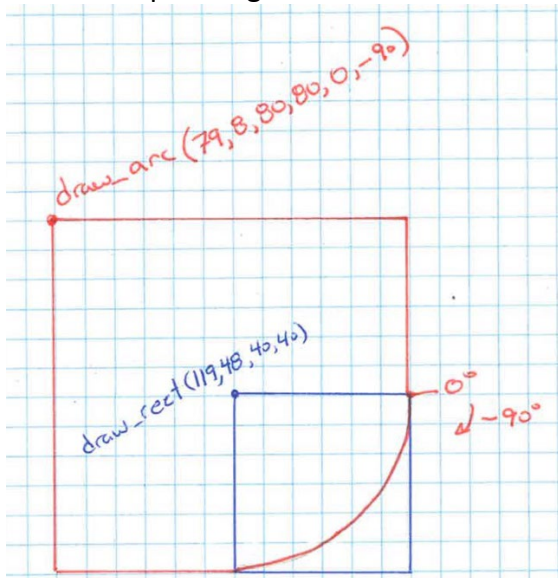
*Note: The last square will require a red square that has an upper left coordinate that is off the screen. It is possible to plot it in negative values.*

Once done with this half, you will repeat the process for the other half. In order to plot different color squares/arcs you will have to change your color by using `set_color(255,0,0)` which is red and `set_color(0,0,0)` for black. Black is the default color, if one is not set. It is not necessary to change colors but I did for demonstration purposes. You can also instead of `draw_arc()` use `fill_arc()` which will create a filled in arc and get you that full heart

Below, you can see some of the prep-work that goes into creating images. Encourage students to draw things out on paper first, as it allows for you to have a nice plan prior to coding anything. It helps you organize, think, and save some time. I will do this a lot when working on more complex drawings.



Here I was planning out the first arc or the first square.



There are probably many ways to do this project. This is just one that I happened to come up with. If you use this in your classroom to spread math love, or an activity to celebrate Fibonacci Day, or even as a Valentine's day gift to your special someone, and you tweet it out, please be sure to tag me @POSitive\_MATH

## Spoiler Alert...

Here is the complete code if you get stuck:

codeLines.py 2/31

```
from ti_draw import*

set_color(255,0,0)
fill_rect(119,48,40,40)
fill_arc(79,8,80,80,0,-90)
fill_arc(79,8,80,80,0,90)
fill_arc(39,8,160,160,90,90)
fill_arc(39,-32,240,240,180,90)
fill_rect(159,48,40,40)
fill_arc(159,8,80,80,180,90)
fill_arc(159,8,80,80,180,-90)
fill_arc(119,8,160,160,90,-90)
fill_arc(39,-32,240,240,0,-90)
set_color(211,211,211)
draw_rect(119,48,40,40)
draw_rect(119,8,40,40)
draw_rect(39,8,80,80)
draw_rect(39,88,120,120)
draw_rect(159,8,40,40)
draw_rect(159,48,40,40)
draw_rect(199,8,80,80)
draw_rect(159,88,120,120)
draw_arc(79,8,80,80,0,-90)
draw_arc(79,8,80,80,0,90)
draw_arc(39,8,160,160,90,90)
draw_arc(39,-32,240,240,180,90)
draw_arc(159,8,80,80,180,90)
draw_arc(159,8,80,80,180,-90)
draw_arc(119,8,160,160,90,-90)
draw_arc(39,-32,240,240,0,-90)
```