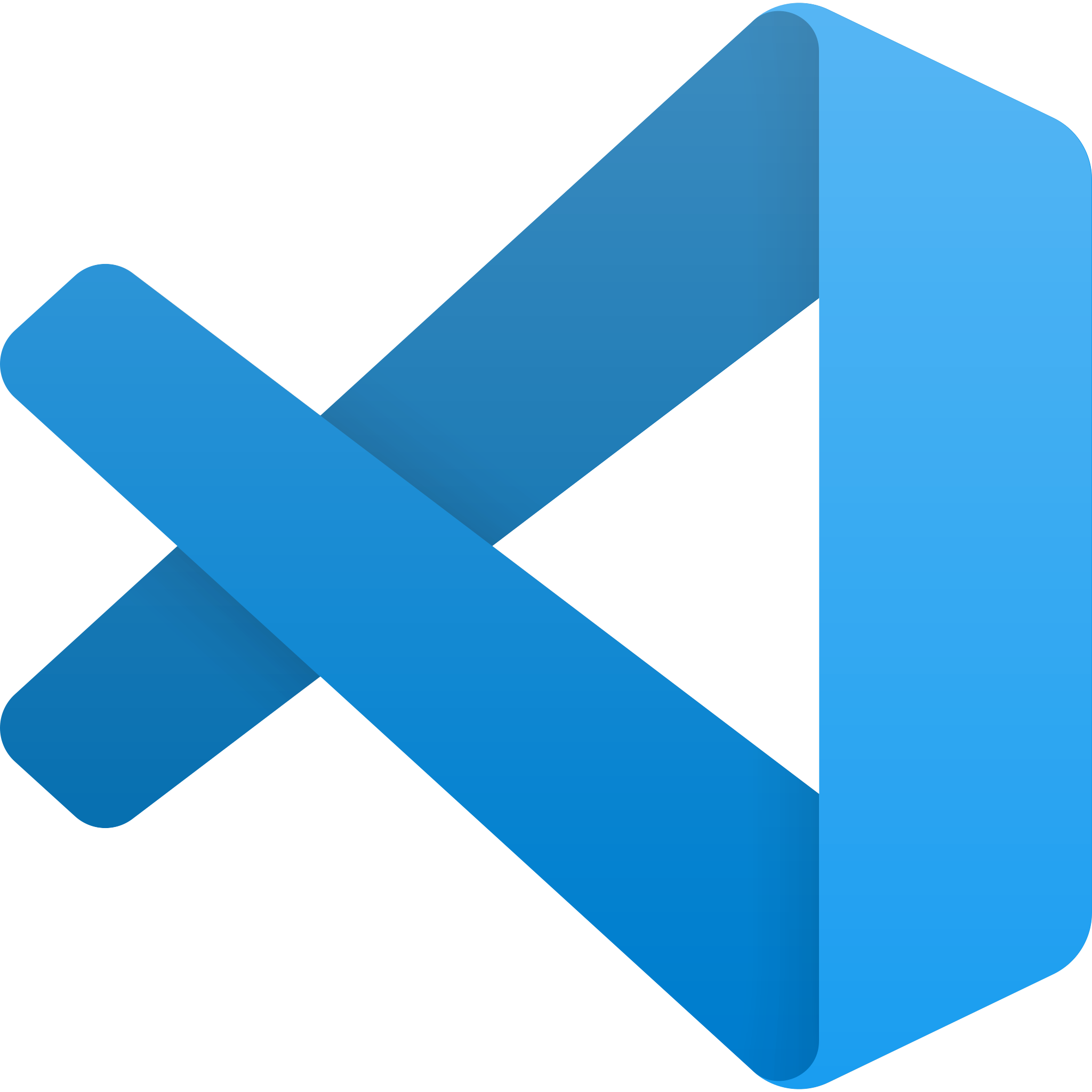
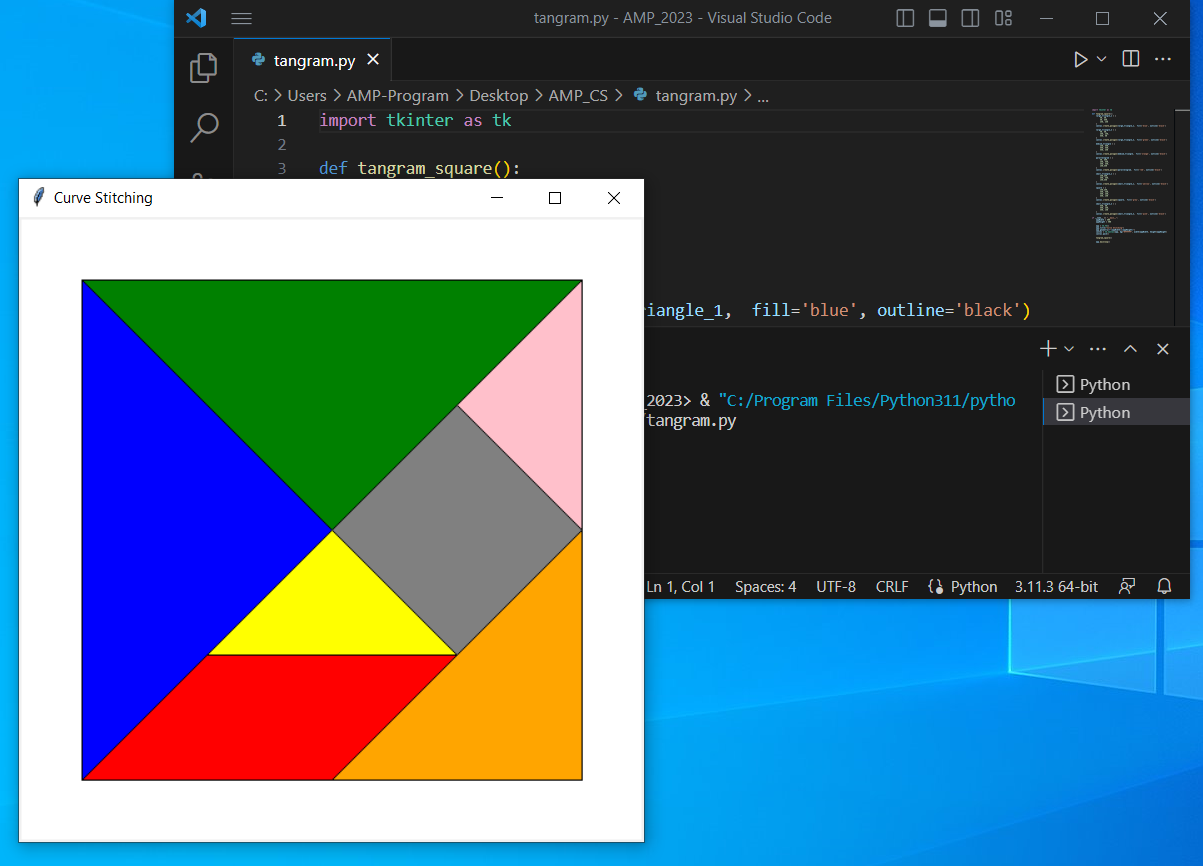
**Computational Thinking - Warm-Up #2**

| **Use VSCode to run a Python program (**[**tangram.py**](http://tangram.py)**) which draws a Tangram square.** |
| --- |

1. Create an **AMP\_CS** folder on your desktop.
   1. This folder will help you organize all of the Python files you’ll work with over the summer.
2. Download [tangram .py](https://drive.google.com/file/d/1BQCI4ITZo6j_pbr8_q5WUxW658oy5f1I/view?usp=drive_link) to your **AMP\_CS** folder.
3. Open VS Code 
   1. VS Code is a source code editor that makes coding in Python (and most other languages) really easy. You may have to search for VSCode in the application search bar. The VSCode application icon looks like this:
4. Use VS Code to open **tangram.py**
5. Click on the Play button to Run the Python File. 
   1. If this button doesn’t appear in the upper right corner once you open **tangram.py**, you haven’t successfully installed the VSCode extensions yet.
6. A graphics window will open to display a picture of tangrams in a square configuration.
   1. **Note**: You close the graphic window by clicking the X in the upper righthand corner of the graphics window



| **Getting to Know tangram.py** |
| --- |

There are three sections of code in **tangram.py**:

* Library imports
* The **tangram\_puzzle()** function definition
* The **if \_\_name\_\_ == "\_\_main\_\_":** entry point for the Python interpreter

#### Library imports

The first line in tangram.py loads a graphics library called tkinter:

| **import** tkinter **as** tk |
| --- |

#### tangram\_puzzle()

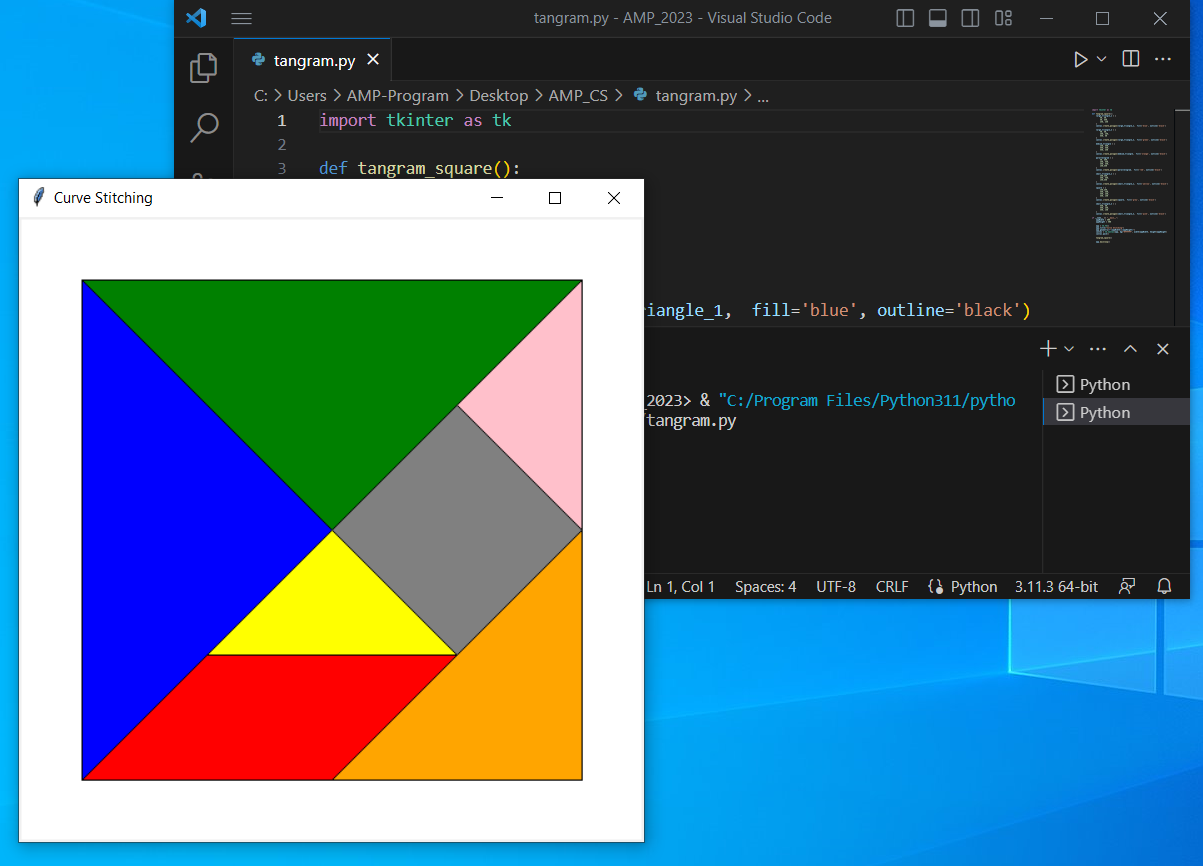
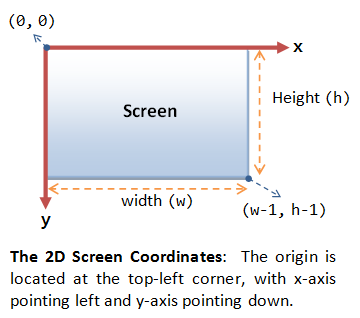
This section of code defines all the steps/code needed to actually draw each piece of square tangram puzzle. For instance, the following code draws the parallelogram:

| parallelogram = [  50, 450,  250, 450,  350, 350,  150,350  ]  canvas.create\_polygon(parallelogram, fill='red', outline='black') |
| --- |

First, a list of vertex coordinates is defined and stored in a variable called parallelogram.

Those coordinates are then used to draw a red parallelogram with a black outline.

Notice how the **origin (0, 0)**  for computer graphics is in the upper-lefthand corner of the graphics window:



#### if \_\_name\_\_ == "\_\_main\_\_":

This section of code is where the program actually starts running. We'll refer to this as the "\_\_main\_\_" section.

You can probably infer from the text that this where we can set and change the size of the graphics window:

| **if** \_\_name\_\_ == "\_\_main\_\_":  appWidth = 500  appHeight = 500  ...     **tangram\_puzzle()**   ... |
| --- |

Lines 59 - 63 configure the graphics window by, including setting the title, size, and background color of the graphics window.

The **tangram\_puzzle()** function is invoked on line 65. This is what runs the code that actually draws the tangram square.

It might seem weird to put the code that runs the **tangram\_puzzle()** function at the bottom of the file. However, Python requires that you define functions before you can use them.

### **Indentation**

Indentation is very important for Python syntax. Notice how code in the the  **tangram\_puzzle()** section uses a consistent indentation. Code in the **if** \_\_name\_\_ == "\_\_main\_\_": section also uses a consistent indentation.

**Syntax Highlighting**

VS Code provides graphical hints to help you keep track of syntax. This includes using color to indicate different

| **Exercise #1: Modify the Python program.** |
| --- |

**For each of these activities, be sure to close the old graphics window before moving on to the next exercise. If you don't close the old graphics window, then it will look like the Play button isn't working.**

1. **Change the appWidth variable to 800.**
   1. Save your changes, then rerun the code.
   2. What happens as a result of this change?
2. **Change the appHeight variable to 800**
   1. Save your changes, then rerun the code.
   2. What happens as a result of this change?
3. **Change the fill color of all 7 tangram pieces to "purple"**
   1. Save your changes, then rerun the code.
   2. What happens as a result of this change?
4. **Make a small change to the indentation of the tangram\_puzzle() statement in the "\_\_main\_\_" section (line 65):**

| **if** \_\_name\_\_ == "\_\_main\_\_":  appWidth = 500  appHeight = 500   app = tk.Tk()  app.title("Curve Stitching")  app.geometry(f"{appWidth}x{appHeight}")  canvas = tk.Canvas(app, bg="#FFFFFF", width=appWidth, height=appHeight)  canvas.pack()   tangram\_puzzle() *#two extra spaces should be added to the start of the line*    app.mainloop() |
| --- |

* 1. Save your changes, then rerun the code.
  2. How does VS Code let you know there is a mistake in the code?

1. **Fix the indentation from exercise #4, then remove the underscore in the call to the canvas.create\_polygon which draws the large blue triangle (line 9):**

| **def** **tangram\_puzzle**():  large\_triangle\_1 = [  50, 50,  50, 450,  250, 250  ]  canvas.create polygon(large\_triangle\_1, fill='blue', outline='black')  .... |
| --- |

* 1. Save your changes, then rerun the code.
  2. How does VS Code let you know there is a mistake in the code?

| **Exercise #2: Create a Python version of your tangram puzzle.** |
| --- |

1. **Start with the original version of tangram.py, then change the coordinates of each tangram piece to construct the solution to the tangram puzzle you solved in the first Warm-Up activity.**
   1. Feel free to use an image editing program like MS Paint to find the coordinates of the corners of each tangram shape in your solution.
   2. Don’t forget that the **origin (0, 0)** in computer graphics is the upper-lefthand corner of the graphics window.
   3. Save this as **tangram\_solution.py**
2. **Take a screenshot of your Python-coded tangram solution.** 
   1. Save this as **tangram\_solution.png**

| **Submit your materials to the Computational Thinking Warm Up #2 Assignment on Gradescope.** |
| --- |

1. Visit the Gradescope link for your class
2. Click on the Computational Thinking Warm-Up #2 Assignment
3. Answer the following questions:
   1. What happens when you change the **appWidth** and **appHeight** variables?
   2. How did you change the fill color of all tangram pieces to purple? What changes did you make to the Python code?
   3. How does VS Code let you know that there are mistakes in the code? Was there a difference between a mistake involving indentation versus a mistake involving naming?
   4. How did you know which numbers to change in tangram.py order to create the solution for your tangram puzzle?
4. Upload the file you created for this assignment:
   1. **tangram\_solution.py**
   2. **tangram\_solution.png**