Objective:
The purpose of this lab is to practice defining and testing methods as well as implementing conditional logic.

Assignment:
For this lab, you will be writing your own methods to draw lines.

- This assignment should be completed individually.
- Your program should contain useful comments
  - The program should begin with comments identifying the lab title (lab2), author’s name, TA name, and lab section.
  - Each section of your program should be commented sufficiently that it is easy for the TA to understand.
- The file should be in your lab directory and named lab2.py.

Part 1.

Add the `drawHLine` method to your program:
1. Using the method developed in lecture as an example, add a `drawHLine` method to your program that draws horizontal or sloped line segments. Details for this method are as follows:
   a) The method has 6 input parameters, as listed below. Use appropriate names for each and define them in the following order:
      - reference to a Raster image
      - starting x coordinate for the line
      - starting y coordinate for the line
      - ending x coordinate for the line
      - ending y coordinate for the line
      - color for the line
   b) The method should include computation of the slope of the line. The method should be able to draw either a horizontal or shallow sloped line segment whenever it is called within your program and is passed the required parameter data.

Add the `drawVLine` method to your program:
2. Clearly, the same approach used to draw a horizontal or shallow sloped line is not appropriate for use with vertical lines. Using the `drawHLine` method in your program as an example, add a `drawVLine` method to your program that draws vertical line segments. Details for this method are as follows:
   a) The method has 6 input parameters, as listed below. Use appropriate names for each and define them in the following order:
      - reference to a Raster image
      - starting x coordinate for the line
      - starting y coordinate for the line
      - ending x coordinate for the line
      - ending y coordinate for the line
      - color for the line
   b) The method should include computation of the “slope” of the line. The method should be able to draw a vertical line whenever it is called within your program and is passed the required parameter data.
Issue with the drawing of lines:
3. Alter your `drawHLine` and `drawVLine` so that they will still work if the starting point has larger values than the ending point. For example, a call to `drawHLine(pic,0,0,50,50,white)` will work fine; however, a call to `drawHLine(pic,50,50,0,0,white)` will not work as intended. Modify your code so that the methods will work correctly with any starting point and any ending point.

Testing suggestions:
4. Call your `drawHLine` and `drawVLine` methods a minimum of 3 times each, providing different data for each call. You should use the same Raster image already created.
   a) Select test data values for points that will cause horizontal lines to be drawn in various locations.
   b) Select test data values for points that will cause vertical lines to be drawn in various locations.
   c) Select test data values for points that have larger starting points than ending points for both horizontal and vertical points.
   d) Verify that your lines are drawn correctly.

Part 2.

In Part 1 of this lab, you had the responsibility of calling the appropriate line-drawing method, either `drawHLine` or `drawVLine`, based on your knowledge of the data you selected for testing. It is much more effective to implement process abstraction in such a way that you do not have to be concerned about the details of which method has to be called or exactly how each method handles the actual drawing task.

Add the `drawAnyLine` method to your program:
5. Define and include a method named `drawAnyLine` in your program. Details for this method are as follows:
   a) The method has 6 input parameters, as listed below. Use appropriate names for each and define them in the following order:
      - reference to a Raster image
      - starting x coordinate for the line
      - starting y coordinate for the line
      - ending x coordinate for the line
      - ending y coordinate for the line
      - color for the line
   b) The `drawAnyLine` method should call either the `drawHLine` or `drawVLine` method, based on the change in x coordinate values and the change in y coordinate values. This method should also handle drawing a single point.

Testing `drawAnyLine`:
6. Draw a house. If you’re really feeling creative we suggest:
   a. Two-story mansion.
   b. Oak tree on the side.
   c. Sun shining.
   d. Dog house accompaniment.