**Part 1 Getting Started**

Both partners should log-in to Moodle and find Lab 2. One partner downloads and displays the directions. On the other computer, make a folder on your h: drive named Lab 2. Download the four .py files. Make a folder on your h: drive named Lab2. Move the four .py files from the Downloads folder to your Lab 2 h: drive folder.

**Part 2 Tracing code**

On paper "by hand", trace through the following code giving the output.

x = 5

y = 50

a = 1

for z in range(1, 6, 2):

x = x + z

print('x =', x)

y = y – x

print('y =', y)

a = a \* z

print('a =', a)

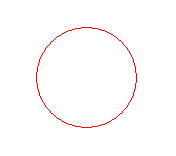
Open the file trace.py and execute the code. If your "by hand" answer is not correct, but your partner's is, have your partner explain step-by-step how to trace through the code. If neither partner is correct, raise your hand for assistance.

**Part 3 Graphics with variables**

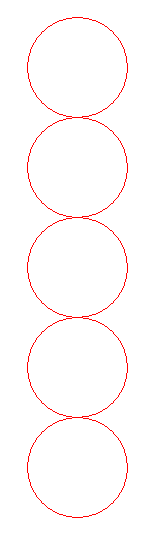
In Lab 1 smiley.py, every x and y coordinate used to draw a shape was a numeric literal (ie. 375). Those coordinates can, however, be a variable storing a number.

* Open circle\_design.py. Add your names to the top doc string. Execute the program.

Right now, just one circle is visible even though five circles are being drawn. They are all drawn on top of each other.



The first circle is supposed to be drawn with center x = 55 and y = 55. The other circles, however, should have different y coordinates so that the five circles will be drawn in a stack like this.



* Change the y-coordinates of circles 2-5 so that the circles will be drawn as a stack. Execute your code to verify.

Those stacked five circles can be drawn using a loop where the y-coordinate changes in the body of the loop. Look at the y-coordinates for circles 1-5. Notice a pattern in the y-coordinates?

* Open the file circle\_design\_with\_loops.py and add your names to the top comment.
* Replace the ??? in the range statement and the statement accumulating y so that the loop will draw the same stack of circles.
* Save and execute your code. You should have a stack of 5 circles just like before, but this time drawn with a loop.
* Replace the statement

numCircles = 5

with a statement that will allow the user to enter numCircles from the keyboard. Assume that the user will enter a positive integer and small enough so that the circle stack will fit in the window. Your code should draw a stack of the number of circles entered by the user. Execute your code to check your change.



* Replace the statement

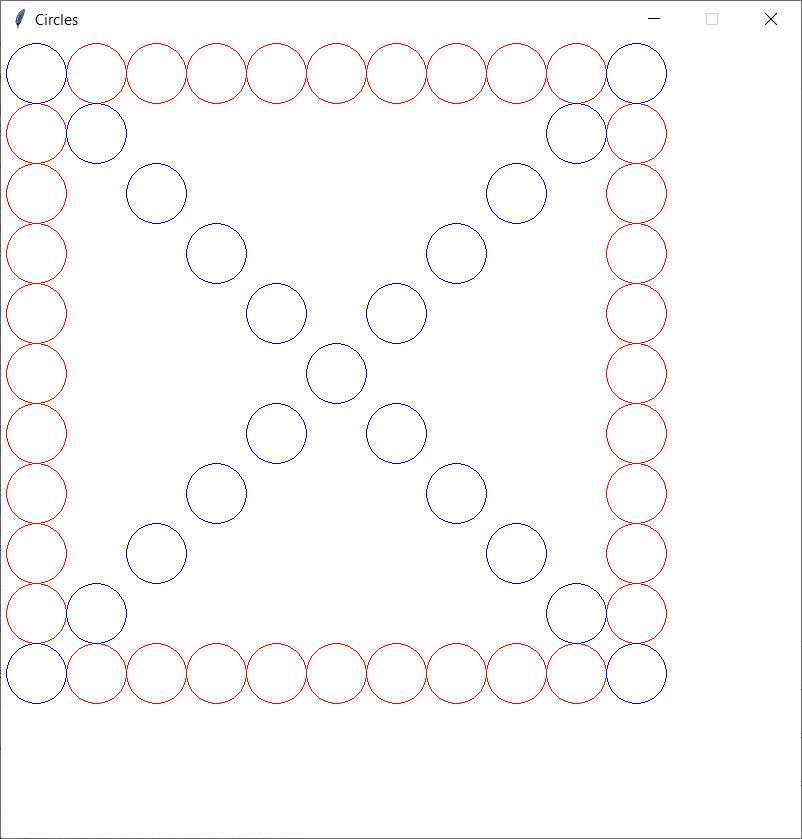
5 pixels

radius = 50

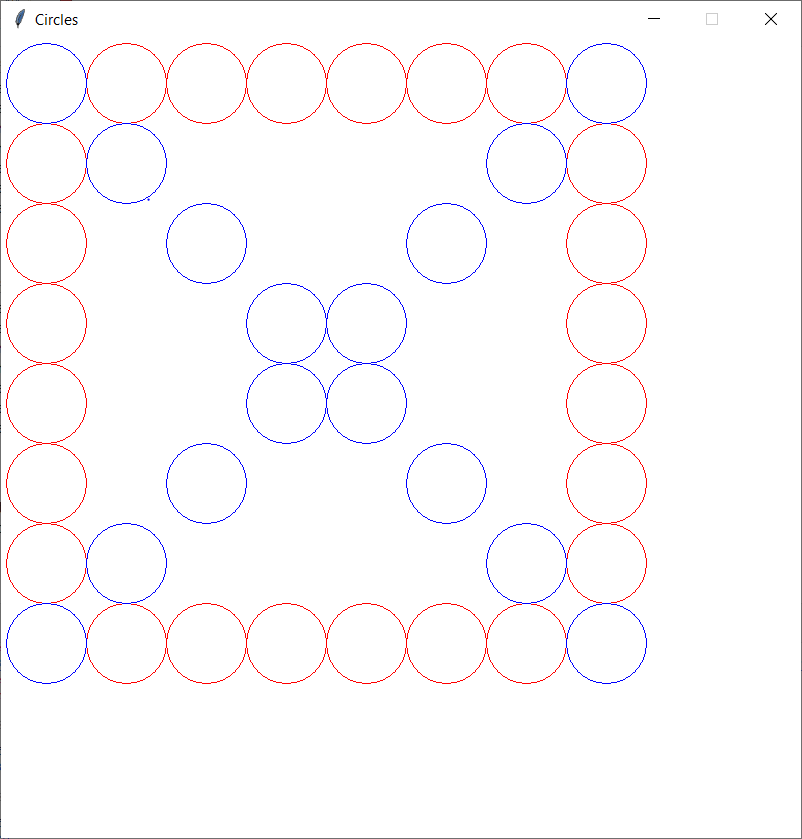
with a statement that will allow the user to enter the radius from the keyboard. Assume that the user will enter a positive integer and small enough that the circle stack will fit in the window. Your code should draw a stack of the number of circles and radius entered by the user so that the top circle is 5 pixels from the top and left edge. (Hint: change the x and y before the loop.) Execute your code to check your changes. The circles should still be "tangent".

* Test your code with different numbers of circles and different radii. Be confident that this is correct before continuing since you will be adding to this code.

For the final part of the lab, you will alter circle\_design\_with\_loops.py to generate images like the one below. The user will enter the number of circles on each side of the square shape and the radius of each circle. This image was generated when the user entered 11 for the number of circles and 30 for the radius.



This is the one generated when the user entered 8 for the number of circles and 40 for the radius.



Begin by adding the comment

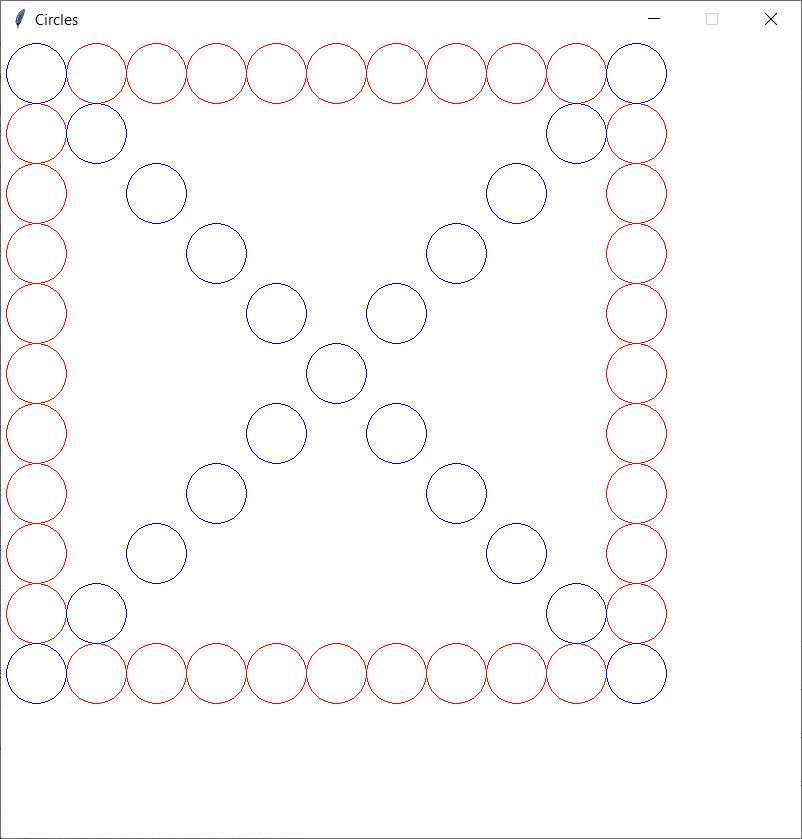
**# draws the left vertical column of circles**

above x = ??? statement which begins the code creating the left vertical column.

Although there are a variety of ways to accomplish this design, I suggest that you use 6 groups for loops, one for each step shown on the diagram below. For each step in the design, make a copy of the code drawing the first vertical column from the comment through the end of the for loop and adjust it to draw the step you are working on. Think about the x, y coordinate of the first circle and then how the x and/or y coordinates are changing to get to the next circle.

**If this seems overwhelming, start by drawing the design for a radius 50 and then generalizing to the radius entered by the user.**

If you choose a different strategy, your comments should clearly document your strategy.



Step 6

Step 5

Step 4

Step 3

Step 1

Step 2

If you need more hints, look on the next page.

**More detailed hints for the top row:**

Copy from your comment **# draws the left vertical column of circles** though the end of the for loop and paste it after the for loop. Change the comment to say **# draws the top horizontal row of circles**. The first circle in the top row has the same x, y coordinates for its center as the first circle in the top row which means the x = radius + 5, y = radius + 5 are still the place to start the top row. For the top row, the x coordinate of each circle changes value instead of the y coordinate. That means the statement y = y + 2 \* radius, would instead need to be x = x + 2 \* radius.

Make the changes describes above, save and execute your program. It should draw the left column and the top row for the number of circles and radius entered by the user.

**Starting the bottom horizontal row:**

The key is to determine the starting coordinates of the center of the first circle (the left most). How does the x coordinate of its center relate to the x coordinate of the top circle of the left column? How does the y coordinate of its center relate to the y coordinate of the top circle of the left column, the number of circles entered by the user, and the radius entered by the user?