The goal of this fourth Python programming assignment is again to write your own code inside a provided program framework, with some new graphical and mathematical considerations. You must create a pair of staring Anime Eyes, which have an iris, a pupil, and a highlight. Your code must adapt to the width and height of the canvas, so that the eyes perfectly fit the screen and do not overlap any edges, for any arbitrary canvas size. This is shown as follows:

In the JES environment, type in the following program code exactly as you see it here and on the next page, including the comments, and save the program as Lab4.py in your Python folder. Where you see my name in the comment code, replace it with your own name. As always, be very careful about indentation and capitalization. Where you see the | symbols open up several blank lines; this is where you will be writing your own code at a later time.

```python
# William T. Verts - Lab #4 - Anime Eyes

# This function paints a filled circle of radius R
# and color NewColor on the Canvas, using <Xc,Yc> as
# the coordinates of the center of the circle.
# Xc, Yc, and R may all be floats.
#--------------------------------------------------------------

def addCircleFilled (Canvas,Xc,Yc,R,NewColor=black):
    |
    |
    return
```
This function paints one anime eyeball on the Canvas, centered at \(<X_c,Y_c>\). The color of the iris is NewColor, the pupil is black, and the highlight is white. The sizes and positions of the iris, pupil, and highlight are derived from center point \(<X_c,Y_c>\) and from radius \(R\).

```python
def EyeBall (Canvas,Xc,Yc,R,NewColor):
    return
```

This function paints two anime eyeballs on the canvas by calling EyeBall twice (once for each eye). It first must determine the correct radius and center positions for each eye before calling EyeBall.

```python
def Stare (Canvas,NewColor):
    repaint(Canvas)
    return
```

This function establishes the size of the canvas and the color of the anime eyes. These values are currently hard-coded into the program, but could be input from the user instead.

```python
def Run():
    Canvas = makeEmptyPicture(640,400,white)
    show(Canvas)
    Stare(Canvas,blue)
    return
```

In JES click the Load Program button. At the >>> prompt, type Run() with the parentheses and press Enter. The program should run as it is shown here, but should not do anything except put a blank canvas on screen. Fix any syntax errors or other mistakes. We will not change Run while developing the program, but we will fill in code for the other three functions. Once the program is complete, you are allowed to change the size of the screen (the 640 and the 400), and the color of the eyeballs (the blue) to values of interest to you.
addCircleFilled

In the blank space you opened up in the addCircleFilled function, write new code to paint a filled circle with the center of the circle at location \( <X_c, Y_c> \) with radius \( R \) and color NewColor. Unfortunately, all you have to work with is the JES function addOvalFilled, which has the following formal parameters:

\[
\text{addOvalFilled(picture, startX, startY, width, height, color)}
\]

In this JES function, \( startX \) and \( startY \) are the coordinates of the upper-left corner of an oval (not a circle), and \( width \) and \( height \) describe the size of the oval’s bounding box. We want to use those parameters to create a circle, as shown below:

You have to figure out how to convert the \( X_c, Y_c \), and \( R \) of our new addCircleFilled function into the \( startX, startY, width, \) and \( height \) that can be used to call the addOvalFilled function of JES. (We’ve done a similar function in class.)

Remember that addOvalFilled requires integer parameters; many of the values calculated in this program are passed to addCircleFilled as floating-point numbers and must be rounded and converted to integers before they can be used in the call to addOvalFilled. You are allowed to type in and use the INT function described in the Companion book.

Eyeball

Next, in the blank space you opened up in the Eyeball function, write new code to paint a single anime eyeball with the center of the circle at location \( <X_c, Y_c> \) with iris radius \( R \) and iris color NewColor. You will need to call the addCircleFilled function three times, once for the iris, once for the pupil, and once for the highlight. The hard part is figuring out the centers and radii of the three circles.

The radius of the iris (the colored area) is parameter \( R \). The radius of the pupil (the black center of the eye) is half the radius of the iris, and has the same center coordinates as the iris. The radius of the highlight (the white spot) is one-quarter that of the iris. Its center is located relative to the center of the eyeball at a distance offset to the right (in the +X direction) and upwards (in the –Y direction) by an amount calculated as follows:

\[
\text{Offset} = \frac{1}{4}R \times \frac{1}{2}\sqrt{2}
\]

This will put the center of the highlight directly on the boundary between the pupil and the iris, at 45° upwards and to the right of the center of the pupil.
**Stare**

Next, in the blank space you opened up in the `Stare` function, **write new code** to call `Eyeball` twice, once for the left eye and once for the right eye, with the correct radius and coordinates for each eye extracted from the size of the `Canvas`. If the width and height of the `Canvas` are extracted into variables \( W \) and \( H \) as follows:

\[
\begin{align*}
W &= \text{getWidth}(\text{Canvas}) \\
H &= \text{getHeight}(\text{Canvas})
\end{align*}
\]

then all the other measurements needed to find the position of the eyeballs are computed as follows:

What remains for `Stare` to compute is the radius \( R \) of the eyeballs. This is done by computing the minimum (using Python’s `min(\ldots, \ldots)` function) of \( X_{\text{left}} \) and \( Y \), times \( \frac{3}{4} \), minus 5 pixels. This may seem overly complicated, but doing it this way guarantees that the eyes will always fit onto the screen.

**Finishing Up**

That is enough information for you to figure out how to fill out the three functions and run the program. Once complete, try changing the size of the screen (the eyeball sizes should track this correctly) and the color of the irises.

When you are done, turn in your assignment through the on-line code submitter Web form.