Lab 05

Using Functions in Graphics

Turtle Graphics: Modularizing Code with Functions

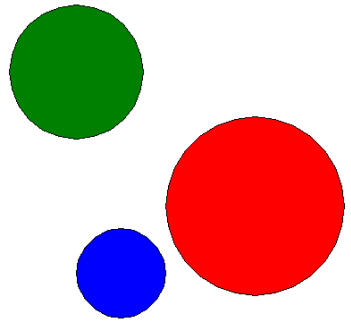
- The following function draws a circle. The parameters specify the location, radius, and color.

```
def circle(x, y, radius, color):
    turtle.penup()              # Raise the pen
    turtle.goto(x, y - radius) # Position the turtle
    turtle.fillcolor(color)    # Set the fill color
    turtle.pendown()            # Lower the pen
    turtle.begin_fill()         # Start filling
    turtle.circle(radius)       # Draw a circle
    turtle.end_fill()           # End filling
```

Turtle Graphics: Modularizing Code with Functions

- The following code calls the previously shown `circle` function to draw three circles:

```
circle(0, 0, 100, 'red')
circle(-150, -75, 50, 'blue')
circle(-200, 150, 75, 'green')
```



Program 5-30 (draw_circles.py)

```
1  import turtle
2
3  def main():
4      turtle.hideturtle()
5      circle(0, 0, 100, 'red')
6      circle(-150, -75, 50, 'blue')
7      circle(-200, 150, 75, 'green')
8
9  # The circle function draws a circle. The x and y parameters
10 # are the coordinates of the center point. The radius
11 # parameter is the circle's radius. The color parameter
12 # is the fill color, as a string.
13
14 def circle(x, y, radius, color):
15     turtle.penup()                # Raise the pen
16     turtle.goto(x, y - radius)    # Position the turtle
17     turtle.fillcolor(color)       # Set the fill color
18     turtle.pendown()              # Lower the pen
19     turtle.begin_fill()           # Start filling
20     turtle.circle(radius)         # Draw a circle
21     turtle.end_fill()             # End filling
22
23 # Call the main function.
24 main()
```

Program 5-30 Draw_ Circles.py

Turtle Graphics: Modularizing Code with Functions

- The following function draws a line. The parameters specify the starting and ending locations, and color.

```
def line(startX, startY, endX, endY, color):
    turtle.penup()               # Raise the pen
    turtle.goto(startX, startY) # Move to the starting point
    turtle.pendown()             # Lower the pen
    turtle.pencolor(color)       # Set the pen color
    turtle.goto(endX, endY)      # Draw a square
```

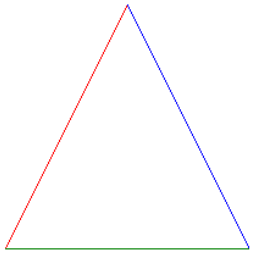
Program 5-31 (draw_lines.py)

```
1  import turtle
2
3  # Named constants for the triangle's points
4  TOP_X = 0
5  TOP_Y = 100
6  BASE_LEFT_X = -100
7  BASE_LEFT_Y = -100
8  BASE_RIGHT_X = 100
9  BASE_RIGHT_Y = -100
10
11 def main():
12     turtle.hideturtle()
13     line(TOP_X, TOP_Y, BASE_LEFT_X, BASE_LEFT_Y, 'red')
14     line(TOP_X, TOP_Y, BASE_RIGHT_X, BASE_RIGHT_Y, 'blue')
15     line(BASE_LEFT_X, BASE_LEFT_Y, BASE_RIGHT_X, BASE_RIGHT_Y, 'green')
16
17 # The line function draws a line from (startX, startY)
18 # to (endX, endY). The color parameter is the line's color.
19
20 def line(startX, startY, endX, endY, color):
21     turtle.penup()                # Raise the pen
22     turtle.goto(startX, startY)   # Move to the starting point
23     turtle.pendown()              # Lower the pen
24     turtle.pencolor(color)        # Set the pen color
25     turtle.goto(endX, endY)       # Draw a square
26
27 # Call the main function.
28 main()
```

Program 5-31 Draw_ Lines.py

Turtle Graphics: Modularizing Code with Functions

- The following code calls the previously shown `line` function to draw a triangle:

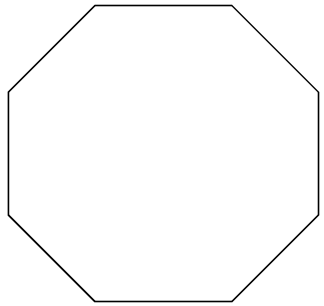


```
TOP_X = 0
TOP_Y = 100
BASE_LEFT_X = -100
BASE_LEFT_Y = -100
BASE_RIGHT_X = 100
BASE_RIGHT_Y = -100
line(TOP_X, TOP_Y, BASE_LEFT_X, BASE_LEFT_Y, 'red')
line(TOP_X, TOP_Y, BASE_RIGHT_X, BASE_RIGHT_Y, 'blue')
line(BASE_LEFT_X, BASE_LEFT_Y, BASE_RIGHT_X, BASE_RIGHT_Y, 'green')
```

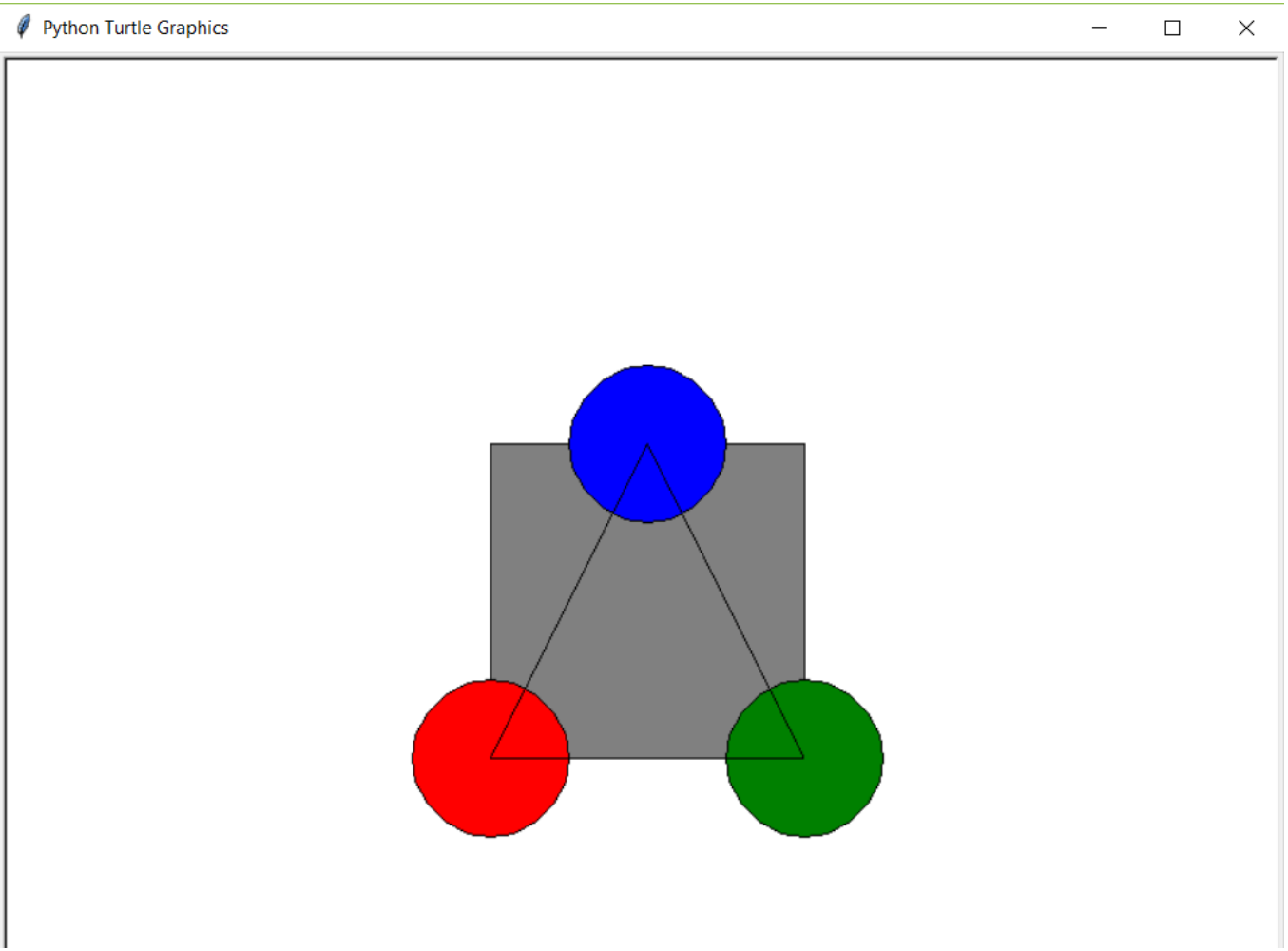
Turtle Graphics: Using Loops to Draw Designs

- This `for` loop iterates eight times to draw the octagon:

```
for x in range(8):
    turtle.forward(100)
    turtle.right(45)
```

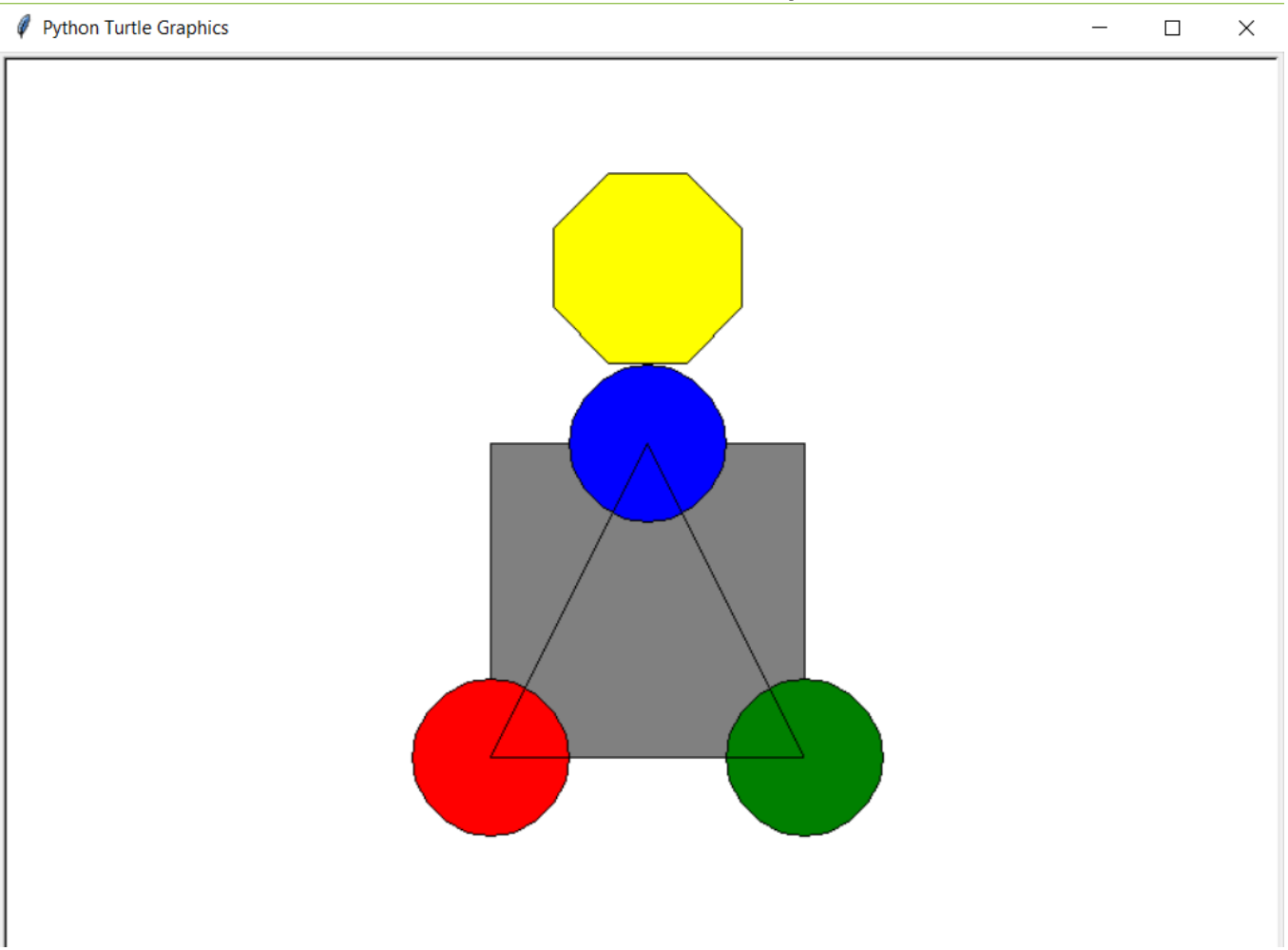


Laboratory 05A



- Using Program_5-32 and Program_5-33 as guides, type in these programs in order to obtain the result shown in the figure to the left.
- When you have got this result working, then call over the Instructor to give you credit for your work.

Laboratory 05B



- Now we want to add one more function called: **octagon(beginX, beginY, side, color)** to the functions contained in Program5-32.
- The value of **side** = 50.
- The value of **color** = Yellow.
- Choose values for **beginX** and **beginY** so that the Octagon is positioned **precisely** as shown at left.
- Add the code to implement the Octagon in Program5-33.
- Run the program, and then call the instructor to evaluate your work.

Program 5-32 (my_graphics.py)

```
1  # Turtle graphics functions
2  import turtle
3
4  # The square function draws a square. The x and y parameters
5  # are the coordinates of the lower-left corner. The width
6  # parameter is the width of each side. The color parameter
7  # is the fill color, as a string.
8
9  def square(x, y, width, color):
10     turtle.penup()                # Raise the pen
11     turtle.goto(x, y)             # Move to the specified location
12     turtle.fillcolor(color)       # Set the fill color
13     turtle.pendown()              # Lower the pen
14     turtle.begin_fill()           # Start filling
15     for count in range(4):        # Draw a square
16         turtle.forward(width)
17         turtle.left(90)
18     turtle.end_fill()             # End filling
19
20 # The circle function draws a circle. The x and y parameters
21 # are the coordinates of the center point. The radius
22 # parameter is the circle's radius. The color parameter
23 # is the fill color, as a string.
```

Program 5-32
My_graphics.py

Program 5-32 *(continued)*

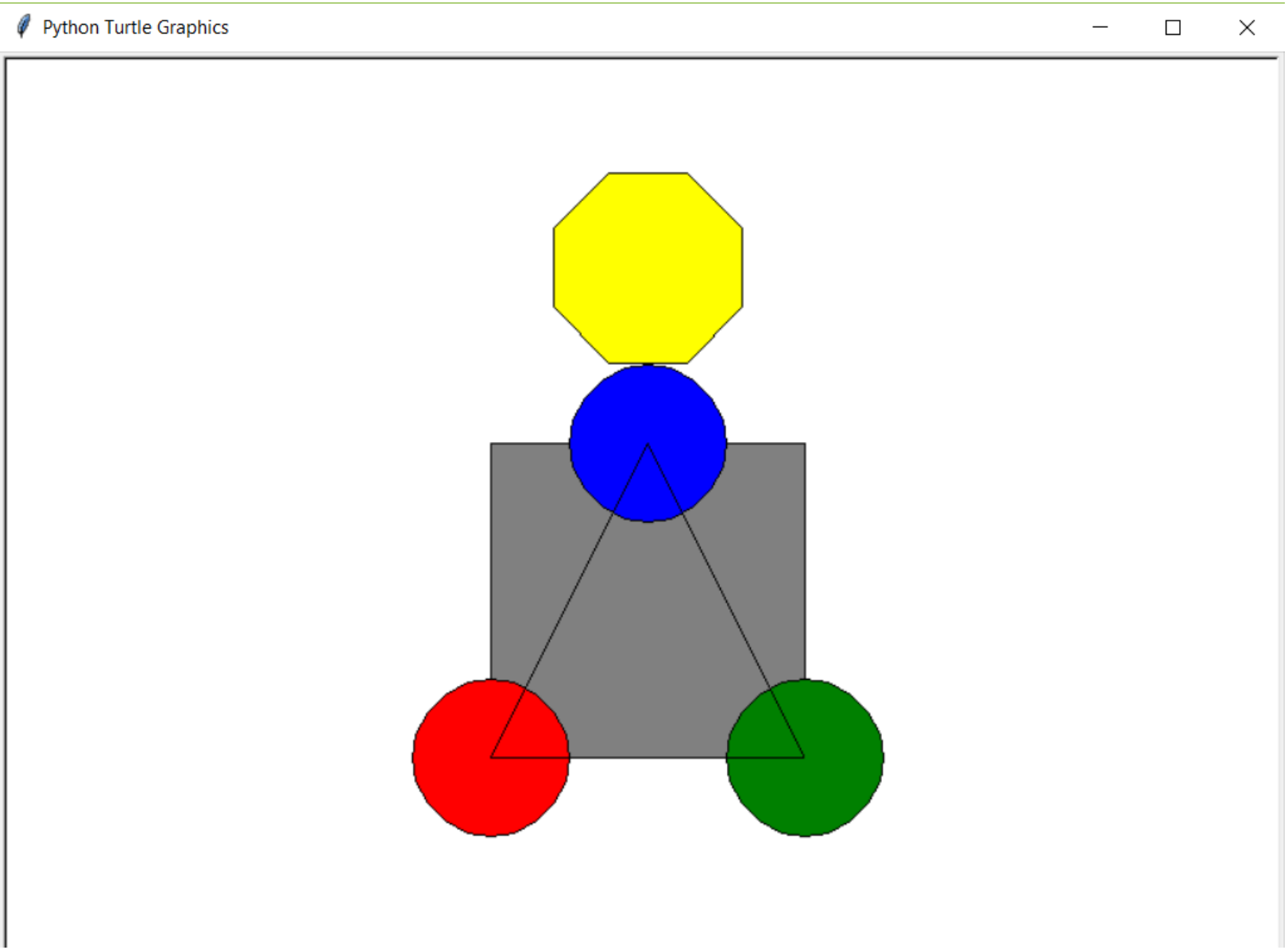
```
24
25  def circle(x, y, radius, color):
26      turtle.penup()                      # Raise the pen
27      turtle.goto(x, y - radius)          # Position the turtle
28      turtle.fillcolor(color)             # Set the fill color
29      turtle.pendown()                    # Lower the pen
30      turtle.begin_fill()                 # Start filling
31      turtle.circle(radius)               # Draw a circle
32      turtle.end_fill()                   # End filling
33
34  # The line function draws a line from (startX, startY)
35  # to (endX, endY). The color parameter is the line's color.
36
37  def line(startX, startY, endX, endY, color):
38      turtle.penup()                      # Raise the pen
39      turtle.goto(startX, startY)         # Move to the starting point
40      turtle.pendown()                    # Lower the pen
41      turtle.pencolor(color)              # Set the pen color
42      turtle.goto(endX, endY)             # Draw a square
```

Program 5-32 My_graphics.py

Program 5-33 (graphics_mod_demo.py)

```
1  import turtle
2  import my_graphics
3
4  # Named constants
5  X1 = 0
6  Y1 = 100
7  X2 = -100
8  Y2 = -100
9  X3 = 100
10 Y3 = -100
11 RADIUS = 50
12
13 def main():
14     turtle.hideturtle()
15
16     # Draw a square.
17     my_graphics.square(X2, Y2, (X3 - X2), 'gray')
18
19     # Draw some circles.
20     my_graphics.circle(X1, Y1, RADIUS, 'blue')
21     my_graphics.circle(X2, Y2, RADIUS, 'red')
22     my_graphics.circle(X3, Y3, RADIUS, 'green')
23
24     # Draw some lines.
25     my_graphics.line(X1, Y1, X2, Y2, 'black')
26     my_graphics.line(X1, Y1, X3, Y3, 'black')
27     my_graphics.line(X2, Y2, X3, Y3, 'black')
28
29 main()
```

Program 5-33 graphics_mod_demo.py



Lab05B Final Result