ECS 102 Baruch

Spring 2019

HW 7

Due: Thursday, March 28, 11:59 pm

For this assignment you will be writing 2 programs in files, **each with a main**.

1. Start each program with a comment like this:

# Official Name: *Amanda Kitredge*

# Nickname: *Mandy*

# email: *ajkitred@syr.edu*

# Assignment: *Assignment 3, problem 1.*

# Date: *September 13, 2018*

# *a brief description of the problem.*

Replace the *italicised fields* with appropriate information for you and for this assignment.

* In the "Official Name" field write your name as it appears on University documents.
* "Nickname" is the name you would like to be called by the course staff. If you don't use a nickname you can leave this line out.

2. In addition, each program should have comments

describing the main sections of the program. They should be specific to the problem you are solving. Under each of these comments should be a chunk of code implementing the comment.

Each function should have a comment explaining the parameters and return values of the function, as well as what the function does.

3. Good readability:

* Use meaningful names for all variables.
* Use blank lines to delineate small portions of code that work together to accomplish a task.

4. Test your programs. Run each program with a variety of data.

**A. ShapeMenu.py**

For this program you will be making shapes just using \*s, not in the graphics window.

Write a 3 functions. The only print statements they should use are

**print("\*", end="", file=pictureFile)** and

**print(file=pictureFile)**

**pictureFile** refers to an output file, already opened in **main**, and to be closed, later, in **main**.

Each of these 3 functions should use **nested for loops**.

def drawRect(width,height,pictureFile):

**width** is the number of stars across, and **height** is the number of stars high.

def drawSquare(side, pictureFile):

**side** is the number of stars across and the number of stars high.

drawSquare should call drawRect.

def drawTriangle(side,pictureFile):

**side** is the number of stars for both the height and base of the triangle. It is a right triangle, starting with one star, and getting one more star each row down.

Once you have tested your functions in a simple main that opens the output file, prints some shapes to it, and closes the file, you should move on.

Write a function

def menu():

It should ask the user for their choice: t for triangle, s for square, r for rectangle, or q to quit. These are the only acceptable answers. It should keep asking until a good choice is given. (Use a **data verification loop.**) The function should return the good choice.

Finally, main can do some of the work. After opening the output file, it should start asking the user for their choice, using menu. Using a **sentinel loop**, with 'q' as the sentinel, it should

* keep asking for choices (using menu)
* ask for additional needed information (lenght, height, side, base, whatever is needed)
* and drawing the shape, by calling the appropriate function.

When the user chooses to quit, main should close the output file, and let the user know the picture is ready.

Your main and all functions should be well commented.

Sample dialogue in the shell:

Enter choice:

r for rectangle

s for square

t for triangle

q to quit

>t

Enter length of base: 5

Enter choice:

r for rectangle

s for square

t for triangle

q to quit

>t

Enter length of base: 7

Enter choice:

r for rectangle

s for square

t for triangle

q to quit

>y

Enter choice:

r for rectangle

s for square

t for triangle

q to quit

>f

Enter choice:

r for rectangle

s for square

t for triangle

q to quit

>t

Enter length of base: 9

Enter choice:

r for rectangle

s for square

t for triangle

q to quit

>r

Enter length, height: 3,8

Enter choice:

r for rectangle

s for square

t for triangle

q to quit

>s

Enter length of a side: 7

Enter choice:

r for rectangle

s for square

t for triangle

q to quit

>q

Your picture is available in shapePix.txt

Sample output file, shapePix.txt

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

**B. ColorChooser.py**

This will be a graphics program. You will draw 4 rectangles and a circle. The rectangles will be labeled **red, green, blue,**  and **quit**. (You can use whatever colors you like.)

The goal will be to click a colored rectangle, and have the circle become that color. This can be repeated until the user clicks the quit rectangle.

Write a function

**def checkButtonClick(clickedPoint, leftLower, rightUpper):**

All three of the parameters are Points.

clickedPoint is the Point where the mouse has just clicked

leftLower and rightUpper are the Points defining a Rectangle.

The function should determine (using a bunch of ifs elifs elses) if the clickedPoint was inside the rectangle. It should return 1 if the click is in the rectangle and 0 if the click is not in the rectangle.

For each time the mouse is clicked **main** should then call **checkButtonClick** up to four times (in if elifs), to determine which box was clicked, if any. Once that is determined, the Circle should change to the specified color.

This all should be in a sentinel loop that keeps responding to clicks until the quit box is clicked.

Your main and all functions should be well commented.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Upload:**

ShapeMenu.py

ColorChooser.py

Make and save a receipt each time you upload a file.