

Assignment 6

Note: When you turn in an assignment to be graded in this class, you are making the claim that you neither gave nor received assistance on the work you turned in (except, of course, assistance from the instructor or teaching assistants).

Write the following Python program in a python class named Assignment6.py and upload the file to Blackboard when the assignment is complete. The goal of this program is to take a collection of rectangles and display them on a canvas so that none of the rectangles overlap with each other. To accomplish this task, your program will define several methods and classes. Note: parameters and return values **must** be in the order they are listed in the assignment manual

A class CustomCanvas

CustomCanvas will have a constructor which takes two explicit arguments, height and width, which are expected to be of type int. The constructor will create and display a new Canvas object with the height and width provided. To do this, you will use Canvas, a built-in object defined in the python package tkinter. Documentation for the Canvas class can be found here: <http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/canvas.html>

A class Rectangle

Rectangle will have a constructor which takes four explicit parameters, height, width, x, and y. All four arguments are expected to be of type int. x and y represent the origin point of the given rectangle. The origin point will be in the upper left-hand corner of a canvas. Parameters x and y should have default values of zero. The values received as parameters should be stored in instance variables of the same name.

A function pack

The pack function will have two parameters, allRect and canvasSize. allRect will be a list of Rectangle objects and canvasSize a tuple containing a canvas' height and width (in that order). pack will take the given list of rectangles and determine a location for each rectangle so that each rectangle does not overlap another and each rectangle exists within the given canvas size. pack will then return a list of placed Rectangle objects. Each given rectangle must be included in the returned list. (Note: Each given rectangle is referring to the logical concept of the rectangle shape not the specific Rectangle object. Two Rectangle objects are logically equivalent if they have the same height and width. When generating the list of Rectangles to return, you can modify the given Rectangle objects or create new, but logically equivalent, Rectangle objects)

A function main

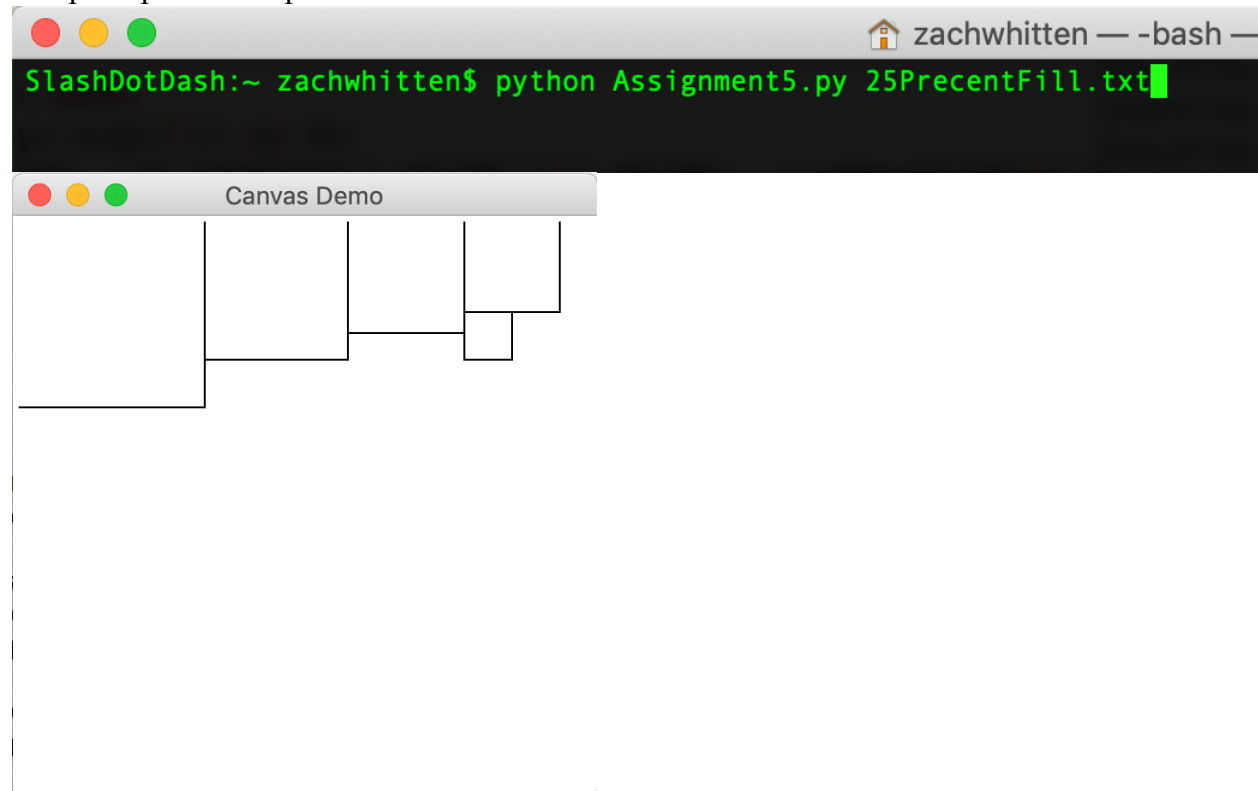
The main function will read in a filepath as a command line argument. You may assume the filepath is always the second command line argument given (the first being the name of the class being executed) The given filepath will point to a txt file containing a canvas size and rectangles. The first line of the given text file will contain two int's separated by a comma. These int's represent a canvas' height and width (in that order) All following lines represent the height and width of an individual rectangle. main should parse the data in the file and use the

information to create a new CustomCanvas object and a new list of Rectangles. Once generated, the list of Rectangles and the size of the canvas should be passed to the pack function. Main should then print each Rectangle contained in the returned list to the instantiated CustomCanvas object. Each printed Rectangle should have a black border and a colored (not black or white) fill. Main should be called whenever Assignment6.py is run as a stand-alone file but not when Assignment6.py is loaded as a library.

A few notes about the assignment:

- You can create other classes or other methods in the described classes as you see fit
- Four data files are included with this lab. You must successfully pack and draw all rectangles specified in 25PrecentFill.txt, 50PrecentFill.txt, and 75PrecentFill.txt.
 - Successfully packing and drawing the rectangles specified in 95PrecentFill.txt will result in +5 bonus points.
- You may use external libraries for this project **IF** the external libraries are installable via [pip](#)
 - If you use external libraries, include a text file in your submission indicating what libraries you used

Sample input and output:



The image shows a terminal window and a canvas demo window. The terminal window has a title bar with a home icon and the text 'zachwhitten — -bash —'. The terminal content shows a green prompt 'SlashDotDash:~ zachwhitten\$' followed by the command 'python Assignment5.py 25PrecentFill.txt' and a green cursor. Below the terminal is a window titled 'Canvas Demo'. It contains a large white rectangle with a black border. Inside this rectangle, several smaller rectangles are drawn with black borders. These rectangles are arranged in a way that they appear to be packed together, with some overlapping and others adjacent. The rectangles have different sizes and are positioned in a non-uniform, stepped pattern.

Grading Rubric:

Category	Points
Methods and Classes properly defined	20
Main function correctly reads input file when Assignment6.py is run as a standalone file	20
Proper use of canvas to draw rectangles	20
Pack function produces correct output	20
Code is commented and produces no interpreter errors	20
Total	100