

CS 1410: Illustrate

In computer science, classes and object-oriented programming are very commonly used when creating programs that involve computer graphics and graphical user interfaces. To display something visually on the screen, like a button, a class is used to represent a Button, and multiple instances of the Button class can be created to display multiple buttons on the screen at one time. This is the foundation on which all modern applications are created, whether it's on your computer, phone, or television.

Assignment

Your assignment is to create a program using Python and [Pygame](#) that displays a simple illustration that is drawn using basic shapes. Your program will use a series of classes to organize your code that draws the shapes, where the instances of your classes will contain data members which are instances of other classes. For example, you might create a `Cloud` class that is responsible for drawing a single cloud, a `Bird` class that draws a single bird, and a `Sky` class that draws a blue sky behind three clouds and six birds. In this case, the `Sky` class would have data members for the clouds and the birds, which all get drawn in turn to display the sky and everything in it.

At a minimum, your program should include:

- At least six different classes, each responsible for drawing a specific part of your illustration (e.g. `Picture`, `Sky`, `Cloud`, `Bird`).
- At least twelve instances of these classes (total), in order to draw multiple similar copies of certain elements (e.g. a white cloud on the left, and a gray cloud on the right).
- One class (of the six or more that are required) that represents the whole of your illustration. There should be just one instance of this class that contains all other instances of your other classes, either directly or indirectly (e.g. a single instance of the `Picture` class, which contains an instance of the `Sky` class, which contains several instances of the `Cloud` class, etc.).
- At least five of the following draw methods provided by Pygame: `rect()`, `polygon()`, `circle()`, `ellipse()`, `arc()`, `line()`. Rendering text also qualifies as drawing and can be considered one of your five.

Your illustration should be different than the class example and should demonstrate your own personal creativity.

Start by downloading the provided Pygame starter kit by [clicking here](#).

Extra Challenges

- Create one or more variations of your illustration, and allow the user to toggle each variation by pressing a different key on their keyboard or mouse.

Hints

- Before starting, grab a piece of paper and create a rough sketch of your illustration. Use this to help you visualize what you intend to draw, to help plan the classes that you will need to create, and to approximate the coordinates that you will need when drawing the individual shapes.
- Refer to the class example when planning your classes and how they will relate to each other, and also to see examples of how to use the various Pygame draw methods. An example sketch is also included in the class example.
- Refer to the [Pygame documentation](#) to understand which parameters are necessary when calling each of the Pygame draw methods. Specifically, you should be interested in `pygame.draw` and `pygame.Rect`.
- When creating colors, use a helpful tool to determine the RGB values. Here are two good options: color.adobe.com and colorpicker.com
- Instructions for [installing PyGame](#).

Sample

An example running program:

