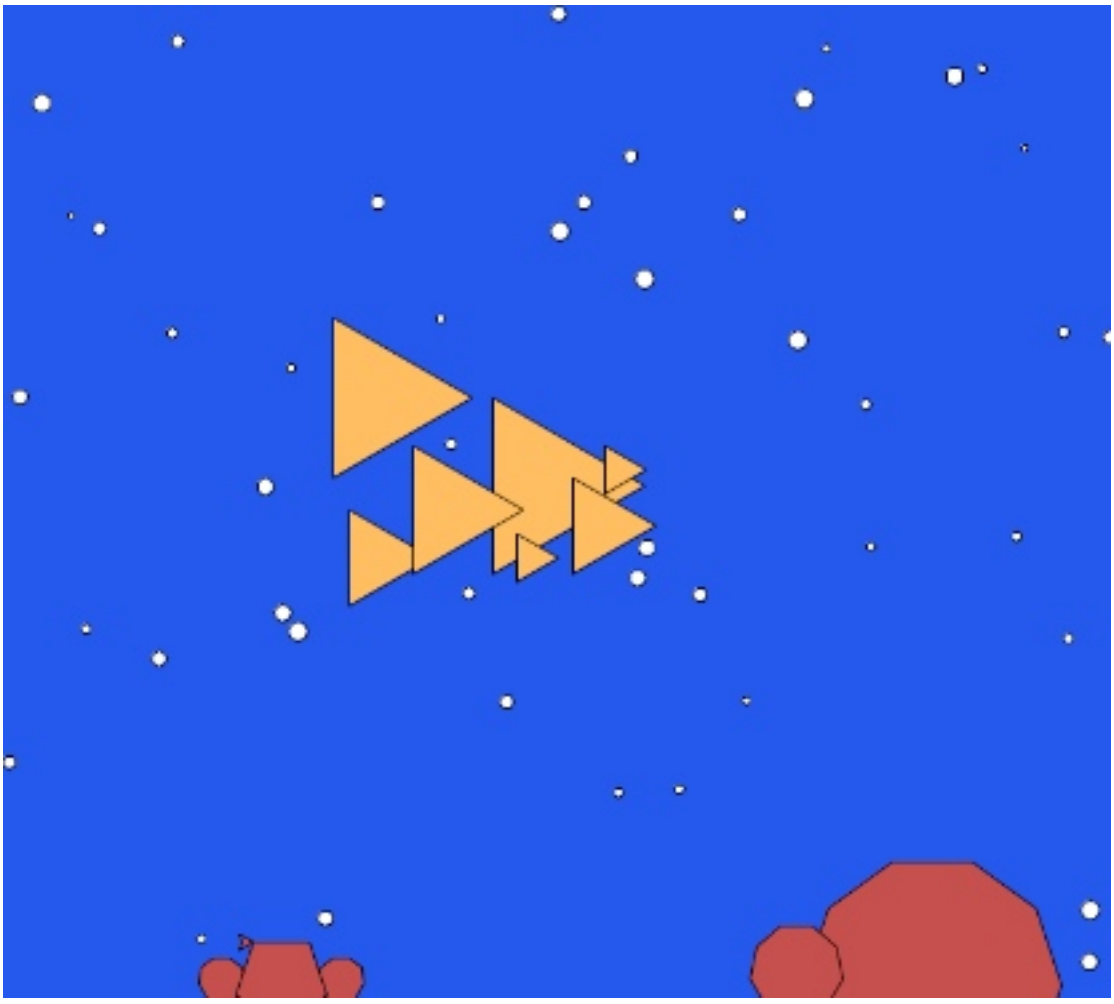
CS150 - Assignment 2
Due: Wednesday Sept. 26, at the beginning of class

For this assignment, we will be using the `turtle` graphics module to draw a picture. For your lab preparation, you should have designed the drawing and figured out the rough placement of the different components. See the lab preparation for the details of what should be included in your drawing. For this assignment you will be working on a single python file/program that generates your picture. Below is an example completed picture:



1 Style

Before you start coding, a few brief comments on style. We've talked about style components in class and now we're going to start utilizing these in your assignments. In particular, make sure you keep the following in mind as you're writing your program:

- All functions should have appropriate docstrings
- Comment your code appropriately. Comment complicated parts of the code and include your name, date, etc. at the top of the file.
- Follow the variable naming conventions discussed in class and use good variable names
- Use whitespace appropriately to make your program easier to read
- Use constants. If there are values that don't change within your program but represent constant values, you should defined capitalized constants for these at the top of your program.

2 Basic Shapes

To get started, we're going to write some functions to make some basic shapes for us. Make sure that you have these working before moving on to the next part.

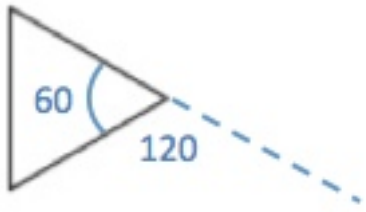
I encourage you to refer to the documentation online as you work on this:

`http://docs.python.org/library/turtle.html`

Don't forget to include the import statement for the `turtle` module at the top of your file:

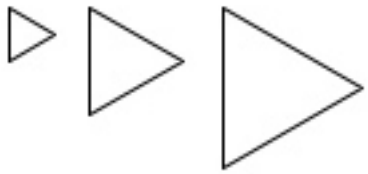
```
from turtle import *
```

- **triangle** - Write a method called `triangle` that draws an equilateral triangle (i.e. a triangle with all three sides the same length). Your function should take three parameters: the x and y coordinate where it should be drawn and the length of the side of the triangle. The triangle should be drawn so that the left edge of the triangle is vertical. For those rusty on geometry, the interior angles of an equilateral triangle are all 60 degrees, which means the angle between a straight line drawn from one side and the adjacent side is 120 degrees.



Your x and y coordinates may indicate any part of the triangle (e.g. the top left, the center, etc.). Pick whichever seems easiest.

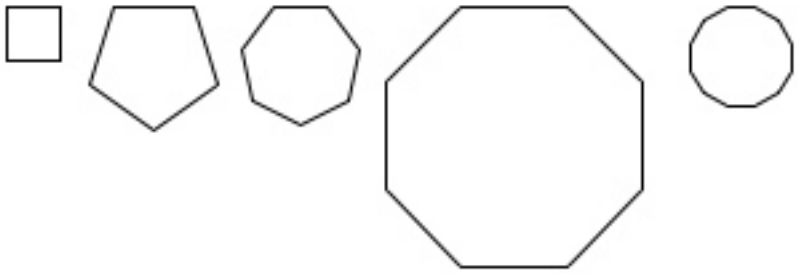
For example, here are three triangles of increasing size:



- **polygon** - Write a method called `polygon` that draws a polygon. An n-sided polygon has n equal length edges and the angle between a straight line drawn from one side and the adjacent side is $360/n$. Your function should take 4 parameters: the x and y location of the polygon, the number of sides and the length of the each side. Again, the x and y coordinates may indicate any point on the polygon, so pick whatever seems easiest.

  Unlike the triangle where you know when you're writing the function exactly how many sides the object will have, for the polygon you can't just hard-code the different line segments. Instead, you'll have to use a `for` loop.

  For example, below are some polygons of differing number of sides and size. All can be drawn with the `polygon` function.



3 The Background

At this point, you should have two functions written that draw triangles and polygons anywhere on the screen. We'll now work on filling in the background.

The key component of the background is randomly spaced circles. Write a function called `add_circles` that takes as a parameter the number of circles to add and randomly places circles of radius 4 throughout the screen. Some hints for how to do this:

- The `circle` function draws a circle with a given radius

- You'll need to figure out the dimensions of the screen (think lab prep)
- The `randint` function from the random module may be useful
- To make this available, don't forget to include `from random import randint` at the top of your program

To check that everything is working right, try adding differing numbers of circles and make sure that they're distributed throughout the screen and that the right number are actually being drawn.

Once you're sure it's working, add a bit more code to make the size of the circles random. You'll have to play with different size ranges to see what looks best. For example, if you look at the picture on the first page of this handout, you'll see that the bubbles range in size. Think about using constants here rather than hard-coding numbers in your code.

4 Putting it all together

You now have all the components required to put together your final picture. Create a function called `generate_picture` that draws the entire picture. One way to do this is:

1. Change the background to the appropriate color.
2. Change your `triangle` and `polygon` functions so that they create filled shapes (see the `turtle` documentation on how to do this if you don't remember).
3. Change your `add_circles` method to also draw filled circles.
4. Actually create your picture using your three methods. If your design from the lab prep is good, this should be fairly painless.
5. Change the fill color of the objects to make it look more realistic (e.g. orange fish and brown rocks).

5 Extra Credit

You can earn up to 2 points of extra credit on this assignment. Extra credit will be awarded by including additional elements to your picture. The amount of points given will be assigned based on creativeness and difficulty of implementation. Here are some examples, but I encourage you to include your own:

- Instead of circles, write a function called `star` that draws stars (not asterisks) (use a `for` loop)
- Include other types of objects in your scene

- Modify the `triangle` function to draw more interesting triangles, for example isosceles triangles or triangles with fins

To receive extra credit you MUST include in your comments at the top of the program what the extra credit additions you added were (otherwise, it can be hard to figure out sometimes).

6 When you're done

When you're all done you should have a working program that draws your picture. Make sure that your program is properly commented:

- You should have comments at the very beginning of the file stating your name, course (including section number), assignment number and the date.
- Each function should have an appropriate docstring
- Other miscellaneous comments to make things clear

In addition, make sure that you've used good *style*.

What to hand in:

For this assignment we're going to use a combination of paper submission *and* digital submission to hand-in the assignment.

Paper submission:

- Your program printed out
- Your drawn picture printed out. The easiest way to do this is to do a screen capture.

  On mac:
  Hold command+shift+4 and then while holding these, hit the spacebar (on a mac). If you then click on the window where your picture is drawn, an image file will be saved on your desktop entitled "Screen shot...". You can double-click on this file and then print it. You may have to click "scale" in the print options to get it to fit on one page. It won't print it color if you print in the lab, but this will give us most of the details

  On windows:
  There are plenty of resources online about how to do a screen capture. One option is:
  `http://graphicssoft.about.com/cs/general/ht/winscreenshot.htm`

Digital submission:

I'd also like you to submit your image file that you generated via screen capture online. This will give you practice using our digital submission system and I'll put together a collection of all of the images for the class.

Go to the course web page and click on the "digital submission" link. Enter the appropriate information and select your screen capture image file and submit it.

Grading

		points
`triangle`		
	equilateral	1
	correct x, y	1
	correct direction	1
`polygon`		
	correct shape	3
	varying sizes	1
	x, y	1
`add_circles`		
	correct number of circles	1
	covers entire area	1
	random locations	2
	random sizes	1
`generate_picture`		
	background color	1
	proper fills	1
	6+ fish/ships	2
	6+ rocks/planets	2
Lab prep		3
Comments, style		3
extra credit		2
total		25 (+2)