

# Homework 1: Python Basics

## Comp 123

Susan Fox

### 1 Overview

**This assignment should be done individually, not collaborating with another student.** Do not seek out solutions online. Do not work through these problems with a friend. DO ask for help from me or our preceptors. DO write down any people or web sites where you get some measure of help.

See the Homework Guidelines document for further information about standards for homework.

#### 1.1 Assignment goals

For this assignment you will work with Python data structures and operators. The goal is to practice writing and debugging simple scripts that calculate things about numbers and strings.

#### 1.2 Background

We will be using Wing for this assignment. Wing IDE 101 is a typical Python programming environment, available for free for any kind of computer.<sup>1</sup> Wing includes a program editor, an interactive shell where you communicate with Python, and a debugger. You will learn to use the debugger as time goes on. We will be using Wing in class for all of our “normal Python” work, as it uses the standard dialect of Python, called C-Python. There is a video on Moodle that shows the features of Wing..

**Hint for turtles:** You can change the speed of the turtle using the `speed` method. Speeds 1 through 10 range from very slow to very fast. Speed 0 means “go as fast as possible!”

#### 1.3 Preparing and handing in the assignment

Please read the Homework Guidelines document in Moodle before starting this assignment. It lays out my expectations, and gives some helpful tips on how to prepare homework effectively.

I would like you to put all your answers in a single Python file. Separate each question by several blank lines, and put in a hash-mark comment that indicates which question is being answered. Your comments should also state, in full sentences, what the script is doing. Any non-code answers may be included as either triple-quoted strings or comments.

---

<sup>1</sup>An IDE is an “Integrated Development Environment”, in other words, an application to help you write programs.

## 2 Assignment questions

1. (10 pts) You are planning to start a non-profit to rescue abandoned guinea pigs. You've have rented a building to convert to a guinea pig rescue center. The building has 800 square feet of space in it. You want to know the maximum number of guinea pigs you can house at your rescue center. You are going to plan cages to hold at most two guinea pigs, and you can dedicate no more than 40% of your space to cages (the rest will be for supplies, visitors, and general comfort). A cage for two guinea pigs should be 10.5 square feet at a minimum.

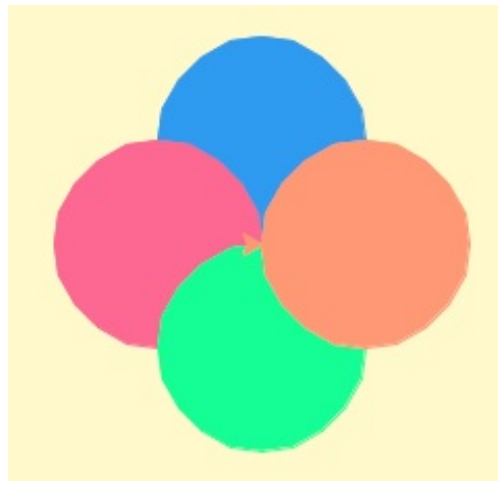
If you stick to the minimum cage size, what is the maximum number of guinea pigs you can house? Report two numbers, the number you can house if you put one guinea pig per cage, and the number if you have two guinea pigs per cage. Be sure to report the final numbers as *integers*, since you can't house a fraction of a guinea pig. Round down using the `math.floor` function from the `math` module.

You will write a program to do these calculations. You want your program to be general, in case you rent a different building in the future, or you find out that really guinea pigs need more space, or that a different percentage of the space is needed for cages versus other needs. Therefore your program should be as general as possible.

The statements below are in the `hw1Starter.py` file. Add statements to calculate the two results you need. Use a call to `print` to display the results, and include string elements to make it more readable.

```
buildingSize = 800    # square feet, size of building
cagePercent = 0.40    # percentage dedicated to cages
cageSize = 10.5       # square feet, minimum size of cage
```

2. (10 pts) I have provided in the file `hw1Starter.py` a script using the `turtle` module. The script is supposed to cause the turtle to draw four overlapping circles like the picture shown below. Note the ending position and direction of the turtle; it should be exactly like this when it is done.



However, the script has two bugs in it. One bug is just a single line; to fix it you must modify just that one line. Put a comment at the end of its line describing the bug, and then fix it!

The second bug is a systematic one; the same bug repeated multiple times as the same context repeats. Find this bug, mark all the lines on which it occurs, along with a description of what was wrong, and then fix it!

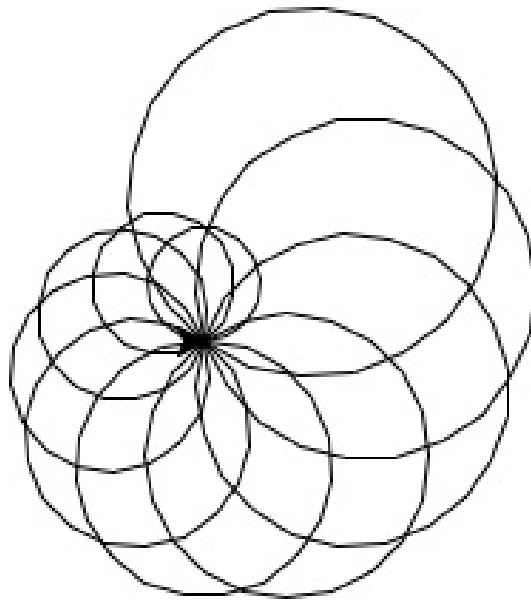
3. (10 pts) Write a script (adding it to the earlier file) that uses the `turtle` module. This question is open-ended, use your creativity! Your script should draw a large tree filling most of the window. Use different colors and shapes, and make parts of the tree filled in. You can make this simple like a child's drawing, or more complex. You can draw an evergreen, a deciduous tree, or a bare-branched winter tree. The tree should have at least 4 different components, and 3 different colors, but feel free to use as many as you are inspired to do.

Be sure to include a call to `exitonclick` at the end of this script to clear away this window.

4. (10 pts) Write a new script (adding it to the starter file) that uses the `turtle` module. Your new script should draw the picture shown below.

Use a `for` loop that loops over circle radii (radiuses?). The radii should range from 20 through 65, going up by 5 each time (20, 25, 30, ..., 60, 65). For each radius value, ask each turtle to draw a circle with that radius.

Here is what the final result should look like:



Once you get that working, add another `for` loop outside the first one, encapsulating the first one inside of it. The outer `for` loop should look like the following:

```
for pos in [[-225, 225], [-225, -225], [225, -225], [225, 225]]:  
    <your circles code plus a little here>
```

Inside the outer loop, use the `up`, `down`, and `goto` methods of the turtle to move the turtle to position `pos`. Then include the `for` loop you created first. The result should be a script that draws four copies of the circles in its window.

Be sure to include a call to `exitonclick` at the end of this script to clear away this window.