CIS 122 Spring 2016
Project 0: Hello, Python
Due: Monday, April 4, 6p

Goals

By the end of this assignment, you should

- be able to open IDLE, a Python integrated development environment, and work with the Python Editor and Python Shell;
- understand how Python evaluates simple expressions
- be familiar with Python’s numeric data types int and float and numeric operators
- understand Python assignment statements
- be able to write and comment simple Python code to solve a problem

Grading

This project will be marked out of 8 points, with points given for both problem solving and Python code. A grading rubric is given at the end of this project specification.

Getting Started

For this project you will be using the IDLE Python development environment. Features of IDLE, in particular, the Python Shell and Python Editor, will be introduced in lab this week.

For this project, you will enter and save Python code and other responses to the exercises given below in the Python Editor window. In IDLE, open a new Editor window. Add a comment at the top of the Editor window with the project/problem name, author name, credits to sources of any significant help, and a short description of the file. For example:

```
CIS 122 Spring 2016 Problem 0

Author: Diligent U. Student

Credit: Worked with partner, Rigorous S. Oregon

Description: Introduction to programming problem set uses Python numeric data types and operations to solve a variety of small problems.
```

Everything between the triple-quotes is treated as a comment, i.e., ignored, by Python. Python will also ignore everything that follows the '#' symbol on one line. For example:

```
# A comment like this should be used to briefly document each individual problem.
```

Learning to write comments well – succinct and informative, for yourself and other human users of your code – is an important programming skill.

After typing the header comment, save the file in the Editor window with the name cis122project0.py. (The .py suffix may be automatically added – check on your computer.)
Now add additional code to this file to solve each problem given below.

Be sure to read the project specification closely and carefully. This may require multiple readings. *A thorough understanding of the project specification is a very important aspect of good programming.*

Save and test your code for each individual problem; do not try to do this all at once at the end. *Working with code in manageable, solvable chunks is a very important aspect of good programming.*

(0) You wish to buy a t-shirt as a gift for each of 15 people. Green t-shirts cost $25 each and yellow t-shirts cost $15 each. Add and complete the following lines of Python code to a Python file, cis122project0.py, to determine the total amount you must spend when four of the people get green t-shirts.

```python
ttl_shirts =
ttl_green =
ttl_yellow =
cost_green =
cost_yellow =
ttl_cost =
```

Then add one more line of code: `print(ttl_cost)`.

To run the code, save the Editor window to a file named cis122project0.py. From the Run menu, choose Run Module. The `print` output will be displayed in the Python Shell.

What happens if the `print(ttl_cost)` line of code is not included in the Python file? Run the code with this line commented out, and then answer the question in a commented line in your file.

You may want to include one more line of code: `print()` will print a blank line in the Python Shell. This will make your results more readable in the Shell window when you add more code to your file.

(1) The volume of a sphere with radius $r$ is $\frac{4}{3}\pi r^3$. Add code to your Project 0 file to determine and print the volume of a sphere with radius 1 and radius 5. (You can use 3.14159 as the value of $\pi$.)

(2) "525,600 minutes" is a reference to the number of minutes in a year in the well-known song "Seasons of Love", from the Broadway musical *Rent*.

Write Python code that will confirm that the number in the song is correct.

(3) You are offered a choice: you can take $1$ million dollars now, or, you can start with $1$ cent and double it each day for a month. At the end of the month, you will be given the resulting amount. Write Python code to help you make a decision.

A good way to start is to look at simple cases, and try to detect a pattern.
day1_amt = .01
day2_amt = day1_amt * 2
day3_amt = day2_amt * 2

This is straightforward and clear, but tedious. Consider an alternative, which is to look for a pattern that can be exploited in the solution:

day1_amt = .01
day2_amt = d1_amt * 2
day3_amt = d1_amt * 2 * 2
day4_amt = d1_amt * 2 * 2 * 2

Do you see the pattern that allows you to write the Python code that will help you make the decision without having to write a line of code for each day of the month?

Execute your code to determine your answer. Include your answer to the offer as a comment in your Python file.

Challenge (extra credit): What if the offer were for an immediate $10 million? Include your answer as a comment in the Editor window.

**Finishing & submitting your work**

When you have completed all of the problems, use the Save command from the IDLE File menu to save the Editor window as a file with the name cis122project0.py.

Submit your file via Canvas. You may re-submit your project up until the project deadline (i.e., as long as the submission link is available). Only the final submission will be graded.

Note: as per CIS 122 class policy (see Syllabus), it is not possible to submit a project after the deadline. Projects that are not submitted by the deadline will receive a default grade of zero. Two project grades will be dropped at the end of the term, to provide the flexibility you need for busy weeks, individual technical difficulties, misunderstandings, etc. You do not need to contact the CIS 122 instructional staff about this; it will happen automatically.

Even if you do not submit a project, you should complete the project and check your work against the posted solution.

**Grading Rubric**

8 points possible (2 pts. per problem):

1: Python program has correct structure and good style, including a comment with the project identification, author, credits, and short file description at the top of the program file; a comment at the beginning of each individual problem; code is easy to read, e.g., includes good use of white space around operators.

1: Code correctly implements the project specification.