This problem set is due on **Thursday, October 1 at 11:55pm**.

- This problem set assumes that you have installed a Python programming environment such as Canopy on your computer.
- Once you have installed Canopy, you can use it to create programs using the editor that comes with Canopy or use your favorite text editor.
- Your program should run without syntax errors. Please do not submit programs that have syntax errors! It is better to submit a partial implementation that runs as opposed to a full implementation that does not even run.
- Add your name and email address as a comment at the top of each file you submit.

Submit the following file(s) to the Assignments area on Sakai as an attachment. Multiple submissions are allowed before the due date. Please do not submit other files that I did not ask for.

`ps2.py`

**Naming Conventions In Python And Programming Style In General**

Please use these conventions as you establish your programming style. Programming professionals use them, too.

- **Names**: Choose informative names, e.g., `hourlyRate` rather than `hr` if it is to be a variable name. If you like `hourly_rate` better than `hourlyRate`, use that form. Whichever style you choose, stay with that style consistently.

- **Variable and function names**: We start a variable or function name with a lower-case letter, e.g., `count`, `hourlyRate`, etc.

- **Use of white space**: Adopt a good indentation style using white spaces and be consistent throughout to make your program more readable.

**Your Solution Format**

See `sample.py` in the [given] folder and put your solution in that format. It is a good format to use and it will make the grader’s life much easier. Please run `sample.py` on your computer to see how it works first.
Problem 1

If you have not tried sample.py yet, do that before you continue, PLEASE!

In a file named ps2.py define a function named myFavoritePoem (or my_favorite_poem) that takes no parameter and prints a favorite poem of yours. Inside the function you will have a series of print statements each of which will write a line of the poem. Your poem must have at least five lines.

This function does NOT return any useful value, thus called a void function. As such, there should be no return statement in this function.

Now that the function is defined, call the function once. By calling the function, your program will now generate the poem expected. Include that call in the main function in the form suggested in sample.py.

You will hand in ps2.py which will contain your solutions for all the problems in this problem set.

Problem 2

In the same file that you created for Problem 1 above (ps2.py), define another function named celsius_to_fahrenheit (or celsiusToFahrenheit) that takes a Celsius temperature value as a parameter and converts it into Fahrenheit and returns the computed value.

This function does return a useful value, thus called a fruitful function. Therefore, there must be a return statement in this function.

Now that the function is defined, call it three times as follows:

    print celsius_to_fahrenheit(100)
    print celsius_to_fahrenheit(0)
    print celsius_to_fahrenheit(-40)

Include these calls in the main function.

Problem 3

In ps2.py define a function named milesToKilos (or miles_to_kilos) that takes one parameter named numMiles (the number of miles to be converted) and converts it into number of kilometers. Inside the function do the following:

- Declare a variable named conversionFactor that is initialized with an appropriate value.
- You should then perform the conversion from miles to kilometers and return the computed value. This is a fruitful function.
- Now, in main call the function and print the computed value to the standard output divide, namely the computer screen.

Include at least two calls to this function in main.

Problem 4

In ps2.py define a function named sphereVolume (or sphere_volume) that accepts a radius value as an argument and returns (not prints) the volume of a sphere with that radius. For example,

- sphereVolume(2.0) should return 33.510321638291124 or a number close to it.
- sphereVolume(5.0) should return 523.5987755982989 or a number close to it.
For PI, use the one from the math module. Include the example calls I included above in the main of ps2.py and print the computed value to the screen.

This time define a function named cylinderSurfaceArea (or cylinder_surface_area) that accepts a radius and height as parameters and returns the surface area of a cylinder with those dimensions. For example,

- \texttt{cylinderSurfaceArea(3.0, 4.5)} should return 141.3716694115407 or a number close to it.
- \texttt{cylinderSurfaceArea(4.5, 6.0)} should return 296.8805057642354 or a number close to it.

Again include two calls in main and print the results.

**Problem 5**

In ps2.py define another function named dollars_to_bills (or dollarsToBills) that takes \texttt{n_dollars} as a parameter and converts it into the number of twenty dollar bills, the number of ten dollar bills, the number of five dollar bills, and the number of one dollar bills. It will print the result in the following format to the computer screen.

\texttt{99 dollars is equal to}
\texttt{4 twenties,}
\texttt{1 tens,}
\texttt{1 fives, and}
\texttt{4 ones.}

if the function was called with its actual argument value of 99.

In doing this conversion, you should first maximize the number of twenty dollar bills, then maximize the number of ten dollar bills, etc. Inside the function, declare a variable named \texttt{n_twenties} to hold the number of twenty dollar bills, \texttt{n_tens} to hold the number of ten dollar bills, \texttt{n_fives} to hold the number of five dollar bills, and \texttt{n_ones} to hold the number of one dollar bills. You may use other variables as needed. This is a \textit{void function}.

Now include at least two calls to this function in main as follows:

\texttt{dollars_to_bills(99)}
\texttt{dollars_to_bills(293)}

Now that you have used two different styles of naming functions and variables, you probably have your preference by now. Stay with whichever style you prefer from now on.

You have also used both void functions and fruitful functions and use them appropriately in the future.

Hand in ps2.py.