# 2 — PHY 494: Homework assignment (62 points total)

Due Sunday, Jan 28, 2018, 11:59pm.
Submit a PDF through Blackboard (name it `lastname_firstname_hw2.pdf`). Homeworks must be legible or may otherwise be returned ungraded with 0 points.

This assignment contains **bonus problems**. A bonus problem is optional. If you do it you get additional points that count towards this homework's total, although you can't get more than the maximum number of points. If you don't do it you can still get full points. Bonus problems and bonus points are indicated with an asterisk "*".

Note: In general, for full credit you need to (1) show the commands that you used and (2) answer the question. Sometimes you should also copy and paste output.

## 2.1 Python data types (7 points)

What is the Python data type of each of the following values?

(a) `3.14515` [**1 points**]

(b) `0` [**1 points**]

(c) `False` [**1 points**]

(d) `'To be or not to be'` [**1 points**]

(e) `[3, 2, 1, "lift off!"]` [**1 points**]

(f) `3, 2, 1, "lift off!"` [**1 points**]

(g) `None` [**1 points**]

(h) Bonus: `{'name': 'Hamlet', 'occupation': 'Prince'}` [**bonus +1***]

## 2.2 Operators (8 points)

(a) What does the following code output? What are the names of the three mathematical operations that are carried out in lines 4–6? [**4 points**]

```
1     a = 5.0
2     b = 10
3
4     result_1 = a + b
5     result_2 = a / b
6     result_3 = a ** b
7
8     print(result_1, result_2, result_3)
```

(b) What error does the following code produce and why? [**4 points**]

```
name = 'Hamlet, Prince of Denmark'
age = 30
print(name + age)
```

## 2.3 Python Lists and Strings (20 points)

Lists and strings share some similarities but also have important differences. Let's look at them. (Type code in the Python interpreter e.g., `ipython`).

```
bag = ["guide", "towel", "tea", 42]
ga = "Four score and seven years ago"
```

(Note that spaces are shown explicitly in the second string with the symbol "␣"—just type a space.)

(a) How do you have to slice `bag` in order to get `['towel', 'tea']`? [**1 points**]

(b) What does `bag[::-1]` do?

How do you slice bag in order to get `['tea', 'towel']`? [**2 points**]

(c) Strings can also be sliced. How do you have to slice `ga` to get

- "Four"
- "seven"

[**2 points**]

(d) You can access elements of a list in a variety of ways:

(i) Explain what

```
bag[0] = 'book'
```

does? (Hint: print `bag`!) [**1 points**]

(ii) Create two new variables:

```
mybag = bag
yourbag = bag[:]
```

and use them:

```
mybag[3] = "mice"
yourbag.append("money")
```

What is the content of `bag`, `mybag`, `yourbag`? [**2 points**]

(iii) From your observations, explain how the assignment `x = a` differs from `y = a[:]`? [**3 points**]

(e) Try

```
ga[:4] = "Three"
```

(i) Describe what happens?[1] [**1 points**]

(ii) How would you construct the string `"Three score and seven years ago"` from `ga` and the string `"Three"`? [**1 points**]

(f) What do the commands

```
ga.split()
a, b, c = ga.split()[:3]
list([1,2,3])
list(ga)
```

do? You can show the output but you need to explain in your own words what is happening. [**4 points**]

(g) Nested lists: Given the list

```
bags = [['salt', 'pepper'], ['pen', 'eraser', 'ruler']]
```

how do you have to index `bags` to get

(i) `['salt', 'pepper']` [**1 points**]

(ii) `'pepper'` [**1 points**]

(iii) `'ruler'` [**1 points**]

### 2.4 Very Simple Temperature Calculator (13 points)

Write a Python program `addtemperatures.py` that adds a temperature difference in Fahrenheit, $\Delta\theta$, to an absolute temperature, given in Kelvin, $T$. The program should

- ask the user for two floating point numbers $T$ (absolute temperature in Kelvin) and $\Delta\theta$ (temperature difference in degrees Fahrenheit) as input
- print the sum "$T + \Delta\theta$" in units of Kelvin (where $\Delta\theta$ must be converted to Kelvin)

The conversion of an absolute temperature from Fahrenheit to Kelvin is (written with numbers $T/\mathrm{K}$ and $\theta/{}^\circ\mathrm{F}$)[2]

$$T/\mathrm{K} = \frac{5}{9}(\theta/{}^\circ\mathrm{F} - 32) + 273.15 \tag{1}$$

[1] Note that strings are "immutable" objects in Python whereas lists are "mutable".

[2] We use "symbol/unit" to indicate a number without the unit so that we can write equations where all units correctly balance. For instance, if $T = 373$ K then $T/\mathrm{K}$ is the number 373. This approach is more precise than just saying "Take $T$ as the temperature in Kelvin and $\theta$ in Fahrenheit."

(a) Bonus: Derive an expression for $\Delta T = T_2 - T_1$ as a function of $\Delta\theta = \theta_2 - \theta_1$ (where $T_2$, $T_1$, $\theta_2$, and $\theta_1$ are arbitrary and only introduced to make the connection to Eq. 1). Show that the the difference in Kelvin is 5/9-th of the difference in Fahrenheit, [**bonus +3***]

$$\Delta T/\mathrm{K} = \frac{5}{9}\Delta\theta/^\circ\mathrm{F}. \tag{2}$$

(b) Use your result from the previous problem (i.e., Eq. 2) to derive a mathematical expression to compute the sum of $T$ and $\Delta\theta$. [**2 points**]

(c) Write the `addtemperatures.py` program and copy and paste the code. [**7 points**]

(d) Show the complete input and output (copy and paste) for the input $\Delta\theta = 63^\circ\mathrm{F}$ and $T = 265\,\mathrm{K}$. [**4 points**]

### 2.5 Loops (14 points)

(a) Use a `for` loop to print each element of the list

```
sentence = ["We", "must", "walk", "before", "we", "can", "run"]
```

Show your code and your output. [**4 points**]

(b) Bonus: Find a way to only print every other line from `sentence`, i.e., the output should read “We walk we run”. Show your code and your output. [**bonus +2***]

(c) Use a loop to sum the integers from 1 to 1000 and print the final result (500500). [**4 points**]

*Note*: You can increment a variable `total` with a value `x` with the assignment

```
total = total + x
```

or equivalently but more compactly written

```
total += x
```

Both expressions do the same thing: add two values together and then assign the sum to the variable `total`, overwriting the old value of `total` in the process.

(d) Write a program that counts down from 10 to 0 and prints each number (“10 9 8 ... 3 2 1”).

(i) Use a `while` loop. [**3 points**]

(ii) Use a `for` loop. [**3 points**]

(You can use any other Python functions that you might need, such as `range`.)

Show your code and output.