

Objectives.

1. Prepare for the course: sign up for Piazza and Gradescope; apply for a CS account; setup the programming environment and get familiar with it.
2. Write simple straight-line Python programs that involve: command-line input; the four basic built-in types of data (`str`, `int`, `float`, and `bool`); and calls to library functions.

Problem 1. (*Preparing for the Course*) Take care of the following action items (see course website [↗](#) for details):

- Sign up for Piazza using your UMass Boston email.
- Sign up for Gradescope using your UMass Boston email.
- Apply for a CS account.
- Setup the programming environment and familiarize yourself with it.

Edit the Python program `course_prep.py` by replacing the placeholders `[Name]`, `[UMass Boston Email]`, and `[CS Account Username]` with relevant information and test the program by running the following command on the terminal:

```
$ python3 course_prep.py
I acknowledge that I have fully read and understood the contents of the course
website. I understand that if I have any questions or concerns about the course
mechanics, it is my responsibility to discuss them with the instructor.

Jane Doe
jane.doe@umb.edu
jdoe
```

Problem 2. (*Name and Age*) Write a program `name_age.py` that takes two strings *name* and *age* as command-line arguments and writes the output “*name* is *age* years old.”.

```
$ python3 name_age.py Alice 19
Alice is 19 years old.
```

Problem 3. (*Greet Three*) Write a program `greet_three.py` that takes three strings *name1*, *name2*, and *name3* as command-line arguments and writes the output “Hi *name3*, *name2*, and *name1*.”.

```
$ python3 greet_three.py Alice Bob Carol
Hi Carol, Bob, and Alice.
```

Problem 4. (*Triangle Inequality*) Write a program `triangle.py` that takes three integers as command-line arguments and writes `True` if each one of them is less than or equal to the sum of the other two and `False` otherwise. Note: this computation tests whether the three numbers could be the lengths of the sides of some triangle.

```
$ python3 triangle.py 3 4 5
True
$ python3 triangle.py 2 4 7
False
```

Problem 5. (*Trigonometric Functions*) Write a program `trig_functions.py` that takes a float *t* (angle in degrees) as command-line argument and writes the value of $\sin(2t) + \sin(3t)$.

```
$ python3 trig_functions.py 60
0.8660254037844388
```

Problem 6. (*Displacement*) Write a program `displacement.py` that takes three floats x_0 , v_0 , and t as command-line arguments and writes the value of $x_0 + v_0 t - gt^2/2$, where g is the constant 9.78033 meters per second per second. Note: this value is the displacement in meters after t seconds when an object is thrown straight up from initial height x_0 meters with velocity v_0 meters per second.

```
$ python3 displacement.py 10 0 1
5.109835
```

Files to Submit

1. `course_prep.py`
2. `name_age.py`
3. `greet_three.py`
4. `triangle.py`
5. `trig_functions.py`
6. `displacement.py`

Before you submit:

1. Make sure your programs meet the input and output specifications by running the following command on the terminal:

```
$ python3 run_tests.py -v [<problems>]
```

where the optional argument `<problems>` lists the problems (`Problem1`, `Problem2`, etc.) you want to test, separated by spaces; all the problems are tested if no argument is given.

2. Make sure your programs meet the style requirements by running the following command on the terminal:

```
$ pycodestyle <program>
```

where `<program>` is the `.py` file whose style you want to check.