Program cs01
Fall 2022: Updated September 23, 2022

CS 5245: Data Programming
R. M. Parry

Read chapters 1-3 in Fundamentals of Python Programming and complete the following exercises.

Complete computer setup by following the directions here:
http://cs.appstate.edu/~rmp/cs5245/setup.pdf

Create a folder called `cs01` to store your files for this lab.

Complete the following activities using the **exact variable names** and save the files using the **exact file names** as specified.

To test your programs, run them from IDLE and check their output.

1 assignments

In IDLE create a new file in your `cs01` folder and save it with *exactly* the following name: `assignments.py`.

Practice using variables, values, expressions, and assignment statements by completing the following in order. When you are asked to print the value of a variable, you should use the same format as the examples below. For example, if you are asked to print the variable x that has the value 5, you should use something like the following:

```
print(f'x = {x}')
```

When you run your script, you should see the following in the Python Shell window:

```
x = 5
```

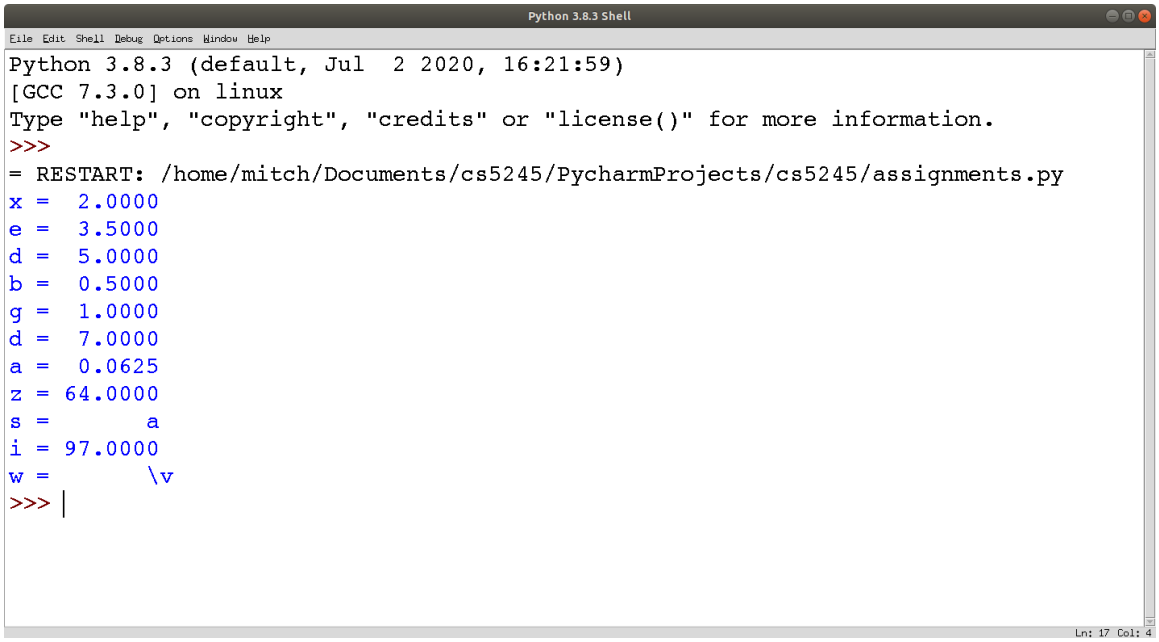
1. Assign the value 2 to variable x and print its value.
2. Assign the value 3 to the variable y.
3. Assign the value 2.5 to the variable b.
4. Assign the value 3.5 to the variable e, printing e.
5. Assign a value of $2b$ to the variable d, printing d.
6. Assign a value of $3x$ to the variable f.
7. Assign the value 6 to a.
8. Assign the value of the expression $a + 2$ to the variable a.
9. Assign the value of the expression $b - 2$ to variable b, printing b.
10. Assign a value $2e$ to the variable d.
11. Assign the value of the expression $4b - 1$ to the variable g, printing g.
12. Print d.
13. Assign the value of the expression $\frac{b}{a}$ to variable a, printing a.
14. Assign the value 4 to k.

15. Assign the value of the expression $\frac{k}{a}$ to the variable z, printing z.

16. Assign the character 'a' to the variable s, printing s.

17. Look up the Python functions `ord` and `chr` and complete the following:
    Assign the numeric representation of variable s to the variable i, printing i.
    Assign the character corresponding to number 11 to the variable w, printing w.

Every number that gets printed should be formatted like this: `XX.XXXX`, so two digits to the left of the decimal and four to the right. The strings should have a minimum width of 7 characters and right-justified. When your program prints correctly, it should look like this in the Python Shell window:



Printing the ASCII character 11 results in a vertical tab. In IDLE it looks like '\v' but in the Python shell it gets interpreted as a taller newline character.

2 convert_temp

Create a new Python file and save it in your `cs01` folder with the name `convert_temp.py`. Complete the following:

1. Prompt the user for `input` asking, "Enter the temperature in Fahrenheit: ".

2. The temperature may be entered as an integer or decimal.

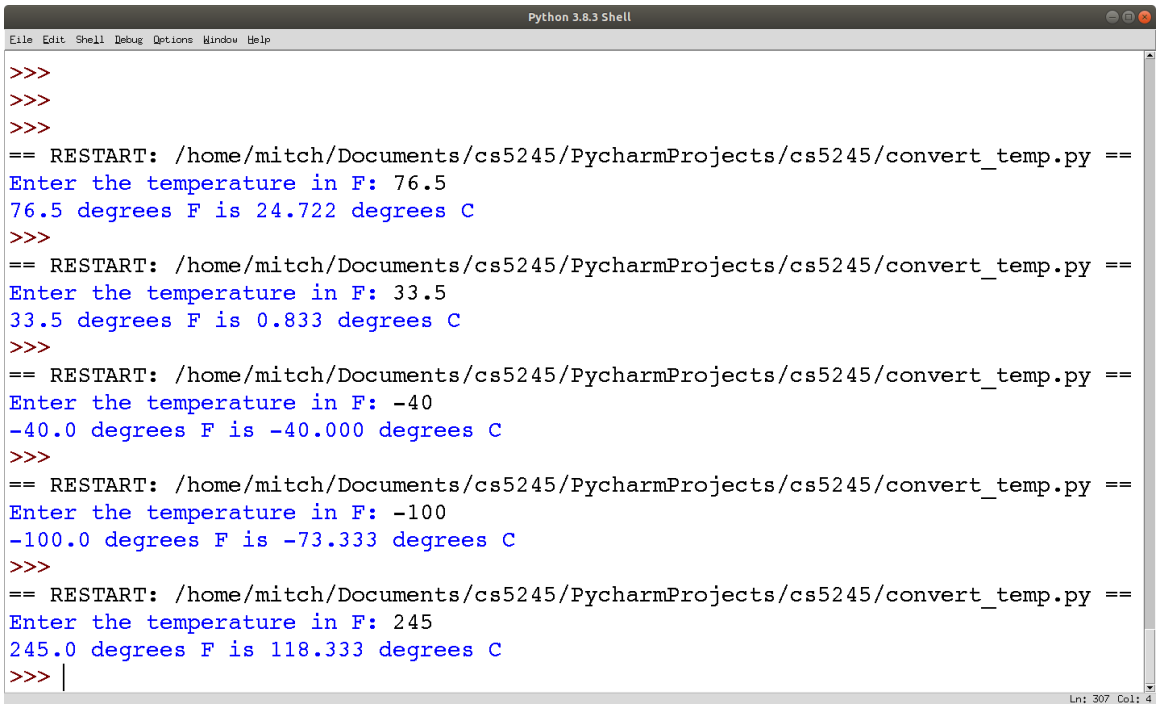
3. Store the temperature in Fahrenheit in a variable called `f`.

4. Convert the temperature to Celsius using the following equation:

$$c = \frac{5}{9}(f - 32)$$

5. Store the temperature in Fahrenheit in a variable called `c`.

6. Print the original temperature in Fahrenheit and Celsius to match the example below:



Notice the degrees Fahrenheit are displayed exactly as they are entered by the user and that exactly of the 3 decimal places are shown for the degrees Celsius.

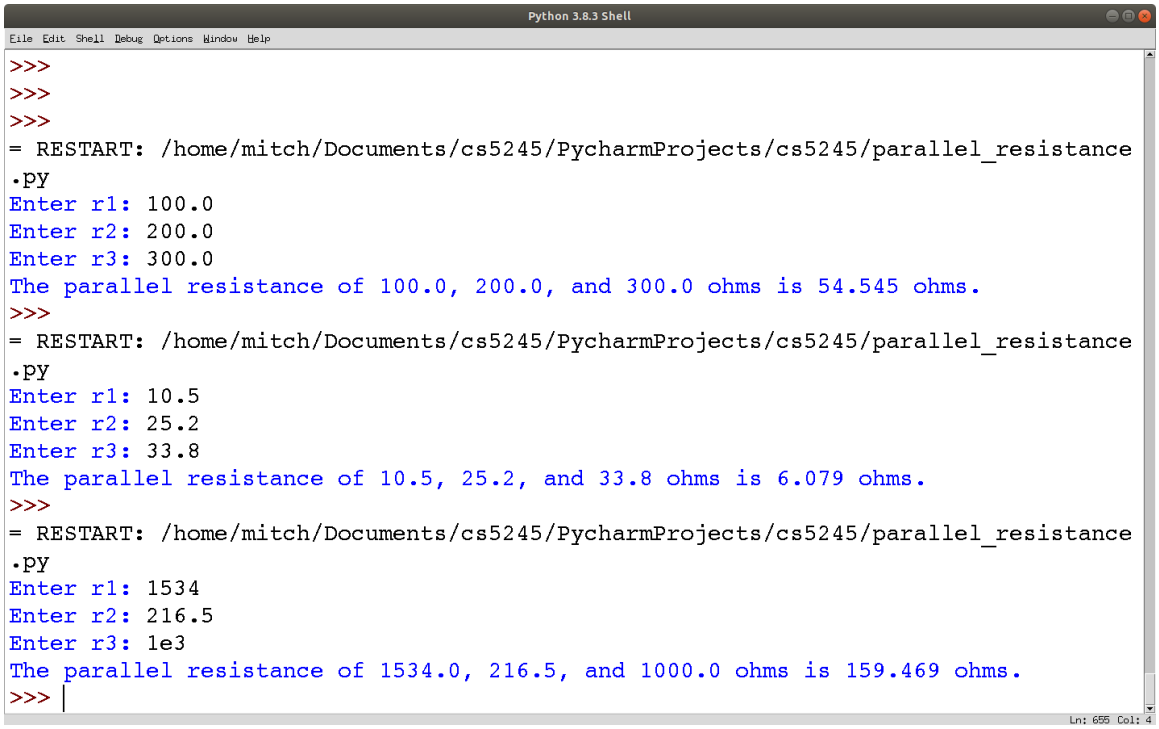
3 parallel_resistance

Create a new Python file and save it in your `cs01` folder with the name `parallel_resistance.py`. Complete the following:

1. The combined resistance of three resistors connected in parallel is given by the following equation:

$$r_t = \frac{1}{\frac{1}{r_1} + \frac{1}{r_2} + \frac{1}{r_3}}$$

2. Prompt the user for r_1, r_2, and r_3 as `input`.
3. Store the resistances in variables named `r1`, `r2`, and `r3`, respectively.
4. Compute the combined resistance and store it in a variable named `rt`.
5. Finally, print the result to match the example below:



Again, notice that the inputs are displayed exactly how Python would print them (by default) and exactly 3 of the decimal places are shown for the computed parallel resistance.

4 `lorentz`

Create a new Python file and save it in your `cs01` folder with the name `lorentz.py`. Complete the following:

1. In special relativity, the Lorentz factor is a number that describes the effect of speed of an object on various physical properties when the speed is significant relative to the speed of light. Mathematically, the Lorentz factor is given by the following equation:

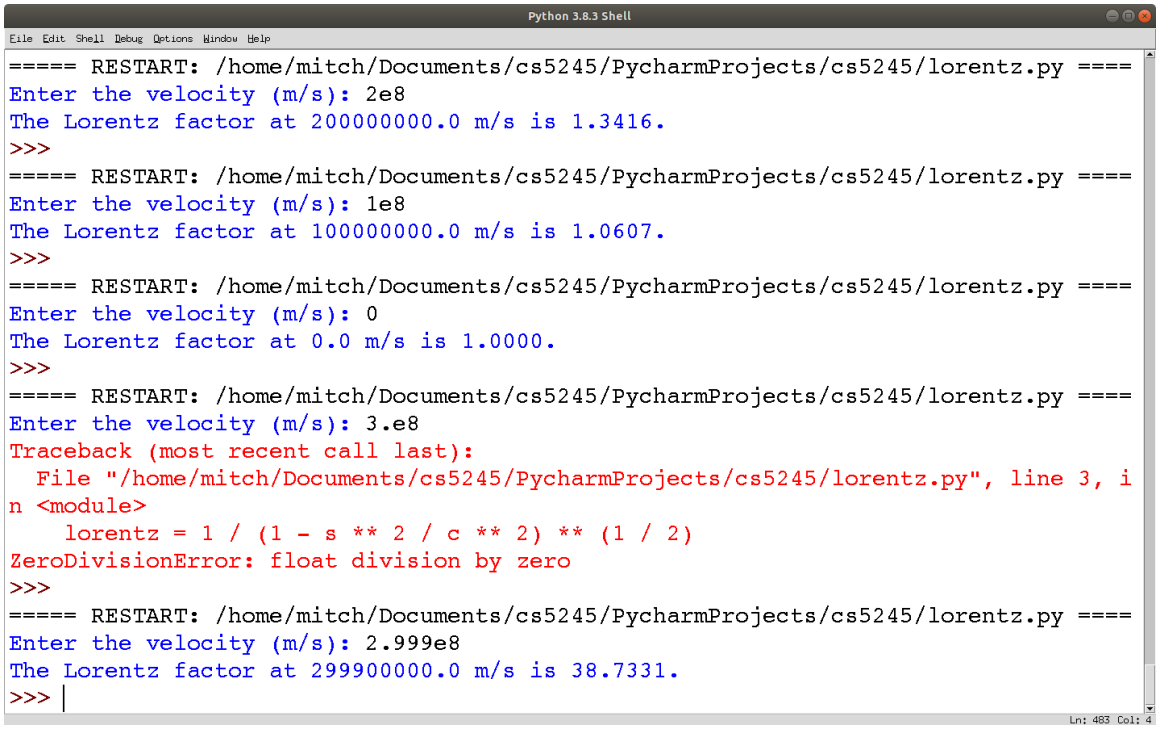
$$\gamma = \frac{1}{\sqrt{1 - \frac{s^2}{c^2}}},$$

   where $c = 3 \times 10^8 m/s$ is the speed of light and s is the positive speed of the object: $0 < s < c$

2. Prompt the user for s and store it in a variable named `s`.

3. Compute γ and store it in a variable called `lorentz`.

4. Print the results to match the following:



Again the input is displayed in Python's default way and exactly four decimal places are used for the computed Lorentz factor.

Submit to Web-CAT to compute your score!

1. Create a ZIP file for your `cs01` folder by right-clicking the folder and selecting:

   - **Send to** → **Compressed (zipped) folder** on Windows
   - **Compress Items** on MacOS
   - **Compress** on Linux

   You should find the new ZIP file in the same directory where `cs01` resides.

2. Login to http://webcatvm.cs.appstate.edu:8080/Web-CAT and submit your ZIP file for grading. You may submit as many times as you want before the deadline.