

1. What are the advantages of using Python *interactively*? What are the advantages of using Python to *execute scripts*?
  
  
  
  
  
  
  
  
  
  
2. Comment statements cannot be understood by Python. Why, then, is it useful to write comments?

Consider the following program, which takes as input a temperature in Fahrenheit and outputs the equivalent temperature in Celsius.

```
t = float(input('Enter the temperature in Fahrenheit: '))
t = (5/9) * (t - 32)
print('The equivalent temperature in Celsius is', t)
```

3. (a) In *reading* the program above, is it *clearer* to use a single variable for multiple purposes (i.e., how `temp` is used) or to have more than one variable each with a dedicated purpose? Explain with examples from the code.
  
  
  
  
  
  
  
  
  
  
- (b) In the program above, are the parentheses necessary to perform the conversion from Fahrenheit to Celsius? Explain what would happen if they were omitted.

4. Given a choice, would it be *clearer* to have variable names be short and easy to type, or longer and more descriptive? Explain using examples from the Fahrenheit-to-Celsius code from Problem 3.

5. Write expressions that:

(a) Compute the (integer) quotient when an integer *a* is divided by an integer *b*? (e.g. *a*=7, *b*=3, results in 2)

(b) Compute the (integer) remainder when an integer *a* is divided by an integer *b*? (e.g. *a*=7, *b*=3, results in 1)

(c) Exchange the values of variables *i* and *j*.

6. Consider the two function definitions below.

```
def emoji(eye, nose):
    return eye + nose + eye
```

```
def manyEmoji(num, eye, nose):
    print((emoji(eye, nose) + ' ') * num)
```

In the table below, fill in the blanks with **what is printed** (if nothing is printed, put down N/A) and **what is returned** for each function call given in the first column. <sup>1</sup>

Function call	Printed	Returned
<code>emoji('*', '.')</code>		
<code>manyEmoji(2, 'o', '&lt;')</code>		
<code>print(manyEmoji(2, 'o', '&lt;'))</code>		

\*

---

<sup>1</sup>You may type out and test these functions on a computer if you'd like, but make sure to understand the underlying logic!