

Lecture 10 - Exercises

10A - Defining Functions

1. Which of the following is/are true of the `print` function?
 1. It does not take inputs (arguments)
 2. It does not return a value
 3. It does not have any effects
 4. Using it requires understanding the code that implements
2. You've seen the `len` function a couple times now. Describe its behavior in terms of its *inputs*, *effects*, and *return value*.
 - Input(s):
 - Effects:
 - Return value:
3. Which of the following programs will **not** cause an error when run?

```
def greet():  
    print("Hi there!")  
greet()
```

```
def greet():  
    print("Hi there!")
```

```
greet()  
def greet():  
    print("Hi there!")
```

```
def greet:  
    print("Hi there!")  
greet()
```

10B - Passing inputs to functions

4. Consider the following program:

```
def h2m(hours, minutes):  
    total_mins = 60 * hours + minutes  
    return total_mins  
  
arg1 = int(sys.argv[1])  
arg2 = int(sys.argv[2])  
  
print(h2m(arg1, arg2))
```

1. Suppose the program is run with two command line arguments and prints `80`. If the first command line argument was `1`, what must the second one have been?
 2. Suppose the program is run with two command line arguments and prints `80`. If you don't know anything about either command line argument, how many possible pairs of numbers could have been given as command line arguments?
5. Consider the following program:

```
def pnmr(n, r):
    print(n % r, end=" ")

size = 7
rad = 3
for num in range(0,size):
    pnmr(num, rad)
```

1. How many numbers does this program print?
2. How many times does the program print the number 1?

B02 Peer Review

Before working on the Problems below, your group should peer review each group member's submission for B02.py from Monday's class. The Problems for today are also the same Problems for Friday's class, so don't rush through the peer review process - you will have plenty of time to work on the Problems.

For **each group member**, complete the following steps:

1. Create a copy of the B02 Peer Review: TEMPLATE slide, and put the group member's name in place of TEMPLATE.
2. Paste your code in the box provided. Delete the instruction comments, if you haven't already, to make the code fit.
3. **As a group**, review and test the code. For each of the rubric items below, decide whether the submission should receive credit or not. Fill in the box to the right of each rubric item.
 1. Prints a message welcoming the user to Mount Baker Snowbank
 2. Gets the two inputs from the command line arguments
 3. For each week, the program prints a message reporting the current snowpack.
 4. The program prompts the user to enter a command (1 point); and the prompt is repeated until a valid command (s, m, or c) is entered (1 point)
 5. The "s" command works: prompts for a number and adds it to the current snowpack
 6. The "m" command works: prompts for a number and subtracts it from the current snowpack
 7. The "c" command works: snowpack remains unchanged and the program continues to the next week
 8. An avalanche halves the snowpack if it's greater than 100 inches after the week's event
 9. The final snowpack is printed in feet and inches.
4. Total up the score and enter it in the **Total Score** text box.
5. The author of the code should now decide whether they would like to work on fixing errors or adding missing features as a group in class, or individually outside fo class.

Problems

1. Write a function `parrot` that takes one string argument (`s`) and one integer argument (`n`) and prints that string `n` times with a space in between; at the end, the function should print a single newline. Examples:

```
>>> parrot("hello!", 3)
hello! hello! hello!
>>> parrot("click", 5)
click click click click click
>>>
```

2. Write a function `rectangle_area` that takes two numbers `height` and `width` and **returns** the area of a `height`-by-`width` rectangle. Recall that a $h \times w$ rectangle has area $h * w$.
3. Write a function `circle_area` that takes one number `radius` and **returns** the area of a circle with that radius. Recall that a circle with radius r has area πr^2 .
4. Write a function `get_float` that takes a string, prints that string, then prompts a user for a floating-point number and returns their input converted to a float.

```
>>> get_float("Enter a radius: ")
Enter a radius: 4
4.0
```

(Note: in the above, `4` is entered by the user, and `4.0` is printed on the Thonny shell in dark blue, showing the

