

15110 Summer 2018
Problem Set 2

Name: _____

Andrew ID: _____

Reading Assignment

Read pages 19-42 of chapter 2 of the book *Blown To Bits*.

Instructions

- Type or neatly write the answers to the following problems.
- Save or scan this file as a pdf and submit to Gradescope

Exercises (10 points total)

1. (2 pts) For each of the following Python expressions, show how they are evaluated in Python (think about the rules for operator precedence and associativity!) by evaluating one operation at a time and showing the reduced expression until you have a final answer. For example, if the Python expression is `3 * 5 + 5`, your answer would look like this:

```
3 * 5 + 5
15 + 5
20
```

HINT: You can (you should!) check your final answers with `python3`.

a. `7 + 10 / 4 * 8 - 8`

b. `58 // 2 ** 3 / 5`

c. `85 % 9 // 2 * 7`

d. `3 - 3 * 3**2 + 3**3`

2. (2 pts) The purpose of this question is to get you to think about various ways programs can go wrong.

For a mortgage with a principal amount of P dollars, an interest rate of r per year (expressed as a decimal fraction such as 0.04 for 4%) for n years, the final debt amount M including the principal can be calculated by the formula

$$M=P(1+r)^n.$$

The following Python function implements this equation to compute the final amount of a mortgage after a number of years, based on the principal and a given interest rate.

```
def final_amount(principal, interest_rate, years):  
    return principal * (1 + interest_rate)**years
```

- a. Suppose we use this function to compute the final value of a mortgage with an interest rate of 10% on a principal of \$1000 after three years (ignore the financial unreality of this!):

```
final_amount(1000, .1, 3)
```

Will we get an integer result or a floating point result? Why?

- b. Suppose we want to do the computation in part (a) but we call our function without the number of years:

```
final_amount(1000, .1)
```

Does the function use a default value for the years, or does Python complain about this function call? Explain.

- c. Suppose we want to do the computation in part (a) but we call our function with the arguments in the wrong order:

```
final_amount(3, .1, 1000)
```

Does Python report an error? Why or why not?

- d. Suppose we replace the `return` statement with a `print` statement as shown below:

```
def final_amount(principal, interest_rate, years):  
    print(principal * (1 + interest_rate)**years)
```

What value is stored in the variable `amount` if we execute the following instruction? Why?

```
amount = final_amount(1000, .1, 3)
```

3. (2 pts) For each of the following invalid Python expressions, evaluate them in `python3` and explain the errors that you see.

a. `return = "bye bye"`

b. `1 // 0`

c. `"15+19" / "cat**110"`

d. `import math`

e. `math.sqrt(-25)`

4. (2 pts) This question is meant to help you understand how the `return` statement works, and in particular, how it interacts with loops. If you investigate the question carefully, it will save you a world of trouble in upcoming programming assignments.

Consider the following Python function definition that uses a loop:

```
def mystery(k, n):
    answer = 1
    for i in range(n):
        answer = answer * k
        print(answer)
```

- a. What does this function display if we call it as follows:

```
mystery(4, 6)
```

- b. What does `mystery(k, n)` display, in general, if `k` and `n` are positive integers?

- c. Using `python3`, see what happens if we replace the `print` function in `mystery` with a `return` statement instead:

```
def mystery1(k, n):
    answer = 1
    for i in range(n):
        answer = answer * k
    return answer
```

Store the function in a file, then load it in `python3` and call it with different positive integers and observe the results. In general, if `k` and `n` are positive integers, what does `mystery1` return? What does this tell you about how the `return` statement works?

- d. Now again using `python3`, see what happens if we indent the `mystery1` function differently so that the `return` is aligned with `for` as seen below:

```
def mystery2(k, n):  
    answer = 1  
    for i in range(n):  
        answer = answer * k  
    return answer
```

In general, if k and n are positive integers, what mathematical function does `mystery2` return? What does this tell you about the behavior of the `return` statement in its new position?

5. (2 pts) Besides learning how computational devices work for us, we should understand what side effects our use of these devices can have. Based on your reading of Chapter 2 (pages 19-42) of *Blown To Bits*, answer the following questions about the digital data you generate and how all of this digital data affects your privacy.
- Explain the difference between what the authors call "digital footprints" and "digital fingerprints" and give an example of each. Your examples could come from the reading or from your own experience.
 - Explain in no more than two sentences two different ways that a user of a web site can be identified.