

Algorithms and Abstraction

1. As you learned in lecture, computer science boils down to two main ideas: algorithms and abstraction.
 - a. Identify/Define the difference between algorithms and abstraction, and provide an example for each.
 - b. What are the 5 key components of a good algorithm?
2. Trace the following algorithm with the input of 5 ($n=5$).

Algorithm:

```

Step 1: Input a number n
Step 2: Set variable x equal to 1
Step 3: Set variable y equal to 1
Step 5: Set variable count equal to 0
Step 6: If count is less than n do the following, otherwise skip to Step 7
    Step 6a: Print the string "Value: " + str(x)
    Step 6b: If count is equal to 5:
        Step 6b i: Print the string "Oops"
    Step 6c: Set variable temp equal to the sum of variables x and y
    Step 6d: Set variable x equal to variable y
    Step 6e: Set variable y equal to variable temp
    Step 6f: Set variable count equal to the value of count + 1
    Step 6g: Repeat step 6 with updated values
Step 7: Print the string "Result: " + str(x)
    
```

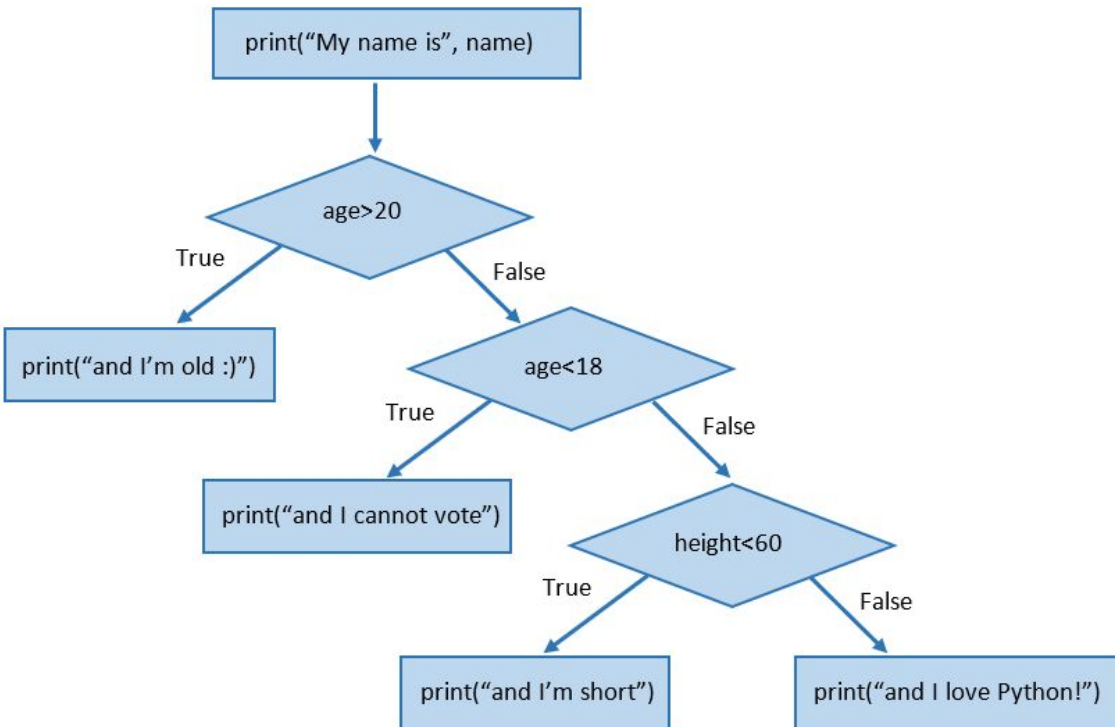
Answer the following questions:

- a. How many times do we run step 6 only? (be sure to ONLY count step 6 itself, not the sub steps of 6) In other words, how many times is the comparison in step 6 evaluated?
- b. How many times do we print "Oops"?
- c. How many lines are printed and what does each line read?
- d. Describe in words what this algorithm does.

Programming Basics

3. For each of the following Python expressions, identify the data type of the value it will evaluate to.
 - a. 3
 - b. 3.0
 - c. "3"
 - d. $3 < 3$
 - e. "3" + "3"
 - f. $3.0 * 3.0$
 - g. $3 / 3$
 - h. $3 // 3$

4. Answer the following questions about variables you could use to describe yourself.
- a. Suppose you are creating a few variables to describe yourself or another person. Name the data type that would **best** fit each variable description:
- i. Name
 - ii. Age
 - iii. Height (inches)
 - iv. Ability to vote
 - v. Address
 - vi. Zip code
 - vii. Weight
 - viii. School you attend
 - ix. Ability to speak more than one language
- b. Consider the following flow chart (which uses some of these variables), and convert it into equivalent Python code.



5. Explain the following error messages and provide one possible fix:

a.

```
Traceback (most recent call last):
  File "<tmp 1>", line 1, in <module>
    print("hello"+1)
TypeError: can only concatenate str (not "int") to str
```

b.

```
print("name",name)
          ^
SyntaxError: EOL while scanning string literal
```

c.

```
def f():
    print("hello 15-110")
    return 15-110
```

```
File "<tmp 1>", line 3
    return 15-110
    ^
IndentationError: unexpected indent
```

6. Evaluate the print statements and state what the value of y and x are at the end of the function.

```
x = 15+1
```

```
def f():
    y = 12
    print(y * 2)
    print(y)
    y = "apple"
    print("bananas, 15-110, and",y)
    print(x+3)

f()
```

Data Representation

7. Answer the following questions regarding number systems:
 - a. What is a number system?
 - b. How many values can a specific digit have in a decimal number? In a binary number?
 - c. Let's say that we are representing length in imperial units. You are given inches, feet, and yards. How could you represent 10,000 inches in this system (show your work)? (hint: 12 inches=foot, 3 feet=yard 1760 feet=mile)
8. Answer the following questions about binary addition
 - a. Add binary numbers 1011 and 1110 and be sure to show your work.
 - b. How many bits did you use to represent the sum? Could the result be represented in fewer bits?
9. Complete the following conversions and show appropriate work:
 - a. Convert 10111011 from binary to decimal
 - b. Convert 11111111 from binary to decimal
 - c. Convert 01101110 from binary to decimal
 - d. Convert 57 from decimal to 8-bit binary
 - e. Convert 63 from decimal to 8-bit binary
 - f. Convert 88 from decimal to 8-bit binary
10. The following questions relate to interpreting binary numbers as abstracted types
 - a. Answer the following questions relating to abstracting colors:
 - i. How many bytes are needed to represent an RGB value?
 - ii. What are the highest and lowest values for any color?
 - iii. Convert the color purple to an RGB value. Provide the values in decimal and binary.
 - iv. Convert 11010101 11110000 00110011 to a color (You can use this link to determine what color is appears to be:
https://www.w3schools.com/colors/colors_rgb.asp)
 - b. Answer the following questions relating to abstracting text (ASCII table:
<http://www.asciitable.com/>):
 - i. Convert 00111010 00101001 to ASCII characters.
 - ii. Convert 'CMU cmu' into 8-bit binary (neglect the quotation marks).

Functions

11. Consider the following function:

```
def helloThere(name):
    print("Hello there,", name, "!")
```

- What will you see in the shell if you call `print(helloThere("Stella"))` ?
- Does this function have side effect(s)? If so, state the side effect(s).
- Does this function have a returned value? If so, state the returned value.
- What do we call the statement `helloThere("Stella")` ? What do we call the string `"Stella"` in that statement?

12. Consider the following algorithm for a function:

Algorithm:

- Input a number `n`
- If `n` is greater than or equal to 0 do the following, otherwise skip to step 3:
 - If `n` is divisible by 10 then print out the value of `n` followed by " is cool number", otherwise skip to step 2b
 - If step 2a did not occur, check if `n` is divisible by 4, and if so print out the value of `n` followed by " is a strange number", otherwise skip to step 2c
 - Skip to step 4
- Print the value of `n` followed by " is a negative number!!"
- Print the string "ALL DONE"

Identify the input(s), returned value(s), and side effects of this algorithm. Then write the corresponding Python function called `divisionCheck`.

13. The function `fruitCalculator` has parameters `percentApples`, `totalFruit`, and `farm`. `fruitCalculator` calculates the number of apples harvested by the farm, prints the number of apples and the number of other fruit, and returns a string announcing the harvest of apples. For example, `fruitCalculator(0.75, 400, "Carnegie Farms")` would return "Carnegie Farms harvested 300 apples this year!", and the following would be printed in the console.
- ```
300 apples were harvested!
100 other fruits were also harvested!
```

Write the function `fruitCalculator` according to the description above.

14. Suppose we had the following segment of code:

```
course = "15110"
grade = 95.0
def randomFunction(x, y):
 sum = x + y
 difference = x - y
 print(course)
 return (sum + difference) / 2
```

Give the scopes, either local or global, of the following variables: course, grade, sum, x, y, and difference.

### How Python Works

15. Answer the following questions about how the interpreter converts Python understandable for the computer:

- 1) What is the process in which the interpreter breaks down a Python file into key words?
- 2) What is bytecode? In which process are sequenced tokens converted to bytecode?
- 3) What is the process by which the interpreter finds the correct sequence of tokens?

16. You have the following segment of bytecode:

```
LOAD_CONST 0
LOAD_CONST 1
BINARY_ADD
STORE_NAME 0
LOAD_CONST 1
LOAD_CONST 2
BINARY_MULTIPLY
STORE_NAME 1
LOAD_NAME 0
LOAD_NAME 1
BINARY_SUBTRACT
STORE_NAME 2
```

| Variables Table |      |       |
|-----------------|------|-------|
| ID              | Name | Value |
| 0               | x    |       |
| 1               | y    |       |
| 2               | z    |       |

| Constants Table |       |
|-----------------|-------|
| ID              | Value |
| 0               | 5     |
| 1               | 6     |
| 2               | 7     |

Translate the bytecode into python using the provided tables

17. The following function takes in two numbers a and b, prints them as “a, b” and then returns their sum (for example, given a=3 and b=7 the function will print “3, 7” and it will return 10). Find the errors and classify them (into one of the three broad types of errors).

```
def printValuesReturnSum(a, b)
 print(a, b)
 return "a" + b
```

18. The following lines of code contain multiple errors.

```
numerator = 0 #line 1
denominator = 6 #line 2
x = denominator/Numerator #line 3
print("x", x) #line 4
```

- Find the errors and categorize them into the three general types of errors discussed in class.
- If you run these lines of code in Pyzo, it will produce an error message. State which error in the code will cause the error message and explain your answer. In other words, which error is detected first and why? (You should be able to answer this without actually running the code)