COMP1005 Fundamentals of Programming

ComSSA Revision Worksheet

---

1 QUESTIONS

Interpreting example code

Explain what each of the following code blocks does.

```
# 1a
class PhoneBook:
    def __init__(self, name, addr, suburb, phone):
        self.name = name
        self.addr = addr
        self.suburb = suburb
        self.phone = phone

# 1b
entries = []

# 1c
with open("phonebook.csv") as f:
    lines = f.readlines()

# 1d
for line in lines:
    a, b, c, d = line.strip().split(",")
    entries.append(PhoneBook(a, b, c, d))
```

Some questions

1. What does `for i in range(5):` mean?
2. If we changed `i` above to `woohoo`, would it work? Why/why not?
3. Which of the following variable names are allowed in Python?
   (a) `av4tar`
   (b) `avatar1`
   (c) `4vatar`
   (d) `ava_tar`
   (e) `ava.tar`
   (f) `list1`
   (g) `list`
4. What is the difference between `for k in range(len(knightList)):` and `for k in knightList)):` ? Focus in particular on what k ends up meaning, and how you would refer to elements of knightList in the loop.
5. You have your functions in one .py file, and your calling code in another (For example, every prac test since PT2). In the calling code, should you have `import functions` or `from functions import *` ?

Spot the error

For each of the examples below, identify what the error is and how you would fix it. You can assume variables are of the indicated data type.

```
# Error 1
if number = 1:
    print("I am one")

# Error 2
if number < 0:
    print("Don't be so negative")
else number > 0:
    print("Accentuate the positive")

# Error 3
myString = thisString + number + thatString

# Error 4
for i in range(len(myList):
    print(i)

# Error 5
if word in ["check", "carefully"]
    print(word, "appears in this list.")

# Error 6 (2 errors)
class myClass:
    def myFunction(self, whatever):
        doStuff()

    print(myFunction(whatever))
```

Error messages

When Python gets stuck, you get a bunch of lines, with an error message and the line it occurred on usually appearing near the bottom.

Explain when each might come up and how to fix them:

- `IndentationError: expected an indented block`
- `IndentationError: unindent does not match any outer indentation level`
- `NameError: name 'x1' is not defined`
- `IndexError: list index out of range`
- `UnboundLocalError: local variable 'x' referenced before assignment`
- `AttributeError: 'module' object has no attribute 'x'`
- `SyntaxError: invalid syntax` (when the line it shows looks OK)
- `SyntaxError: unexpected EOF while parsing`
- `ImportError: No module named matplotlib.pyplot` (The only one here that isn't your fault)

List comprehensions

Find comprehensions that achieve the following:

- Finding all the multiples of 5 up to 50 (i.e. 5, 10, 15, ... , 45, 50)
- Multiplying all numbers in a list by 2. (Can use [1, −1, 5, 2] to test it)
- Returning all words in a list where the second letter is "e" or "E"
- Counting the number of items in a list whose value is 2016, 2019 or 2022.

2 SOLUTIONS

Interpreting example code

1a – creates a class called `PhoneBook` with class variables *name*, *addr*, *suburb* and *phone*.

1b – creates an empty list called `entries`.

1c – opens a file called `phonebook.csv` in the current directory, and reads its contents into a list variable called *lines*. When it unindents, it closes the file.

We can assume the file is formatted "Jones Fred, 123 Smith Street, Perth, 9234 5678" on each line.

1d – "lines" is a list of lines, so "line" is one line of the original csv file. This code strips newline and whitespace characters from it, and splits the line by comma into variables a, b, c and d. Then it creates an instance of the `PhoneBook` class out of the parts of the line, and adds it to the list `entries`.

Some questions

1. What does `for i in range(5):` mean?
   It runs the following block of code using i = 0 at first, then i = 1, 2, 3 and 4. The last number, 5, is never included.
2. If we changed `i` above to `woohoo`, would it work? Why/why not?
   Yes, it would be fine. Any variable name can be used.
3. Which of the following variable names are allowed in Python?
   (a) `av4tar` – Yes. Breaks no rules.
   (b) `avatar1` – Yes
   (c) `4vatar` – No, variable names can't start with a number.
   (d) `ava_tar` – Yes
   (e) `ava.tar` – No, variable names can't contain a dot.
   (f) `list1` – Yes
   (g) `list` – No, variables can't be the same as reserved words.
4. What is the difference between `for k in range(len(knightList)):` and `for k in knightList)):` ? Focus in particular on what k ends up meaning, and how you would refer to elements of knightList in the loop.
   The first k is an index, or number, indicating the position within knightList – we have to use `knightList[k]` to refer to elements. The second k is the element itself.
5. You have your functions in one .py file, and your calling code in another (For example, every prac test since PT2). In the calling code, should you have `import functions` or `from functions import *` ?
   Both have their advantages and disadvantages. When you write `import numpy as np`, you have to use e.g. `np.zeros` to access its functions. If you know there isn't going to be double-ups and you want to just use the function name on its own, use the "from" version. This will more often than not be the case when you've written the functions yourself.

Spot the error

1 – Possibly the most common error of all - `=` means "assign the right value to the left variable", `==` means "does left equal right?"

2 – This could be most easily fixed by changing the "else" to "elif". The `else:` never has an expression after it, it means "or none of the above".

3 – When printing, you can use different data types together. But when making a string, you MUST convert, e.g. numbers to strings. Change `number` to `str(number)` to fix this one.

4 – Another amazingly common error - not closing all your brackets. Add another) to fix this one. This happens especially when your expression has a lot of brackets, it is easy to lose track. Just count from left to right and make sure they match.

5 – Forgot the colon at the end of the if statement. If you type a line like this in vim, hit enter and it doesn't indent, this is the most likely cause.

6 – Two errors: 1. Forgot to put a "return" statement in the function. You don't need one for a purely "doing" function, but if the calling code wants to print the result or set a variable, you must have one. 2. Forgot the `self.` in the print statement - myFunction is a class method so should have been `print(self.myFunction(whatever))`.

Error messages

- `IndentationError: expected an indented block` – Comes up when the block after an if:, for:, while:, def: statement has not been indented.
- `IndentationError: unindent does not match any outer indentation level` – Means some of your lines might be one or two spaces out of alignment. Check that they are all multiples of four spaces.
- `NameError: name 'x1' is not defined` – Comes up when your code uses a variable name that doesn't exist anywhere in your code - check spelling, and check scope (i.e. is the variable defined in a function or method but you're trying to use it in main code?) If the name is "np" or "plt", check that you have, e.g. `import numpy as np` at the top.
- `IndexError: list index out of range` – The code went past the end of a list. This can happen if the maximum value of a loop is longer than the list, or if you have deleted list items while still iterating over it.
- `UnboundLocalError: local variable 'x' referenced before assignment` – The code used the variable up here but defined it down there. For example, appending to a list that hasn't been created yet.
- `AttributeError: 'module' object has no attribute 'x'` – Your class doesn't have a class variable with that name.
- `SyntaxError: invalid syntax` (when the line it shows looks OK) – Check the line before it for incomplete brackets.
- `SyntaxError: unexpected EOF while parsing` – Check the line before it for incomplete brackets, or any if / for / etc blocks ending with a colon that have nothing after them (if you just haven't written the code for it yet, put the single command `pass` inside it).

– `ImportError: No module named matplotlib.pyplot` – This sometimes happens on the lab machines. Type `pip3 install matplotlib` (or whichever library is missing) at the Linux command prompt to fix it.

List comprehensions

– `[i*5 for i in range(1, 11)]` or `[i for i in range(5, 55, 5)]` (start, stop, step)
– `[i*2 for i in myList]`
– `[word for word in wordList if word[1].lower() == "e"]`
– `len([a for a in list if a in [2016, 2019, 2022])`