Pseudocode, Conditionals and Operator Precedence

February 5, 2020

Administrative notes

- First, I owe every student in this class an apology for the fact that the discussion rooms didn't have computers
  - "Trust, but verify" - I trusted; I didn't verify; that's on me
  - We're working on getting this solved before next week. We'll keep you posted
- Homework 1 is still due next Monday
  - We'll talk about it at the end of lecture
  - If you don't have a computer, UMBC has some at the library you can use
-

Pseudocode

A natural language description of what a section of a program is supposed to be doing

Used during program design

- Write the pseudocode first
- Then translate it into the code

An Example

Pseudocode

To calculate a GPA: Assign values to the number of hours earned and total quality points. Calculate the GPA as number of quality points divided by number of hours earned. Print the GPA.

Code:

```
hours = int(input(“Enter the number of credit hours you have”))

q_p = float(input(“Enter the number of quality points you have earned”))

gpa = q_p/hours
```

So, what do you need to know how to do?

1. Write pseudocode. It will help you explain what you are trying to accomplish with your program.
2. Take someone's pseudocode, and produce Python code that accurately implements the pseudocode

Conditionals

Sometimes you want to write code that only gets executed sometimes

- Only print an error message if the user has entered a negative number for “age”
- Only print a congratulatory message if the student has made the Dean’s List
- …

So you use conditionals to control whether or not a section of code (one or more statements) gets executed

Booleans

Remember that Booleans are expressions that can only have two values: True or False

on_Deans_list = True

new_student = False

6 > 5 is a Boolean expression with value True

“Baltimore” == “baltimore” is a Boolean expression with value False

Boolean Operators: And, Or, Not

x	y	x and y
True	True	True
True	False	False
False	True	False
False	False	False

x	y	x or y
True	True	True
True	False	True
False	True	True
False	False	False

X	not X
True	False
False	True

The simplest conditional: the if statement

```
if [boolean]:
   # do something
```

Example:

```
if student == "awesome":
   print("You must be here for CMSC 201")
```

Now you know why we spent so much time talking about True, False, comparison and logical operators! They go after ‘if'!

Two cases: If and Else

Ask a student for her major. If she’s a Physics major, print out “smart choice.” Otherwise print out “there is still time”

```
major = input(“please enter your current major”)

if major == “Physics”:

    print(“smart choice”)

else:

    print(“there is still time”)
```

Remember that input returns a string, so this comparison is valid

Colon after else, as well!

Indent the code that’s part of the “else” block

Python and indentation

The boolean condition is everything between “if” and the colon : A colon terminates the condition. It can be as simple or as complex as you want

Indentation matters!! To python, white space - either tabs or spaces - indicates what’s in the code to be executed. You only have to indent one space, but I’m a believer that you should indent with tabs.

```
If x == 6:
   print (“X is 6”)

   print (“and we’re done”)
```

If x is really equal to 5, this doesn’t print out anything.
This still prints “and we’re done” because that’s not part of the conditional

```
if x == 6:
  print (“X is 6”)
print (“and we’re done”)
```

“if...else…”

- There must always be at least one line of code under the “if” statement
- You don’t have to have an “else” part, but if you do have an “else” there must be at least one line of code under it.

Three or more cases: elif

Input returns a string. This turns it into an integer

Ask a student her age. If it’s less than 18, tell her she’s a minor and faces some restrictions. If it’s between 18 and 21, tell her she’s an adult but still has some limitations. If she’s over 21, tell her she’s legally an adult.

This is ‘pseudocode’

```
age = int(input(“Please enter your current age in years.”))
if age < 18:
        print(“Sorry but you are a minor”)
        print(“there are a lot of things you cannot do on your own”)
elif age < 21:
        print(“you are an adult but there are still some things you can’t do”)
else
        print(“congratulations you are legally an adult”)
```

Three forms of conditionals:

1. One option: If a condition is true, do something. Otherwise, do nothing. “If” statement
2. Two options: If a condition is true, do something, Otherwise, do something else. “If….else…” statement
3. More than two options: If a condition is true, do something. Otherwise, check to see if another condition is true; do something else. Otherwise, keep checking conditions until we find one that’s true or we just give up. “If...elif...elif...else…” statement

Thanks, Dr. Johnson

Order of operations

`**`	Highest
`* / % //`	
`- +`	
`<= < > >= != ==`	
`not`	
`and`	
`or`	Lowest

Remember, like the noble and majestic honey badger, parentheses don't care about order of operations and will take precedence over everything.

Why operator precedence is important

What’s the value of each:

6 * 5.0 + 2 - 5 / 5

(6 * (5.0 + 2) - 5) / 5

6 < 2 or 5 > 1

6 < (2 or 5) > 1

Homework 1

Now let’s spend some time talking about homework #1