COMP 204

Control flow - Conditionals

Yue Li,

based on material from Mathieu Blanchette, Carlos Oliver and Christopher Cameron
Quiz 5 password
Recap from last lecture

Variables in logical comparison

```python
Weight = float(input("Enter weight (in kg): "))
Height = float(input("Enter height (in m): "))
userBMI = Weight/(Height**2)
LowBMI = 18.5
HighBMI = 25
userBMI < LowBMI  # under weight
userBMI >= LowBMI and userBMI <= HighBMI  # normal
userBMI > HighBMI  # overweight
```
Control flow

Until now, every line of our programs was executed exactly once, from top to bottom. This is very limiting!

- **Conditionals**: we may want to only execute a piece of code if a particular condition holds (e.g. if BMI is low, do something)
- **While Loops**: We may want to re-use certain pieces of code multiple times (e.g. keep asking someone the same questions until we get the correct answer)
- **For Loops**: We may want to perform the same operation on a large number of objects (e.g. change every 'T' to an 'A' and every 'G' to a 'C' in a complementary DNA sequence)

This is achieved using control flow instructions. The control flow of a program determines:

- Which part of the code should be executed regardlessly
- Which blocks of code should be executed only under certain circumstances (conditional execution, *today lecture*)
- Which blocks of code should be executed repeatedly, and for how many times
Control flow

Until now, every line of our programs was executed exactly once, from top to bottom. This is very limiting!

- **Conditionals**: we may want to only execute a piece of code if a particular condition holds (e.g. if BMI is low, do something)

- **While Loops**: We may want to re-use certain pieces of code multiple times (e.g. keep asking someone the same questions until we get the correct answer)

- **For Loops**: We may want to perform the same operation on a large number of objects (e.g. change every 'T' to an 'A' and every 'G' to a 'C' in a complementary DNA sequence)

This is achieved using control flow instructions. The control flow of a program determines:

- Which part of the code should be executed regardlessly
- Which blocks of code should be executed *only under certain circumstances* (conditional execution, **today lecture**)
- Which blocks of code should be executed repeatedly, and for how many times
We use conditional execution to only execute a block of code if a certain boolean expression is true.

```python
if booleanCondition:
    # this block of code is only executed
    # if booleanCondition is true
else:
    # this block of code is only executed
    # if booleanCondition is false

# this is outside the conditional
# this gets executed no matter what
```

IMPORTANT: In Python, we use indentation (tab character) to indicate what block a line belongs to.
Example 1: BMI revisited (demo in class)

```python
weight = float(input('Please enter your weight: '))
height = float(input('Please enter your height: '))
BMI = weight / (height ** 2)
print('Your BMI is ', BMI)

if BMI < 18.5:
    print("You are underweight")  # Lines 7 and 8 are only executed if BMI < 18.5
    print("Try to gain weight")
else:
    print("You are not underweight")

print("Thank you for using the BMI calculator")
```

Notes:

- Lines 7 and 8 form a block of code. They are indented together.
- The block 7-8 only gets executed if BMI < 18.5
- The block 10 only gets executed is BMI is not < 18.5
- Line 12 is outside the conditional; it gets executed after the conditional.
Example 2: BMI re-revisited

```python
weight = float(input('Please enter your weight: '))
height = float(input('Please enter your height: '))
BMI = weight / (height ** 2)
print('Your BMI is', BMI)

if BMI < 18.5:
    print("You are underweight")
    print("Try to gain weight")

if BMI >= 18.5 and BMI < 24.9:
    print("Your weight is normal")

if BMI > 24.9:
    print("You are overweight")
print("Thank you for using the BMI calculator")
```

In line 10, we use logical key word “and” to combine two statements “BMI >= 18.5” and “BMI < 24.9”
Example 2: BMI re-revisited (a logical mistake)

This is almost the same code, but it won’t work properly: why?

```python
weight = float(input('Please enter your weight: '))
height = float(input('Please enter your height: '))
BMI = weight / (height ** 2)
print('Your BMI is ',BMI)

if BMI < 18.5:
    print("You are underweight")
    print("Try to gain weight")

if BMI >= 18.5 and BMI < 24.9:
    print("Your weight is normal")
else:
    print("You are overweight")

print("Thank you for using the BMI calculator")
```
Chained conditional

To execute exactly one of several blocks, we can use the if-elif-else structure.

```python
if condition1:
    # this is executed only if condition1 is true
elif condition2:
    # this is executed only if condition1 is false and condition2 is true
elif condition3:
    # this is executed only if condition1 is false and condition2 is false and condition3 is true
else:
    # this is executed only if all three conditions are false
```
Example 2: BMI re-re-revisited

This version works correctly.

```python
weight = float(input('Please enter your weight: '))
height = float(input('Please enter your height: '))
BMI = weight / (height ** 2)
print('Your BMI is ', BMI)

if BMI < 18.5:
    print("You are underweight")
    print("Try to gain weight")
elif BMI >= 18.5 and BMI < 24.9:
    print("Your weight is normal")
else:
    print("You are overweight")
    print("Try to loose weight")
print("Thank you for using the BMI calculator")
```
Nested conditionals

We can have conditionals inside conditionals:

```python
if condition1:
    # this is executed only if condition 1 is true
    if condition2:
        # this gets executed only if
        # both conditions 1 and 2 are true
    else:
        # this gets executed only if
        # condition 1 is true but condition 2 is false
else:
    # gets executed only if condition1 is false
    # we could have more if/else here

# this is outside the conditional
# this gets executed no matter what
```

▶ Note double indentation
Example 3: Nuclear accident evacuation

Task: Write a program to provide the correct evacuation message following a nuclear accident.

- Location of nuclear accident
- 20 km radius: Mandatory evacuation for all
- 40 km radius:
  - Pregnant?
    - If yes, Mandatory evac
    - If no, Recommended evac

Your home
Example 3: Nuclear accident evacuation

Task: Write a program to provide the correct evacuation message following a nuclear accident.

Euclidean distance:

$$\sqrt{(x_{\text{acc}} - x_{\text{home}})^2 + (y_{\text{home}} - y_{\text{acc}})^2}$$
Example 3: Nuclear accident evacuation

```python
import math  # this imports the math module

xAcc = float(input("Enter x coord. of nuclear accident: "))
yAcc = float(input("Enter y coord. of nuclear accident: "))
xHome = float(input("Enter x coordinate of home: "))
yHome = float(input("Enter y coordinate of home: "))

distance = math.sqrt((xHome - xAcc)**2 + (yHome - yAcc)**2)

if distance <= 20:
    print("You must evacuate")
elif distance <= 40:
    pregnant = input("Are you pregnant? (yes/no) ")
    if (pregnant == "yes" or pregnant == "Yes" or pregnant == "Y" or pregnant == "y"):
        print("You must evacuate")
    else:
        print("Evacuation is recommended")
else:
    print("No need to evacuate")
```
import math  # this imports the math module

xAcc = float(input("Enter x coord. of nuclear accident: "))
yAcc = float(input("Enter y coord. of nuclear accident: "))
xHome = float(input("Enter x coordinate of home: "))
yHome = float(input("Enter y coordinate of home: "))

distance = math.sqrt((xHome - xAcc)**2 + (yHome - yAcc)**2)

if distance <= 20:
    print("You must evacuate")
elif distance <= 40:
    pregnant = input("Are you pregnant? (yes/no) ")
    if (pregnant == "yes" or pregnant == "Yes" or pregnant == "Y" or pregnant == "y"):
        print("You must evacuate")
    else:
        print("Evacuation is recommended")
else:
    print("No need to evacuate")
Example 4: Tumor classification by decision tree

Task: Write a program to guide doctors in their assessment of tumors.
# the content of this variable
# will be changed by the code below

```python
tumorType=""

adhesion = int(input("Enter marginal adhesion level"))
if adhesion <= 3:
    clump = int(input("Enter clump thickness"))
    if clump <= 3:
        # tumorType="Benign"
        a=""
    else:
        uniformity = int(input("Enter uniformity of cell shape"))
        if uniformity <= 2:
            tumorType="Benign"
        else:
            tumorType="Cancer"
else:
    bare = int(input("Enter level of bare nuclei"))
    if bare <= 4:
        tumorType="Benign"
    else:
        tumorType="Cancer"
```

print("The tumor type is: ", tumorType)
Assignment 1 will be released tonight after midnight