

Name: _____

Partner: _____

Python Activity 9: Nested IF-ELIF-ELSE Statements

How do we write code that branches within branches?

Learning Objectives
Students will be able to:

Content:

- Explain the purpose of a nested if-(elif-else) statement
- Compare the use of nested if-statements to using only logical operators

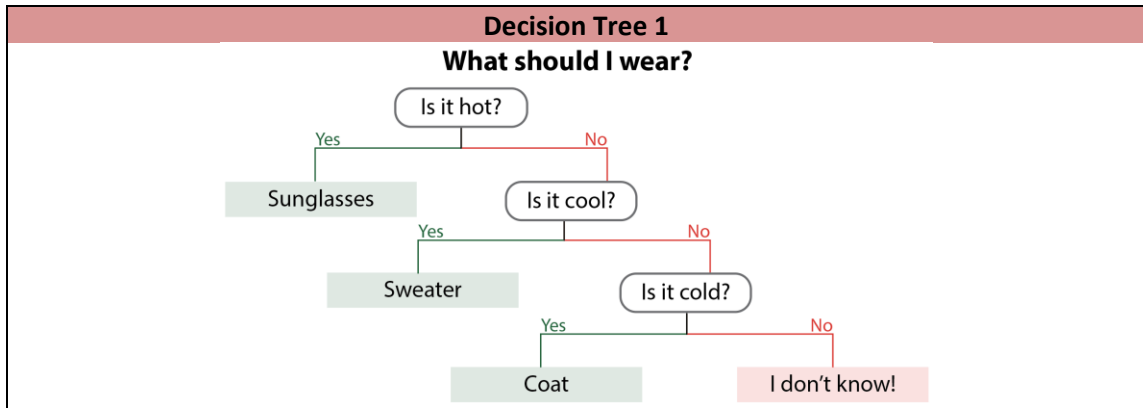
Process:

- Write code that uses nested if-statements

Prior Knowledge

- If..elif..else, bools, variables, types, expressions, assignment, functions

Concept Model:

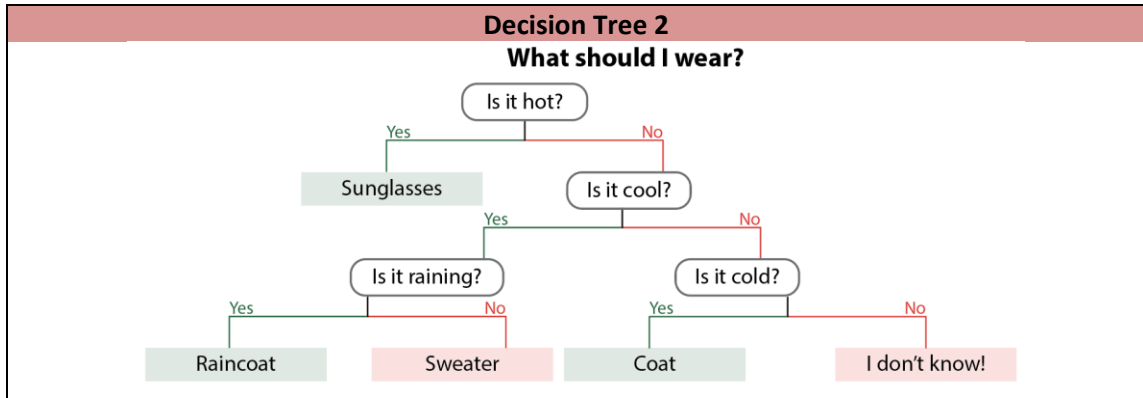


FYI: Pseudocode is a high-level, syntax-free way of communicating about code commands in a [mostly] layperson-readable form that describes the logic of a program without being bogged down by syntax.

CM1. Currently, we can represent the decision tree in the above *Concept Model* with the if..elif...else statements, and conditional & logical operators we learned previously. Fill-in the following code structure below with **pseudocode** of boolean expressions and other pseudocode to represent the decision tree in the Concept Model:

```
if _____ (is hot) _____ :  
    _____ print sunglasses _____  
  
elif _____ :  
    _____  
  
elif _____ :  
    _____  
  
else:  
    _____
```

The approach in CM1 works with the simple structure of Decision Tree 1. Examine the similar, but slightly more complex decision tree below:



CM2. Circle the part of the decision tree that was added in Decision Tree 2.

CM3. How would we have to modify the pseudocode in CM1 to accommodate this additional question using what we've learned so far in class?

if _____ (is hot) _____ :

_____ print sunglasses _____

elif _____ and _____ :

elif _____ and _____ :

elif _____ :

else:

CM4. If we also wanted to add an option for an All-weather coat (versus a Wool coat) when it's cold and raining (or not), how would we have to modify the above pseudocode?

CM5. Will using logical operators *scale* well for much more complex decision trees? Why/not?

Critical Thinking Questions:

1. Closely examine the Python program below, it represents Decision Tree 2.


```



Python Program


def weatherApparel(weather, raining):
    if weather == 'hot':
        print('Sunglasses')
    elif weather == 'cool':
        if raining == 'y':
            print('Raincoat')
        else:
            print('Sweater')
    elif weather == 'cold':
        print('Coat')
    else:
        print("I don't know")


def main():
    wthr = input("What is the weather? (hot, cool, or cold): ")
    rain = input("Is it raining? (y or n): ")
    weatherApparel(wthr, rain)

main()
```

- a. In the Python code, circle the if-block that is **nested** within another if-block.
-  b. How does this nested if-block differ from our approach in CM3 using logical operators?

-  c. List 6 combinations of values for `weather` and `raining` to test different all parts of this program. Indicate what part of the program the input is testing. (Enter and test the code as a class / at home).

weather	raining	Code/Part Tested

- d. Modify the above Python code so it has the option for an All-weather coat when it's cold and raining, and a Wool coat when it's cold and not raining, using **nested-ifs**.
-  e. How might using nested-ifs scale differently than logical operators, for complex decision trees?
