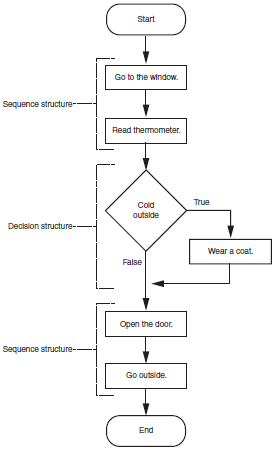
**COSC 1336, Lab 3 Instructions, Decision Structures**

Decisions on digital computers are decided on yes or no (true or false) boolean values. “Maybe” does not apply. Whatever the inputs, booleans are needed to make decisions.  
There are two main decision-making tools in programming languages:

1. control flow, using if then else (if elif else in Python)
2. boolean logic, using boolean (true/false) variables and logical operators: and, or, not.

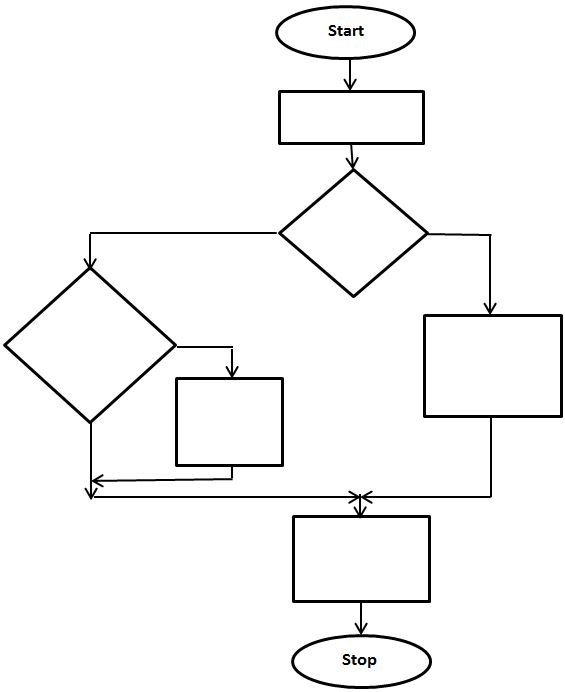
**Part 1:** In the textbook, see: “Nested Decision Structures and the if-elif-else Statement. See textbook: figure 4-12, page 134 (2e); or figure 3-10, page 97 (3e); or figure 3-10, page 125 (4e). This is a program that gives advice on dressing properly for the weather. This is the flowchart:



**First:** implement the flowchart in Python. Where you see commands, such as: go to the window, or wear a coat, use print() statements. Where a decision must be made, such as, cold outside, determine a valid condition statement. You will need input to determine the condition. For “Read thermometer”, use an input statement. Get an integer Fahrenheit value from the user. Based on the input, make the decision – is it cold outside? Take the appropriate action as illustrated in the flowchart.

**Second**: Add a new decision, not depicted in the flowchart above. After the “Open the door.” action is taken (use a print statement for this), also decide whether you need an umbrella. To decide this, ask: “Is it raining (y/n)?” ***Ignore case, so: Y, y, N, n all work***. If so, print “Bring an umbrella”.

Save the code in a file: **DDHH\_L3\_Lastname\_part1.py.** Do not submit this file yet – there is more to do. The code from this part will be combined with other parts into one, large python program which you will submit for lab 3.

**Part 2:** Below is an empty flowchart. It only labels ‘Start’ and ’End’. It uses some nested decision structures. Write a Python program that uses this structure. You decide what it does! Notice that there are only two diamond decision structures, so your code should only make two decisions as shown. Hint: You will have to add some extra statements to prompt for and get information so you can make the decisions. For now, save this in the file: **DDHH\_L3\_Lastname\_part2.py.** Do not submit until the other parts are completed and combined into one file.  


**Part 3:** One role of decision making is deciding who won a game. Consider the game: rock, paper, scissors. Two players simultaneously choose one item: rock or paper or scissors. Rock beats (smashes) scissors; scissors beats (cuts) paper, paper beats (wraps) rock. If both players choose the same item, it is a tie. For example, if PlayerA==rock and PlayerB==scissors, PlayerA wins, because rock beats scissors. Notice the == instead of =. Use == for testing equivalence and = for assigment.

Create a short program that plays the game. For example:  
**playerA choice: r)ock, p)aper, s)cissors? r  
playerB choice: r)ock, p)aper, s)cissors? s  
playerA beats playerB because rock beats scissors**

Verify the program works with all possible inputs for playerA and playerB. To make it easier to play, only require a single letter: **r**, **p**, or **s** from the user. Output a clear statement as to who won or lost, why. If it is a tie, output a message like:  
**playerA ties playerB because paper equals paper**

Details: **if the user choice is invalid, assume rock. *In your test output, do not loop; play the game just once.*** The lab objective is to practice decision making, not looping. After this is done, save the code in a file: **DDHH\_L3\_Lastname\_part3.py**. Do not submit this file. There is more to do. This code will be combined with the other parts and submitted as one file.

End

**Part 4:** Write a Python program that inputs a number for a season, from 1 to 4. If the user enters an invalid number (not 1 to 4) print an error message and move on.

For valid season numbers (1 to 4), output the name of that season and something about that season (a string). For example: input: 1, output: “winter is cold”; input: 2, “spring is green”; input: 3, “summer is hot”; input: 4, “fall is cool”. Hint: You could use nested if/else structures, but that will cause “runaway indenting”. A better choice is nested if/elif structures. You can save this in a file: **DDHH\_L3\_Lastname\_part4.py.** There is a bit more to do. All parts will be combined in one file.

**Part 5:** Write a (short) Python program that decides whether a person can (and should) drive home. Use the following variables *with these settings*: **battery\_charged=True; got\_car=True; drunk=False; gas=2** (gallons); **distance=100** (miles); **mpg=35; nighttime=False; headlights\_out=True**. **Pre-set these variables, do not ask the user for each value.**

You can use this code near the beginning of your program:

battery\_charged=True

got\_car=True

drunk=False

gas=2 # (gallons) # gas currently in the tank of the car

distance=100 # miles from home

mpg=35 # miles per gallon expected to be used driving home

nighttime=False

headlights\_out=True

Create one, large boolean expression that computes the correct value for **can\_drive** (True or False). Based on the value of **can\_drive**, output either: **“Drive home”** or **“Do not drive home”**.

if can\_drive:

print('Drive home')

else:

print('Do not drive home.')

The intention is for you to practice setting and using boolean variables to make decisions. Use all the provided variables in your decision. The boolean literals: **True**, **False** are case-sensitive! Save this in the file: **DDHH\_L3\_Lastname\_part5.py**

**Part 6:** After you are happy with each part above, combine all parts into one, large python file. Make sure all parts work OK before combining them into one file. Name the one, large combined file: **DDHH\_L3\_Lastname\_parts1-6.py.** If you did not do all parts, use the file name to describe which parts you did. Test all parts and paste the test output at the bottom. Submit this one, large python file on the Blackboard assignment link for Lab 3.

**Part 7:** **EXTRA CREDIT.** This is optional. It is for extra credit, up to 5 points. At the top of the large program, add a prompt. Ask the user which part they want to jump to:  
**Choose which part to perform:  
 1) Dress for weather  
 2) Flowchart  
 3) Rock, paper, scissors  
 4) Seasons**

**5) Drive car  
 6) Quit now  
Your choice?**

Based on the user’s choice, perform the one, chosen operation. This makes the code concise and easy to use; any desired option can be run independently. Call the file: **DDHH\_L3\_Lastname\_parts1-7.py**

**Summary: One file should be submitted: DDHH\_L3\_Lastname\_parts1-6.py or DDHH\_L3\_Lastname\_parts1-7.py if you did the extra credit.**