POGIL Activity 1 (MA8): Boolean Expressions and Selection Statements

"True or False and making choices"

Original activity author: Lisa Olivieri

Manager name: ________________________________

Technician name: ________________________________

Recorder name:__________________________________

Learning Objectives

Students will be able to:

Content:

- Explain the three types of programming structures
- Explain how conditional operators and logical operators are used in programming
- Use conditional operators with strings and numeric values
- Implement the Python syntax of an if/else statement
- Determine good test data for programs that include if/else statements

Process:

- Write code that includes if statements and if/else statements
- Write correct Boolean expressions and compound expressions

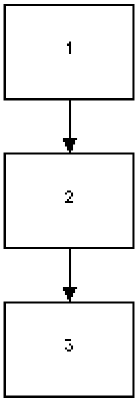
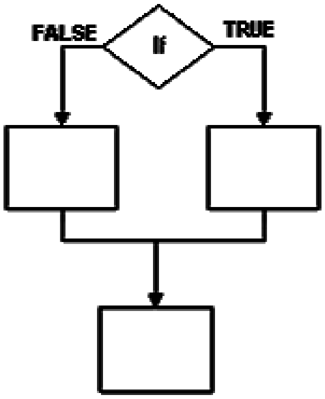
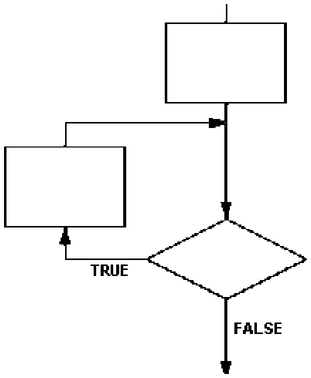
Prior Knowledge

- Python concepts from L1-1 through L5-1
- Understanding of flowchart input symbols (diamond = decision, rectangle = process, arrow = flow direction)

Further Reading

- L5-2: http://nbviewer.jupyter.org/github/gsprint23/cpts111/blob/master/lessons/L5-2.ipynb

Model 1: Programming Structures

Sequence Structure	Decision or Branching Structure	Looping Structure
		

Critical Thinking Questions:

1. Each of the flowchart symbols in the chart above represents lines or segments of code. After examining the chart, write a description of:

   a. sequence structure

   b. decision or branching structure

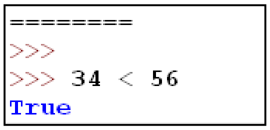
   c. looping structure

2. Which structure best describes the types of Python programs you have written so far? Do you think this structure is the best structure to use for all programs? Why or why not?

3. Which structure allows the programmer to create code that decides what code is executed?

Model 2: Conditional Operators
Conditional operators, also known as relational operators, are used to compare the relationship between two operands. Expressions that can only result in one of two answers are known as **Boolean expression.**

4. Determine the meaning of each of the following **conditional operators.** If you are not sure of the meaning of any symbol, create some example expressions, type them into the IPython console (i.e. a Python *interpreter)* in Spyder (see figure to the right) and examine the results.



```
========
>>>
>>> 34 < 56
True
```

a. <　　　b. >

c. <=　　　d. >=

e. !=　　　f. ==

5. What is the result of each of the following expressions?
Assume x = 4, y = 5, z = 4

a. x > y

b. x < y

c. x == y

d. x != y

e. x >= z

f. x <= z

g. x + y > 2 * x

h. y * x – z != 4 % 4 + 16

i. math.pow(x,2) == math.fabs(-16)

6. What is the result of the following expressions?
Assume word1 = “hello”, word2 = “good-bye”

a. word1 == word2

b. word1 != word2

c. word1 < word2

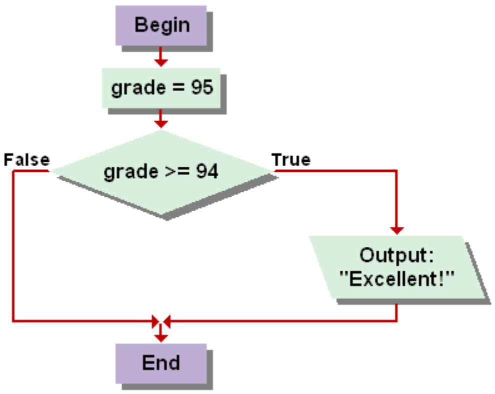
d. word1 >= word2

7. Explain how the conditional operators are used when the operands are strings.

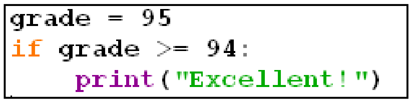
8. What are the only two possible answers to all of the expressions in questions 5 and 6?

Model 3: IF/ELSE Statement

Flow chart



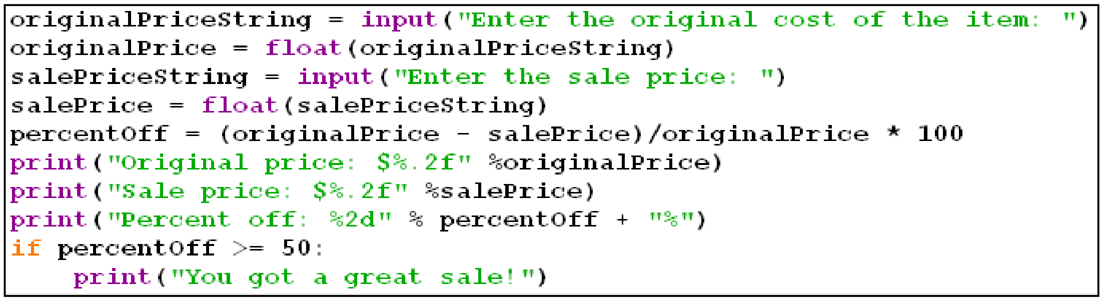
Python Program



```
grade = 95
if grade >= 94:
    print("Excellent!")
```

9. What is the Python code equivalent of the diamond flowchart symbol in the program above?

10. Enter and execute the following code. Use various values for the original cost and the sale price.



```
originalPriceString = input("Enter the original cost of the item: ")
originalPrice = float(originalPriceString)
salePriceString = input("Enter the sale price: ")
salePrice = float(salePriceString)
percentOff = (originalPrice - salePrice)/originalPrice * 100
print("Original price: $%.2f" %originalPrice)
print("Sale price: $%.2f" %salePrice)
print("Percent off: %2d" % percentOff + "%")
if percentOff >= 50:
    print("You got a great sale!")
```

11. Explain what the following lines of code do. Each line appears in the program above.

a. `originalPrice = float(originalPriceString)`

b. `percentOff =(originalPrice - salePrice)/originalPrice * 100`

c. `print("Sale price: $%.2f" %(salePrice))`

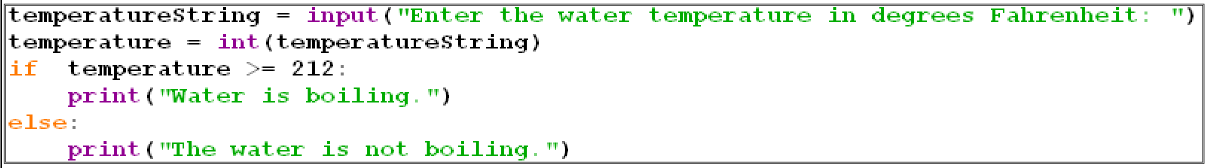
d. `print("Percent off: %2d" %(percentOff) + "%")`

e.
```
if percentOff >= 50:
    print("You got a great sale!")
```

12. Revise the program in #10. If the percent off is 50% or more print “Congratulations!” in addition to what is already printed. Use a second print statement to do this. Write the code for this part of the program.

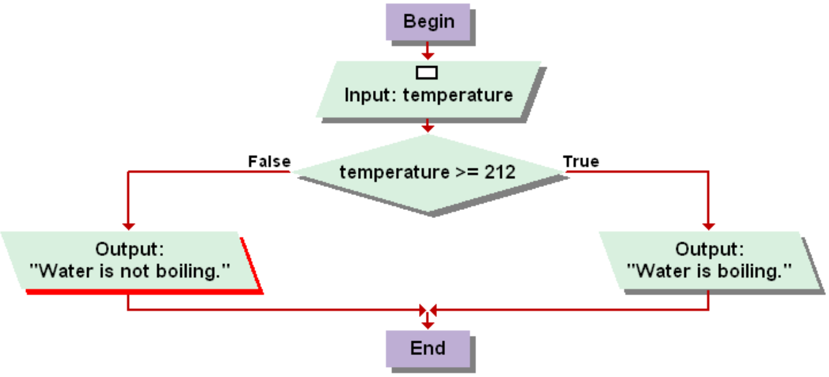
13. Revise the program in #12 so that it prints “Done!” when the program is complete – no matter what the percent off is. How does the placement of this line of code differ from the placement of the code created for #12?

Python Program



```
temperatureString = input("Enter the water temperature in degrees Fahrenheit: ")
temperature = int(temperatureString)
if  temperature >= 212:
    print("Water is boiling.")
else:
    print("The water is not boiling.")
```

Flowchart



14. Compare the flowchart above with the corresponding Python Program. Carefully compare if/else statement in the two programs.

a. Explain the correct syntax of a Python if/else statement.

b. Enter and execute the Python program above. Test it at least three times. List below the three test data you used and the corresponding output. Explain why they were the best data to use as a test for the program.

c. Suppose you wanted to add another print statement to the Python program above so that it printed “That’s really hot!” when the water is 212 degrees or hotter. Rewrite the code below with this statement included.

FYI: We can use **logical operators** to determine logic between conditions (relational expressions).

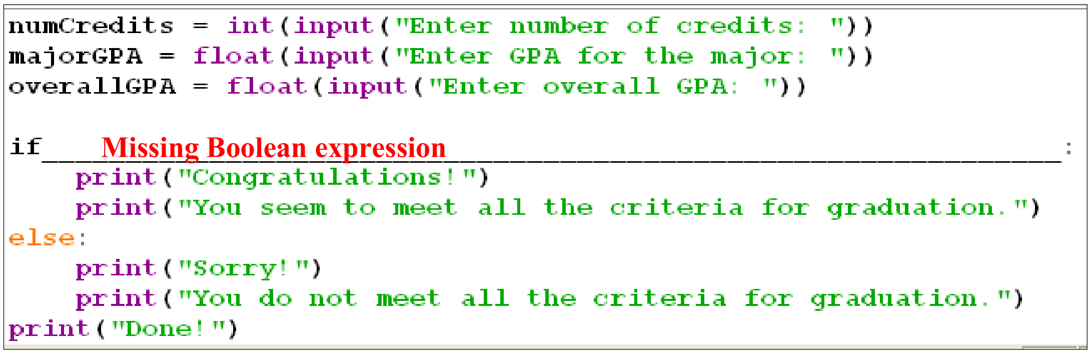
15. Sometimes you want to include more than one condition to determine which code segment is executed. You can use the following **logical operators** to create **compound conditions**. Examine each operator and a sample of its use and provide an explanation of how the operator works.

Operator	Example	Explanation
and	(age >= 17) **and** (hasLicense = = true)	
or	(cost < 20.00) **or** (shipping = = 0.00)	
not	**not** (credits> 120)	

16. Assume the value of the variable **numBooks** is 40. State the values of each of the Boolean expressions that include a compound condition.

Expression	Value
(numBooks > 5) and (numBooks < 100)	
(numBooks < 5) or (numBooks > 100)	
not(numBooks * 10 == 100)	

17. Suppose you want to determine if a student is ready to graduate. The 3 criteria for graduation are that the student has earned at least 120 credits, their major GPA is at least 2.0 and their general GPA is also at least 2.0. Which Boolean expression would be the correct test for Python code?



```
numCredits = int(input("Enter number of credits: "))
majorGPA = float(input("Enter GPA for the major: "))
overallGPA = float(input("Enter overall GPA: "))

if____Missing Boolean expression____________________________________:
    print("Congratulations!")
    print("You seem to meet all the criteria for graduation.")
else:
    print("Sorry!")
    print("You do not meet all the criteria for graduation.")
print("Done!")
```

a. numCredits >= 120 or majorGPA >= 2.0 or overallGPA >= 2.0

b. numCredits > 120 and majorGPA > 2.0 or overallGPA > 2.0

c. numCredits > 119 and majorGPA >= 2.0 and overallGPA >= 2.0

d. numCredits >= 120 and majorGPA >= 2.0 and overallGPA >= 2.0

18. Enter and execute the program in #17. Include your choice for the correct Boolean expression and create several sets of data to test all possibilities for the Boolean expression. List the data you used to test all possibilities for the expression and explain your choices.

Data Set	numCredits	majorGPA	overallGPA	Expression Result (True or False)
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

19. To summarize logical operators, finish filling in the table below (also called a "truth table"). For example, when either operand (i.e. X, Y) of logical **and** is False, X and Y evaluates to False: "False **and** True is False", "True **and** False is False."

X	Y	not X	X and Y	X or Y
False	False			
False	True		False	
True	False		False	
True	True			

Application Questions: Use the Python Interpreter to check your work

1. Write a Boolean expression that tests if the value stored in the variable **num1** is equal to the value stored in the variable **num2**.

2. Write a Boolean expression that tests if the value stored in the variable **time** is less than the value stored in the variable **maxTime** or if the value stored in the variable **cost** is less than the value stored in the variable **maxCost**

3. Write the code for an **if** statement that adds 5 to the variable **num1** if the value stored in the variable **testA** equals 25. Otherwise subtract 5 from **num1.**

4. Write a Python program that prompts the user for a word. If the word comes between the words **apple** and **pear** alphabetically, print a message that tells the user that the word is valid, otherwise, tell the user the word is out or range.

5. Write a Python program that prompts the user for a multiple of 5 between 1 and 100. Print a message telling the user whether the number they entered is valid.