SeaGlide_TEAM	Conditional Statements	Lesson # 4 (60 minutes)

Description:

Your role as an assistant computer programmer means that you must learn to be proficient in many types of programming and coding. For this assignment, you will have to continue learning Python by applying and extending your prior knowledge. Because you are in training, you will be working on several modules and continue to build upon your skills. You will work on various skills in Python and submit them to the head computer programmer to make sure your codes are both sufficient and efficient. For this task, you will learn and utilize conditional statements in Python by designing your own program.

Students will be able to:

- Understand boolean expressions and the bool variable type.
- Differentiate between mathematical symbols and Python relational operators.
- Utilize logical operators.
- Debug given code that has only logic errors.
- Understand and utilize chained conditional statements.
- Apply logic and problem solving skills to design a program.

Students will understand:

Python is a widely known, high-level coding language. Prior to this lesson, students mastered declaring variables, utilizing built-in functions and implementing user defined functions. In this lesson, students will understand the bool variable type and how to breakdown a given code with conditional statements. Students will continue learning how to design their own program that utilizes conditional statements and functions.

Key Definitions & Concepts: [1]

- **Boolean expression:** an expression whose value is either True or False
- **Relational Operator:** one of the operators that compares its operands: ==, !=, <,>, <=, >=
- **Logical Operator:** one of the operators that combines boolean expressions: and, or, not
- **Conditional Statement:** a statement that controls the flow of execution depending on some condition
- **Condition:** the boolean expression in a conditional statement that determines which branch is executed
- **Chained Conditional:** a conditional statement with a series of alternative sequences of conditional statements
- **Nested Conditional:** a conditional statement that appears in one of the alternative sequence of conditional statements

Standards: [Copied from: 2]

1B.AP.10: Create programs that include sequences, events, loops and conditionals.

2.AP.13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

Background Information

Prior Knowledge:

- Variable assignment
- Expression evaluation
- Defining functions
- Utilizing math module

Science Practices: [Copied from: 3]	**Core Ideas: [Copied from: 4]**	**Cross Cutting Concepts: [Copied from: 5]**
• Constructing explanations and designing solutions • Obtaining, evaluating and communicating information	• Developing possible solutions • Optimizing the design solution	• Patterns • Structure and function

Possible Preconceptions/Misconceptions:

Students may struggle through the *Investigation and Programming* exploration worksheet. The prompt given to the students asks them to create a Python code that includes chained conditionals, the datetime module, and string formatting. Since this lesson does not explicitly discuss the utilization of the datetime module and about string formatting, students are not expected to understand how to use them. Hence, a skeleton of the lesson is provided for the students. This skeleton includes the information necessary for students to have the proper string formatting and proper utilization of the datetime module. The students are required to fill in the chained and nested conditionals in order to complete the remaining parts of the prompt.

Lesson Plan - 5E(+) Model

Engage: [6]

Engage the students with an anchoring event: a video clip from Flocabulary. This clip talks about how conditionals are used in computer programming by beginning with a definition of conditional statements, then shows the students a pseudocode, and concludes with proper Python syntax. The purpose of this clip is to get students thinking about the importance of computer programming and to engage them in a class discussion of "what are conditionals, and how are they used?." The following is a link to the website: Flocabulary: Conditionals. While the students are watching the video, they should be filling in the *Conditionals* worksheet independently. This should take less than 5 minutes to complete.

Explore:

Part I: Introduction

The students will be engaged in a whole class discussion after completing the worksheet, *Conditionals*, about how that video clip connects with the lesson plan's goal and objectives. Teachers will use students' responses to lead into the exploration's benchmark of the lesson by reviewing their answers and correcting any misconceptions as necessary. This section should take less than 10 minutes to complete.

Part II: *Benchmark Lesson*: Interacting with Conditional Statements
The instructor will hand out the worksheets titled *Letter Grade Program* and give each pair of students access to tablets or computers that run IDLE and the latest update of Python. The students will upload or copy the code from the *Letter Grade Generator Script* handout into the script mode of IDLE and complete the worksheet in pairs. By interacting with the given code, students will be able to answer the questions about what the chained conditionals are producing as an output. They will also be able to correct the semantic (logical) error. The purpose of this worksheet is to introduce conditional statements in Python and to emphasize the importance of syntax when writing code with chained conditionals. The worksheet forces students to engage with a chained conditional statement, which serves as a precursor to understanding the difference between "if" and "elif" and to understanding the purpose of the "else" syntax. Students will be engaged in a whole class discussion to correct and elaborate on the questions in the worksheet titled, *Letter Grade Program*. Here, teachers should expect to explain Python syntax rules when writing conditional statements. Teachers should also expect to explain the difference between the keywords "if", "elif", and "else." Also, teachers should use this discussion to correct students' responses on their worksheets and address any misconceptions as necessary. The students should be given 10 minutes to complete the worksheet in pairs, and the discussion should last no longer than 10 additional minutes. Hence, this section of the lesson should last less than 20 minutes.

Part III: *Investigation Lesson*: Programming
Students will work individually on the worksheet *Investigation and Programming*. This worksheet provides the students with the RAFT of the unit, which is written in the description section of this lesson plan. This worksheet contains a prompt that asks students to design their own program from scratch. This activity has them utilize built-in functions, variable assignments, the datetime module, string formatting, and chained conditionals. Since this curriculum is geared towards understanding the necessities of coding a SeaGlide, the students are not explicitly taught about the datetime module or string formatting. Hence, the worksheet provides the students with a skeleton of the code required to fully answer the prompt. The skeleton provides the necessary string formatting and datetime lines, as well as the necessary import lines. The students are responsible for recalling proper utilization of built-in functions (like print, input, and round) and of variable assignments. Students will also have to recall variable types by changing the inputs into strings and/or into floats. Further, students will have to utilize this prior knowledge and extend it into completing the given prompt by creating chained and nested conditional statements. The purpose of this part of the exploration is to have students create a Python code from scratch so they can work through a real-world example by developing the skills they have learned from the beginning of this curriculum. The remaining class time should be allotted for the students to complete writing this program. If the students do not complete the program within the remaining 20 minutes, it is recommended to have the students finish the program for homework and submit their codes prior to the next class meeting.

Explain:
Throughout the exploration, the students will engage in discussions that inquire their understanding and knowledge of the information at-hand. Thus, teachers will be informally asking students to explain

all topics and relevant connections throughout the entirety of this lesson. The worksheets include “why” and “how” questions to provide the teacher with their individual explanations and to also give the students an opportunity to verbalize their understandings. The exploration also includes questions that prompt students to defend their responses and explain why they chose a particular solution.

Elaborate:

The elaboration of this lesson is the investigation lesson section. Computer programming is currently a prominent and relevant career path that requires developers to tackle problems by debugging current code and writing new codes with helpful functions. The *Investigation and Programming* worksheet introduces the students to these programming skills. This student-designed program forces the students to think like a programmer by engaging in a real-world scenario.

Evaluate:

This lesson is designed with having both informal and formal evaluations throughout its entirety. The informal evaluations occur throughout the exploration because of the leading and open-ended questions and class discussions. This allows teachers to gauge surface-level student understanding. By surveying the students during completion of the worksheets, teachers will be able to hear and address any misconceptions or misunderstandings as necessary. The formal evaluation of this lesson is the *Investigation and Programming* worksheet. This individual activity forces students to write their own code from beginning to end by pulling on information learned throughout this lesson and both prior lessons.

Enrich:

This lesson could be extended by introducing how to utilize for and while loops. Since most functions contain complicated conditionals and nested loops, the next logical step is to have students interact and explore Python code that includes functions with for and while loops. Further, functions are most commonly used in industry when programmers realize that certain parts of codes are utilized in a multitude of programs. Hence, programmers tend to write a code containing functions that can be called and utilized in other program codes. Having students to learn for and while loops primes them to developing necessary skills that programmers use daily.

All associated documents are attached below

Reference *Annotated Bibliography* on the very last page of this packet

Name: ____ Date: ___________________

Conditionals [6]

1. Define Control Flow

2. ______________ are statements that run only when _____________________________.

3. What is the “gist” of conditional statements?

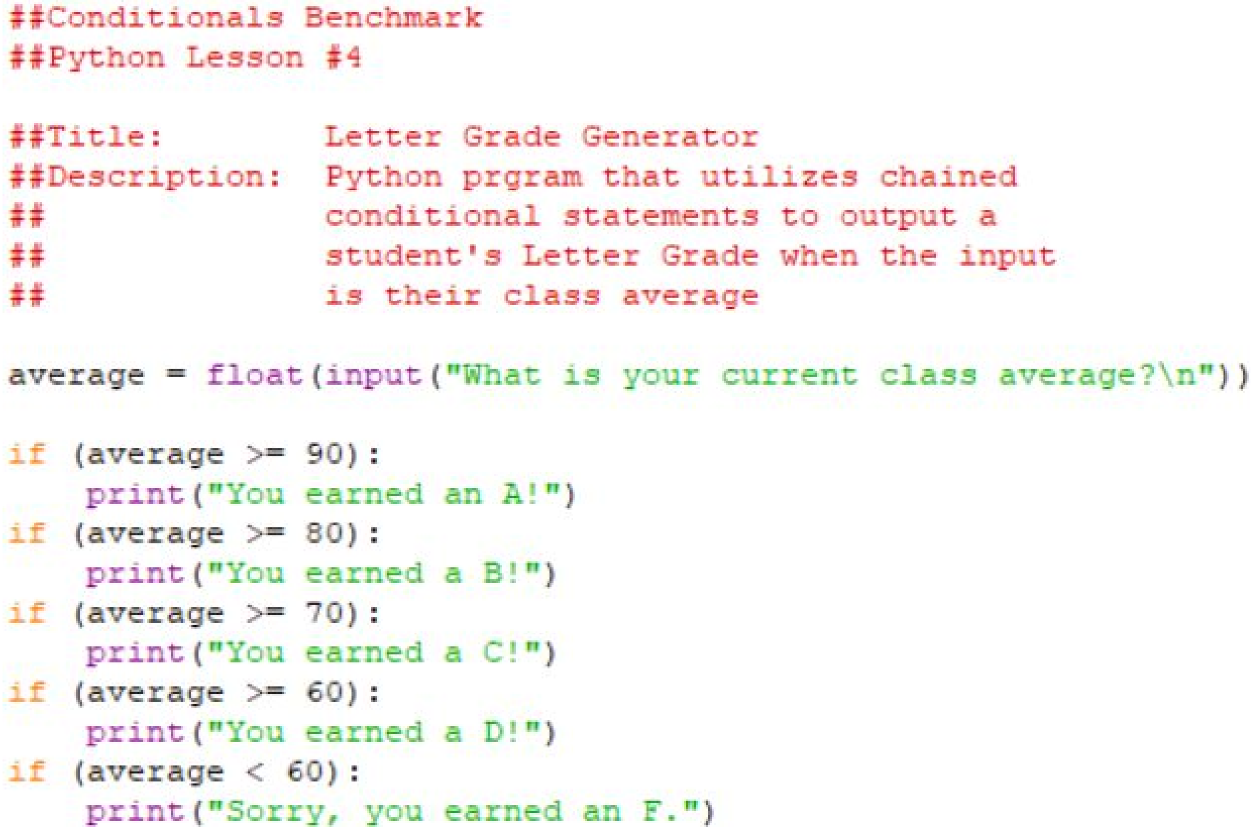
4. Conditionals tell a computer _____________________________________ if a certain

   condition is _____________. What happens when the condition is False?

5. Compare pseudocode to Python code.

Name: ____ Date: ___________________

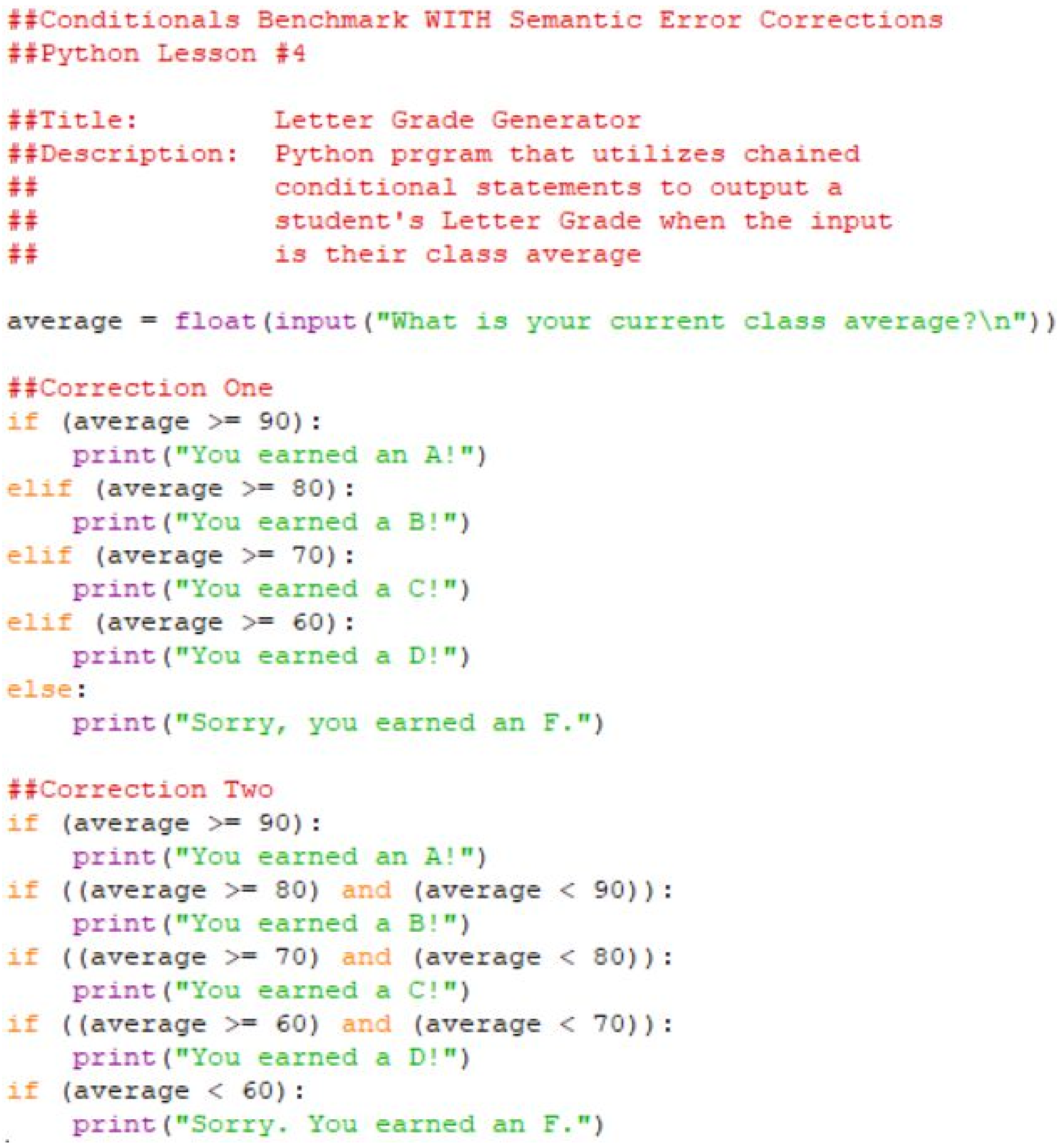
Letter Grade Generator: Script



```
##Conditionals Benchmark
##Python Lesson #4

##Title:        Letter Grade Generator
##Description:  Python prgram that utilizes chained
##              conditional statements to output a
##              student's Letter Grade when the input
##              is their class average

average = float(input("What is your current class average?\n"))

if (average >= 90):
    print("You earned an A!")
if (average >= 80):
    print("You earned a B!")
if (average >= 70):
    print("You earned a C!")
if (average >= 60):
    print("You earned a D!")
if (average < 60):
    print("Sorry, you earned an F.")
```

Name: ___________________Teacher Copy ONLY_________________ Date: ____________________

Letter Grade Generator: Script (Without Errors)



```
##Conditionals Benchmark WITH Semantic Error Corrections
##Python Lesson #4

##Title:        Letter Grade Generator
##Description:  Python prgram that utilizes chained
##              conditional statements to output a
##              student's Letter Grade when the input
##              is their class average

average = float(input("What is your current class average?\n"))

##Correction One
if (average >= 90):
    print("You earned an A!")
elif (average >= 80):
    print("You earned a B!")
elif (average >= 70):
    print("You earned a C!")
elif (average >= 60):
    print("You earned a D!")
else:
    print("Sorry, you earned an F.")

##Correction Two
if (average >= 90):
    print("You earned an A!")
if ((average >= 80) and (average < 90)):
    print("You earned a B!")
if ((average >= 70) and (average < 80)):
    print("You earned a C!")
if ((average >= 60) and (average < 70)):
    print("You earned a D!")
if (average < 60):
    print("Sorry. You earned an F.")
```

Name: ____ Date: ___________________

Letter Grade Program

1. What is the output when the following averages are used as inputs?
    a. 55
    b. 75
    c. 95

2. What <u>should have been</u> the output when the average of 75 was used as an input?

3. Why was the output when the average of 95 not what you expected? Hint: talk about the structure of the conditionals.

4. What kind of error does this code produce? Defend your response.

5. Debug and rewrite the code so that it produces the proper outputs. Provide your corrections by writing your cde below. (Hint: after you make code changes, run the code by testing at least three inputs). Explain WHY you made those changes.

Name: ____ Date: ________________

Investigation and Programming

Directions:

Your role as an assistant computer programmer means that you must learn to be proficient in many types of programming and coding. For this assignment you will have to continue learning Python by applying and extending your prior knowledge. Because you are in training, you will be working on several modules and continue to build upon your skills. You will work on various skills in Python and submit them to the head computer programmer to make sure your codes are both sufficient and efficient. For this task, you will learn and utilize conditional statements in Python by designing your own program.

Prompt:

The Weather Channel wants to incorporate providing custom messages to its users. They want the message to include the user's location, current temperature (in celsius), current precipitation, and a special warning. The special warning is unique to you, as the programmer. The Weather Channel hired you to write a Python code that can be programmed into their computer system so their users can log-in and receive their customized message.

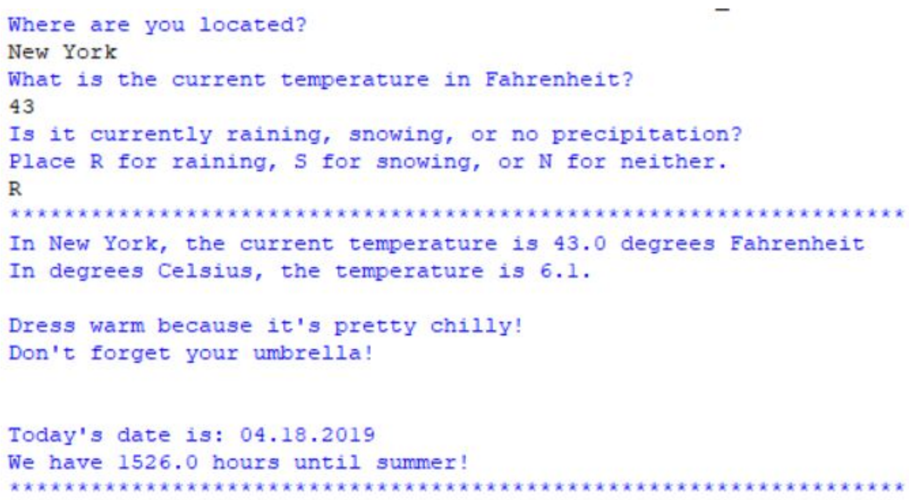
This involves using the input function, a formula to convert temperature, the datetime module, and if/ elif/ else statements. As the assistant computer programmer, you are asking the Weather Channel for information about the weather for a specific area (using the input function) and reporting this information to the public with special warnings (using the print function).

They provided the following rules:

1. Use input() to receive the user's location
2. Use input() to receive the temperature as degrees in Fahrenheit
3. Use input() to ask whether it is raining, snowing or neither (receive: R for raining, S for snowing and N for neither).
4. Convert the Fahrenheit temperature into degrees Celsius
5. Report the results using print(). This includes the following: the location, temperature in Fahrenheit, temperature in Celsius, and whether it is snowing or raining. Show the temperatures rounded to one decimal place.
6. Add a special warning! This means use print() to say "Take your umbrella!" if it is raining, or "Dress warm, it's cold!" if the temperature is below 30 degrees Fahrenheit. Be creative with these!
7. Add the current date to the output, and show the number of hours until the first day of summer -- June 21, 2019.
8. Use IF blocks to give an appropriate warning when it is raining, when it is snowing, and when it is really cold outside.

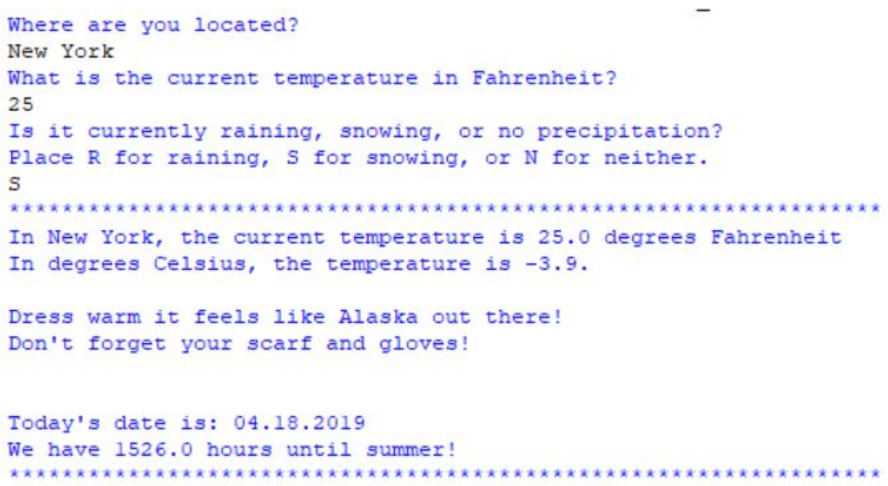
Example Outputs:

Example One:



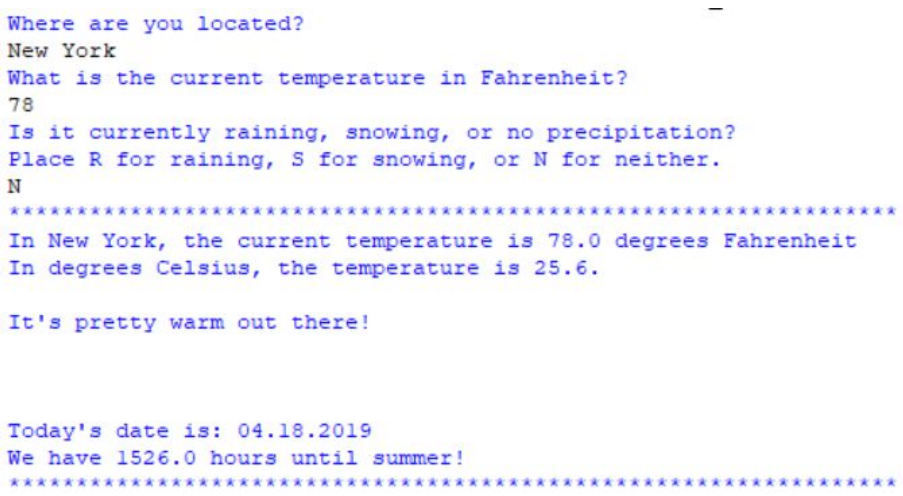
```
Where are you located?
New York
What is the current temperature in Fahrenheit?
43
Is it currently raining, snowing, or no precipitation?
Place R for raining, S for snowing, or N for neither.
R
******************************************************************
In New York, the current temperature is 43.0 degrees Fahrenheit
In degrees Celsius, the temperature is 6.1.

Dress warm because it's pretty chilly!
Don't forget your umbrella!


Today's date is: 04.18.2019
We have 1526.0 hours until summer!
******************************************************************
```

Example Two:



```
Where are you located?
New York
What is the current temperature in Fahrenheit?
25
Is it currently raining, snowing, or no precipitation?
Place R for raining, S for snowing, or N for neither.
S
******************************************************************
In New York, the current temperature is 25.0 degrees Fahrenheit
In degrees Celsius, the temperature is -3.9.

Dress warm it feels like Alaska out there!
Don't forget your scarf and gloves!


Today's date is: 04.18.2019
We have 1526.0 hours until summer!
******************************************************************
```

Example Three:



```
Where are you located?
New York
What is the current temperature in Fahrenheit?
78
Is it currently raining, snowing, or no precipitation?
Place R for raining, S for snowing, or N for neither.
N
******************************************************************
In New York, the current temperature is 78.0 degrees Fahrenheit
In degrees Celsius, the temperature is 25.6.

It's pretty warm out there!



Today's date is: 04.18.2019
We have 1526.0 hours until summer!
******************************************************************
```

Helpful Information:

Since we have not yet used the datetime module or talked about formatting, provided below is a SKELETON of what your code should be. Feel free to copy and paste or type this skeleton into the script mode of IDLE. You are responsible for filling in the necessary information to fully answer the prompt, i.e. replace the highlighted sections below.

```
##Name
##Weather Channel

##Imports
import os
import datetime as dt
import math

##Variable Declaration
now = dt.datetime.now()
date = now.strftime(‘%m.%d.%Y’)
summer = dt.datetime(2019,6,21) - now
hours = round((summer.days*24) + ((summer.seconds)/(60*60)),0)

##Inputs
location = (create input for the user’s location)
temperature = (create input for the current Fahrenheit temperature)
precipitation = (create input asking about precipitation)

##Celsius Conversion
Celsius = round(((temperature-32)*(5/9)),1)

##Output Temperature Statements
print(“In %s, the current temperature is %.1f degrees Fahrenheit” %(location, temperature))
print(“In degrees Celsius, the temperature is %.1f.\n” %(celsius))

##Chained Conditionals & Output Messages (This is the section where you write IF / ELIF / ELSE blocks to print your special messages. Remember, you need a message per precipitation condition AND also per temperature division. You get to decide how many temperature divisions that you want to use.)

##Output Date Statements
print(“Today’s date is: %s” %(date))
print(“We have %.1f hours until summer!” %(hours))
```

Name: ______________________ANSWER KEY_____________________ Date: _____________________

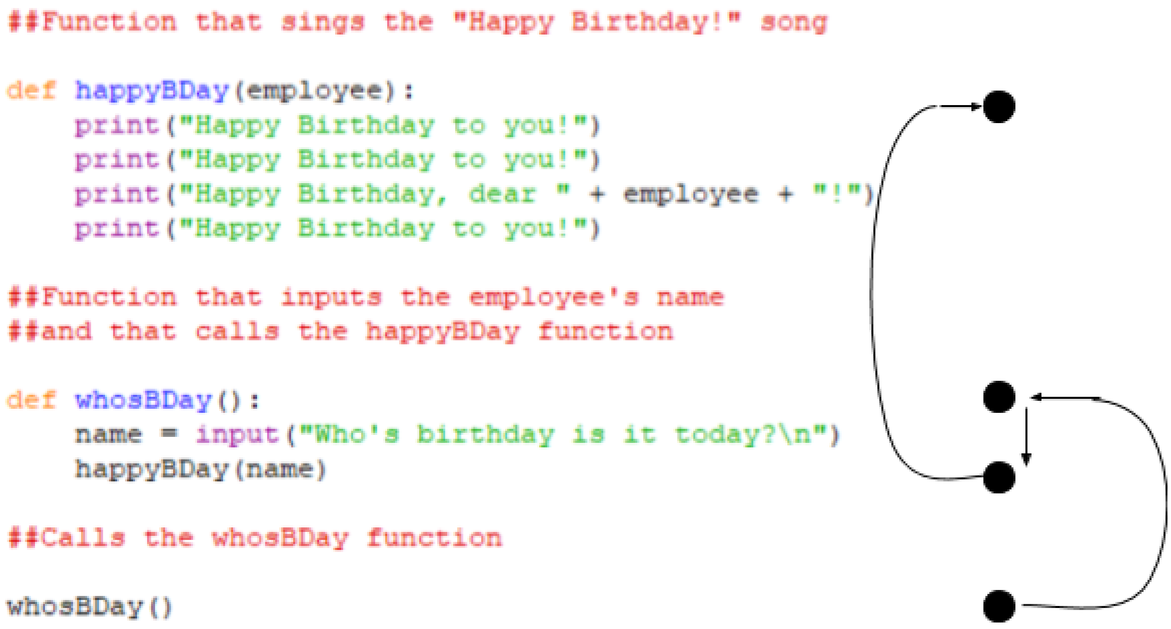
Conditionals [6], [7]

1. Define Control Flow

   The order in which a program carries out instructions.

   Discussion Note: the control flow is begins at the first line of the program and moves line-by-line until the program is complete. Conditional statements are only one example that can manipulate the control flow. This means, instead of reading down one line, that the computer would "jump" back and forth between the lines of the program.

   Discussion Question: What is another way to manipulate the control flow of a program?
   Goal Answer: User-defined Functions. Much like conditionals, user-defined functions make the control flow of a program "jump" to different parts of the program. Recall the prior lesson. The answer to the JP Morgan and Chase prompt utilizes two user-defined functions. One of which calls the other function. Below is a diagram of the control flow for the JP Morgan and Chase prompt:



```
##Function that sings the "Happy Birthday!" song

def happyBDay(employee):
    print("Happy Birthday to you!")
    print("Happy Birthday to you!")
    print("Happy Birthday, dear " + employee + "!")
    print("Happy Birthday to you!")

##Function that inputs the employee's name
##and that calls the happyBDay function

def whosBDay():
    name = input("Who's birthday is it today?\n")
    happyBDay(name)

##Calls the whosBDay function

whosBDay()
```

   Bonus Answer: loops (for and while) also manipulate the control flow because of the repetitive nature of reading portions of the program

2. ___Conditionals__ are statements that run only when ___certain criteria are met___.

3. What is the “gist” of conditional statements?

   To determine if the certain criteria are met, the conditionals tell a computer to check and to see if a condition is true. If a condition is true, then the conditionals tell a computer to do an action. If a condition is false, then the conditionals tell a computer to do a different action.

   Discussion Notes:

   ```
   IF condition
           Expression (action)
   ELIF condition
           Expression (action)
   ELSE:
           Expression (action)
   ```

   If the condition for if is False, it checks the condition of the next elif block and so on. The elif is short for else if. It allows us to check for multiple expressions. If all the conditions are False, body of else is executed.

   Only one block among the several if...elif...else blocks is executed according to the condition.

   The if block can have only one else block. But it can have multiple elif blocks.

   Discussion Question: Where do we see conditionals? This is an open ended question that has infinite answer possibilities. It is recommended to take an example from up to three students then move on for time purposes.

4. Conditionals tell a computer ___what to do___ if a certain condition is __true__. What happens when the condition is False?

   If a certain condition is false, the conditionals tell a computer to do a different thing.

5. Compare pseudocode to Python code.

   Pseudocode is a simplified version of programming that helps programmers develop better codes. Hence, writing in pseudocode before transitioning into Python makes the task of coding less daunting, and it helps alleviate the presence of semantic errors. Pseudocode provides the basic structure of what a program should say. This leaves translating the pseudocode into Python as the final step of developing a program.

   It is important to understand that pseudocode does not alleviate the presence of errors entirely. While using pseudocode as an initial step when programming, programmers still have to understand and utilize proper syntax during translation. Further, some lines of pseudocode may be more useful as a function call, which means that programmers need to write the corresponding function that is not originally involved within the pseudocode.

Name: ____________________ANSWER KEY___________________ Date: ___________________

Letter Grade Program

1. What is the output when the following averages are used as inputs?

    a. 55

        Sorry, you earned an F.

    b. 75

        You earned a C!
        You earned a D!

    c. 95

        You earned an A!
        You earned a B!
        You earned a C!
        You earned a D!

2. What should have been the output when the average of 95 was used as an input?

    You earned an A!

3. Why was the output when the average of 95 not what you expected? Hint: talk about the structure of the conditionals.

    By the way the code is written, the print statements are supposed to output when the condition is true. The first conditional is an IF statement where the condition to have “You earned an A!” as an output is for the average to be greater than or equal to 90. 95 > 90 so “You earned an A!” is part of the output. The next conditional is also an IF statement where the condition tests the truth value of the average being greater than or equal to 90. 95 > 80 so “You earned a B!” is part of the output. This trend follows because each conditional is an IF statement. Hence, the truth value of each conditional is checked by the computer, regardless of the truth value of the leading conditionals.

4. What kind of error does this code produce? Defend your response.

    This is a semantic, or logic, error because the program runs, compiles and produces an output. However, the output is not the expected or desirable output.

5. Debug and rewrite the code so that it produces the proper outputs. Provide your corrections by writing your cde below. (Hint: after you make code changes, run the code by testing at least three inputs). Explain WHY you made those changes.

   The corrected code is provided via the *Letter Grade Generator: Script (Without Errors)* teacher copy.

   Discussion Notes and Reasoning:

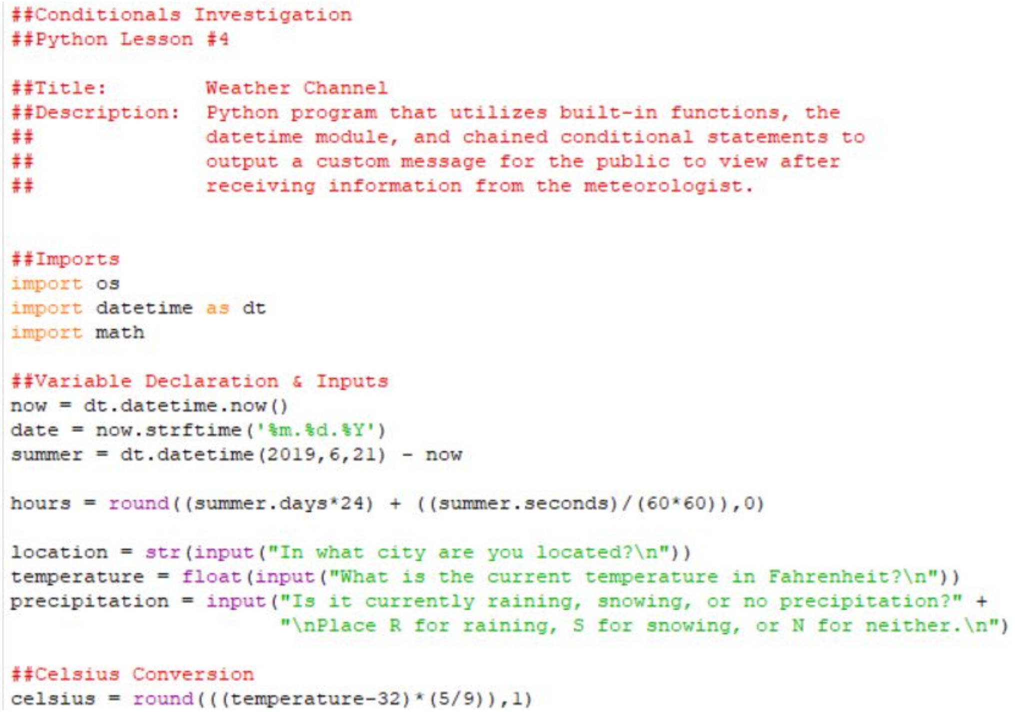
   Notice that the stacked IF statements have been changed to ELIF and ELSE statements. This corrects the semantic error because, once one of the conditionals is checked as true by the computer, then none of the other conditions are checked. IF/ ELIF/ ELSE structure is used when programmers want to use conditionals but exclusively one condition to be true.

   Notice that the program is written to start at 90. If the code was changed to start at 60 but still include the IF/ ELIF/ ELSE statements, then a logic error would still be present. The outputs would only be "You earned a D!" or "Sorry, you earned an F" because any input average is greater than, equal to or less than 60.

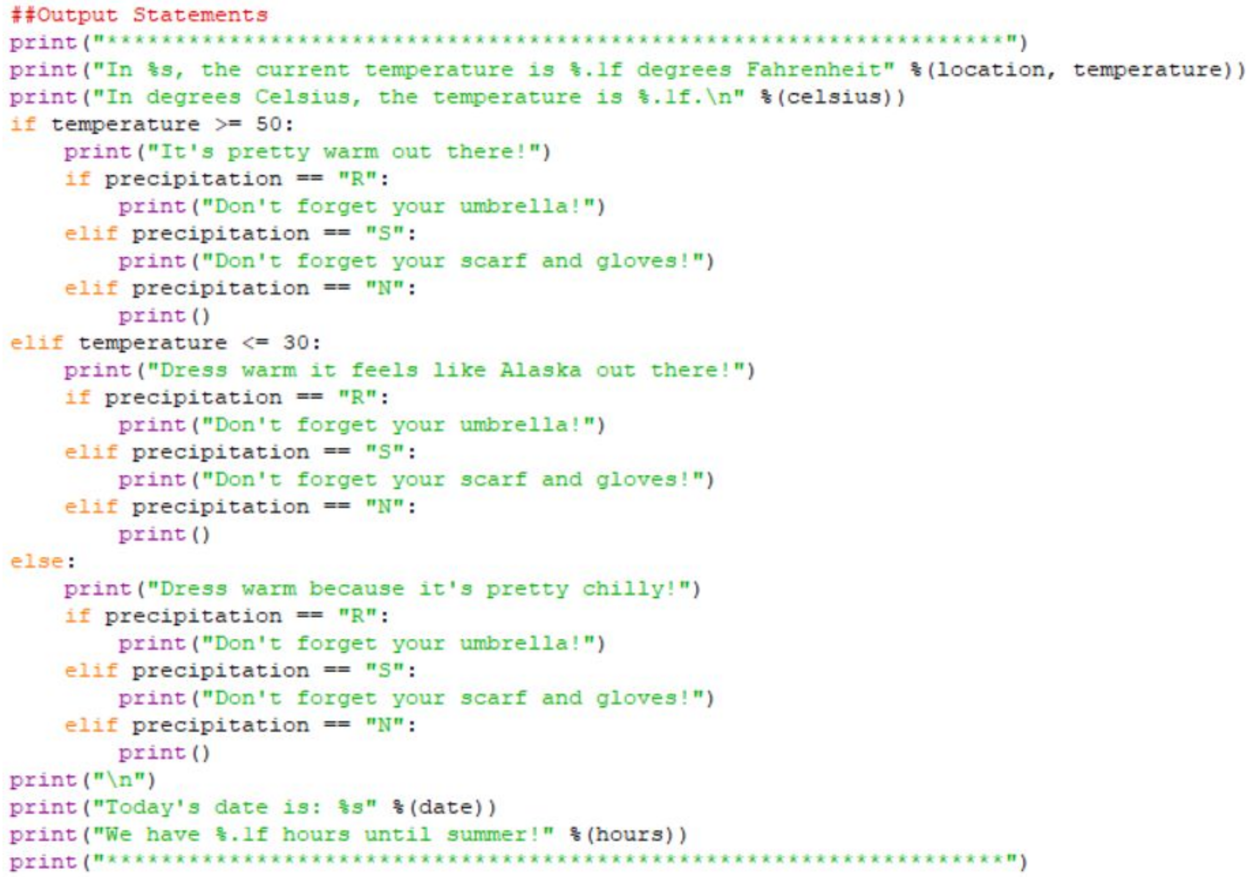
   Notice that there are two options for corrections provided. Students (if the syntax is done correctly) can circumvent having to use ELIF and ELSE statements by making the condition of each IF statement more complex. Since conjunctions have not been covered within this module, it is unexpected that students will gravitate to using the "and" operator.

Name: ______________________ANSWER KEY_____________________ Date: _____________________

Investigation and Programming **Students' codes will vary based on how they divide the temperatures**



```
##Conditionals Investigation
##Python Lesson #4

##Title:        Weather Channel
##Description:  Python program that utilizes built-in functions, the
##              datetime module, and chained conditional statements to
##              output a custom message for the public to view after
##              receiving information from the meteorologist.


##Imports
import os
import datetime as dt
import math

##Variable Declaration & Inputs
now = dt.datetime.now()
date = now.strftime('%m.%d.%Y')
summer = dt.datetime(2019,6,21) - now

hours = round((summer.days*24) + ((summer.seconds)/(60*60)),0)

location = str(input("In what city are you located?\n"))
temperature = float(input("What is the current temperature in Fahrenheit?\n"))
precipitation = input("Is it currently raining, snowing, or no precipitation?" +
                      "\nPlace R for raining, S for snowing, or N for neither.\n")

##Celsius Conversion
celsius = round(((temperature-32)*(5/9)),1)
```



```
##Output Statements
print("******************************************************************")
print("In %s, the current temperature is %.1f degrees Fahrenheit" %(location, temperature))
print("In degrees Celsius, the temperature is %.1f.\n" %(celsius))
if temperature >= 50:
    print("It's pretty warm out there!")
    if precipitation == "R":
        print("Don't forget your umbrella!")
    elif precipitation == "S":
        print("Don't forget your scarf and gloves!")
    elif precipitation == "N":
        print()
elif temperature <= 30:
    print("Dress warm it feels like Alaska out there!")
    if precipitation == "R":
        print("Don't forget your umbrella!")
    elif precipitation == "S":
        print("Don't forget your scarf and gloves!")
    elif precipitation == "N":
        print()
else:
    print("Dress warm because it's pretty chilly!")
    if precipitation == "R":
        print("Don't forget your umbrella!")
    elif precipitation == "S":
        print("Don't forget your scarf and gloves!")
    elif precipitation == "N":
        print()
print("\n")
print("Today's date is: %s" %(date))
print("We have %.1f hours until summer!" %(hours))
print("******************************************************************")
```

Annotated Bibliography

[1] Downey, A. (2012). *Think Python: How to Think Like a Computer Scientist* (Vol. 2.0.17). Needham, MA: Green Tea Press. From http://www.greenteapress.com/thinkpython/thinkpython.pdf

This online textbook was used for excerption within the Python Conditionals lesson plan as part of the Computer Science module. This reference aided in the completion of providing definitions for the key concepts and definitions sections and for the associated worksheets. This book was useful because of its layout and completeness. The lesson expands upon the material used form this book as it uses the material in the creation of worksheets and activities that are not provided in the textbook.

[2] Standards Aligned System. (n.d.). Retrieved from https://www.pdesas.org/

This website was used in each lesson in the Computer Science module to select proper Pennsylvania State standards, which are based in Common Core, that each lesson is centered around.

[3] Nsta. (n.d.). Science and Engineering Practices. Retrieved January 18, 2019, from https://ngss.nsta.org/PracticesFull.aspx

This website used in every lesson in the Computer Science module to find Standards for Scientific Practices that are applicable in each lesson.

[4] Nsta. (n.d.). Disciplinary Core Ideas. Retrieved from https://ngss.nsta.org/DisciplinaryCoreIdeasTop.aspx

This website was used in each lesson in the Computer Science module to select appropriate disciplinary core ideas set forth by the NSTA that are at the center of each lesson.

[5] Nsta. (n.d.). Crosscutting Concepts. Retrieved from https://ngss.nsta.org/CrosscuttingConceptsFull.aspx

This website was used in each lesson in the Computer Science module to selecting appropriate crosscutting concepts set forth by the NSTA that apply to each Python lesson.

[6] Coding: Conditionals. (n.d.). Retrieved from https://www.flocabulary.com/unit/coding-conditionals/

This video on Flocabulary is used as an engagement in the Python Conditionals lesson plan as part of the Computer Science module. Questions were developed based on this video for students to answer.

[7] Python … if else Statement. (n.d.) Retrieved from https://www.programiz.com/python-programming/if-elif-else

Python by Programiz is an online tool that is useful for understanding the underlying concepts of computer programming in Python. The discussion notes for number three of the *Conditionals* answer key were excerpted to be utilized by the instructor during the open class discussion that covers the differences between if/ elif/ and else keywords in Python programming.