**Presentation Notes:**

Slide 2: Python Data Types

1. List the 5 basic Python data types and the result of the sample program.

Slide 4: Float Variable Type

1. List the purpose and features of the float data type.
2. List 2 differences between a float and an int.

Slide 5: Float Operators

1. List the purpose and provide an example of the “int()” operator.
2. List the purpose and provide an example of the “float()” operator.

Slide 6: Modulus Operator

1. List the two results produced by division.
2. List the purpose and provide an example of the “%” operator.

Slide 7: Python Control using Floats

1. Do floats change the way IF statements and WHILE loops work?
2. Was the result of the sample program unexpected? Explain your answer.

Slide 9: Boolean Variable Type

1. What are the possible values of a Boolean variable?
2. Modify the sample program to assign the variable didPass a value of False. Write your code below.

Slide 10: Logical AND Operator

1. Complete the logic table below for the AND operator?

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A and B** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Modify the sample program to check for a level 2 mark (60 to 70). Write your code below.

Slide 11: Logical OR Operator

1. Complete the logic table below for the OR operator?

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A or B** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Modify the sample program if a valid mark must be greater than 30. Write your code below.

Slide 12: Logical XOR Operator

1. Complete the logic table below for the OR operator?

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A or B** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Modify the sample program if a valid mark cannot be both greater than 45 and less than 50. Write your code below.

Slide 13: Logical NOT Operator

1. Complete the logic table below for the OR operator?

|  |  |
| --- | --- |
| **A** | **not A** |
|  |  |
|  |  |

1. Solve the following Logical Statements.

|  |  |
| --- | --- |
| **Statement** | **Result** |
| not (True and False) |  |
| not (True or False) |  |
| True and not(True) |  |
| False or not(False) |  |

Slide 14: Booleans & Python Control

1. What part of the sample program demonstrates that Boolean variables can be used in place of conditional statements to control IF statements and WHILE loops?
2. What part of the sample program demonstrates that Boolean variables can be used in combination with conditional statements to control IF statements and WHILE loops?

Slide 15: Loops & Break Statement

1. Summarize the purpose of the BREAK statement in a WHILE loop.
2. Summarize when the BREAK statement may be used.

**Student Questions:**

1. Write a program that uses Boolean variables and logic to do the following:
   1. Read a high number and a low number from console input. The numbers must be less than 100

lowNum = int(input("Enter a low number: "))

highNum = int(input("Enter a high number: "))

currentCount = 0

inRange = False

if(lowNum and highNum <= 100):

num1 = lowNum

num2 = highNum

elif(lowNum > 100):

print("Please enter a low number less than 100")

elif(highNum > 100):

print("Please enter a high number less than 100")

* 1. Create a continuous loop that starts with a count of 1 and increases the count by 1 each time through the loop
  2. Checks whether the count is between the low number and the high number and assigns the result of the check to a Boolean variable called “inRange”.
  3. Prints the message “Number is in range” if the value of inRange is true.
  4. Breaks out of the loop when the count gets bigger than 100.

while (currentCount <= 100) :

print("Count = %d" % currentCount)

if(currentCount >= lowNum and currentCount <= highNum):

inRange = True

print("Number is in range")

else:

inRange = False

currentCount = currentCount + 1