**Presentation Notes:**

Slide 1: IF Statement Control

1. Summarize the purpose of an ***if*** statement in programming.  
   an IF statement is used to provide decision control in computer programs  
   it allows programs to make decisions

1. Explain what happens in the sample program under the following conditions.
   1. classSize is greater than 30 45>30

it says you need more 15 more computerss

* 1. classSize is less than 30 20>30

it says we can start the lesson

* 1. classSize is exactly equal to 30 30.30

it says we ca start the lesson

1. Draw a sketch of the flowchart diagram   
   for an ***if*** statement.

Slide 2: Indentation & Code Blocks

1. Summarize the purpose and use of a code block in programming with respect to:
   1. Grouping conditional code statements
   2. Use of indentation

Any indentation shows a group of line together

Slide 3: Conditional Expression

1. Summarize the Syntax (format and location) of a conditional expression.

Enclosed brackets

1. Modify the sample program to check for a class size greater than or equal to 28.   
   Write your new conditional expression below.

Change 30s to 28s

Slide 4: Is Equal To (==)

1. Summarize the difference between ***==*** (is equal to) and ***=*** (assignment operator).

Greater than, or equal

1. Explain what the ***!=*** comparison means.

It checks for not equal to

Slide 5: ELSE Statement Control

1. Summarize the purpose of an ***else*** statement in programming.  
   commands to be done when the if condition is false

Provides a second option

Allows the program to say “do this otherwise do that’

1. Explain what happens in the sample program under the following conditions.
   1. classSize is greater than 30 40.30

it says you need 10 more computers and lets start our lesson

* 1. classSize is less than 30 20.30

it says your computers are ok lest start our lesson

* 1. classSize is exactly equal to 30 30.30

it says your computers are ok lets begin our lesson

1. Draw a sketch of the flowchart diagram   
   for an ***else*** statement.

Slide 6: elif Statement Control

1. Summarize the purpose of an ***elif*** statement in programming.  
   each ELIF has its own condition

Conditions are checked in order code is executed for the first one that is true

1. Explain what happens in the sample program under the following conditions.
   1. classSize is greater than 30
   2. classSize is less than 30
   3. classSize is greater than 14
   4. classSize is less than 14
2. Draw a sketch of the flowchart diagram   
   for an ***elif*** statement.

Slide 7: Program Comments

1. Summarize the purpose and syntax of a ***line*** comment.

Begins with# everything after the # is a comment

Used to explain a specific line of code

1. Summarize the purpose and syntax of a ***block*** comment..

Begins with three quotations

Can span multiple line

Used to explain large blocks or sections of code

Slide 8: Conditional Loops

Summarize the purpose of an ***loop*** statement in programming.  
  
A loop is a programming function that iterates a statement or condition based on specified boundaries

1. Explain why you should not use “cut-and-paste” to repeat code blocks.

You wouldn’t because if you make a mistake in a code block and you had copy pasted for example 20 of them you would have to go back an fix all 20 because of the mistake and its too much work

1. Draw a sketch of the flowchart diagram   
   for an ***loop*** statement.

Slide 9: While Loops

1. Explain how changes in the variable start Count affect the operation of the while loop.  
   the

1. Explain what happens to the while loop under the following conditions.
   1. When the condition statement is true
   2. When the condition statement is false

Slide 10: Infinite Loops

1. Explain what an infinite loop is and why it is a bad thing in a computer program.

1. Explain the difference between a Logic Error and a Syntax Error.

1. Explain the difference between a Logic Error and a Run-Time Error.

**Student Questions:**

Use the following code as a template for generating random numbers in your programs.

# This code generates a random number between 0 to 9

# Stores the value of the random number in the variable "targetNumber"

# Then prints out the result

import random

targetNumber = random.randint(0,9)

print("The random number %d is the target." % targetNumber)

1. Create a basic level program to implement a simple guessing game. Your program should do the following: (Provide a listing of your program below.)
   1. Generate and store a random number using the template code above
   2. Ask the user to guess the number (i.e. us the “input” command and sore the answer)
   3. Use an “if” statement to check if the guess is correct
   4. Print “Your guess was correct!” if the guess is correct
   5. Print “Try again” is the check is guess is wrong

import random

guess = int(input("Number between 0 and 9:"))

targetNumber = random.randint(0,9)

if (guess == targetNumber) :

print ("You guessed correctly")

elif (guess != targetNumber) :

print ("TRy again")

1. Create a medium level program to implement a better guessing game. Add the following to your basic level program: (Provide a listing of your program below.)
   1. Use “if” and “elif” statements to check if the guess is correct
   2. Print “Your guess was correct!” if the guess is correct
   3. Print “guess higher” if the guess was less than the target number
   4. Print “guess lower” if the guess was greater than the target number

import random

guess = int(input("Number between 0 and 9:"))

targetNumber = random.randint(0,9)

if (guess == targetNumber) :

print ("Your guess was too high")

elif (guess < targetNumber) :

print ("Your guess was too low")

1. Create an enhanced level program to implement the ultimate guessing game. Add the following to your medium level program: (Provide a listing of your program below.)
   1. Add a loop to keep playing the game until the target number is guessed correctly
   2. Exit the loop when the target number is guessed correctly

import random

guess = int(input("Number between 0 and 9:"))

targetNumber = random.randint(0,9)

while(guess != targetNumber) :

if (guess == targetNumber) :

print ("You guessed correctly")

elif (guess > targetNumber) :

print ("Your guess was too high")

elif (guess > targetNumber) :

print ("Your guess was too low")

guess = int(input("Number between 0 and 9:"))