CSC101 Assignment

Vibe Coding

Overview

In this assignment, you'll build upon your Python skills by adding new features to your personal profile. You'll explore how conditionals and functions work together to create more powerful programs, then deconstruct existing code to see how all the pieces fit together. Finally, you'll try "vibe coding" - using AI as your coding partner to bring your own creative program ideas to life.

Assignment Preparation

Make a copy of this document to record your answers. Rename your copy of the document to **lastname_vibe**, where **lastname** is your last name. You will **place all answers and screenshots** in this Google Document. The steps below will tell you what to put in your answer document - use the spaces provided to insert your answers accordingly.

Part 1: Personal Profile (Continued)

In this part of the assignment you will further enhance your Personal Profile program (from a previous assignment) by adding a few more programming concepts including conditionals and functions.

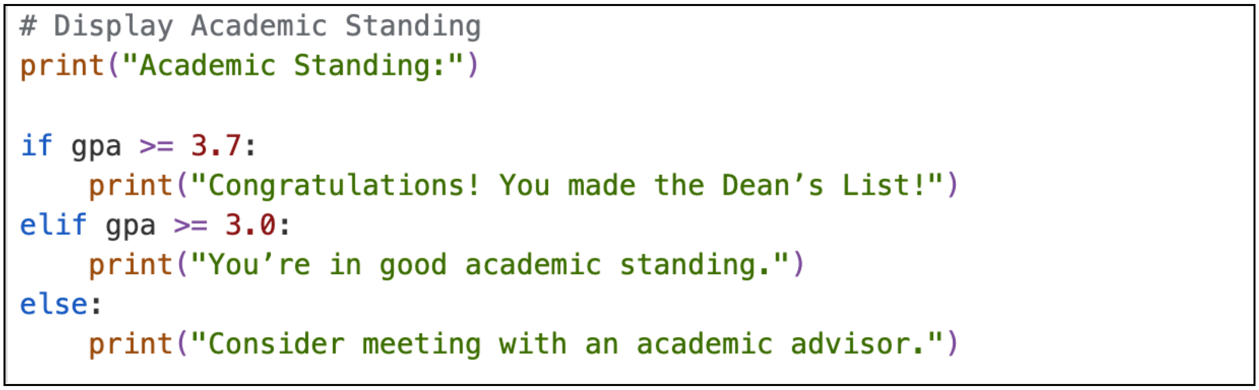
Conditionals

▶️ Watch: If Statements [4:38 min]

▶️ Watch: If-elif-else [4:46 min]

DO THE FOLLOWING:

- ➔ Open your “Profile v2” program from the previous assignment (you can find it on your saved Programs page)
- ➔ Review this sample code that adds a new section to display your academic standing in your profile. You will add similar code to your program in the following steps.



```
# Display Academic Standing
print("Academic Standing:")

if gpa >= 3.7:
    print("Congratulations! You made the Dean’s List!")
elif gpa >= 3.0:
    print("You’re in good academic standing.")
else:
    print("Consider meeting with an academic advisor.")
```

- ➔ With your “Profile v2” program still open, add a comment that starts a new section where you will add your academic standing, for example:

```
# Display Academic Standing
```

- ➔ Add a print statement that prints the text “Academic Standing:”, for example:

```
print("Academic Standing:")
```

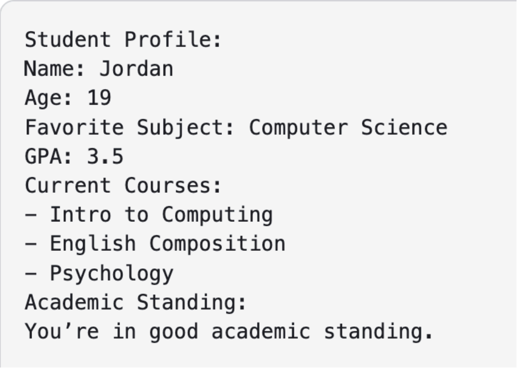
- ➔ Add an if-elif-else statement that prints your academic standing based on your gpa, for example:

```
if gpa >= 3.7:
     print("Congratulations! You made the Dean’s List!")
elif gpa >= 3.0:
     print("You’re in good academic standing.")
else:
     print("Consider meeting with an academic advisor.")
```

- ➔ Test your program by changing the gpa values to see the different messages. For example, try 3.8, 3.2, and 2.5

- ➔ Be sure to fix any bugs in your program and save it as you make changes. Your output should look similar to the example shown here.

- ➔ Run your program and take a screenshot showing the working code and the output



```
Student Profile:
Name: Jordan
Age: 19
Favorite Subject: Computer Science
GPA: 3.5
Current Courses:
- Intro to Computing
- English Composition
- Psychology
Academic Standing:
You’re in good academic standing.
```

[Insert screenshot here]

Part 2: Functions

Now, let’s turn your Academic Standing code into a function that can be reused throughout your program!

📖 Read: What are Functions in Python?

▶️ Watch: Functions [4:41 min]

DO THE FOLLOWING:

- With your “Profile v2” program still open, add a function at the very top of your program for your Academic Standing, for example:

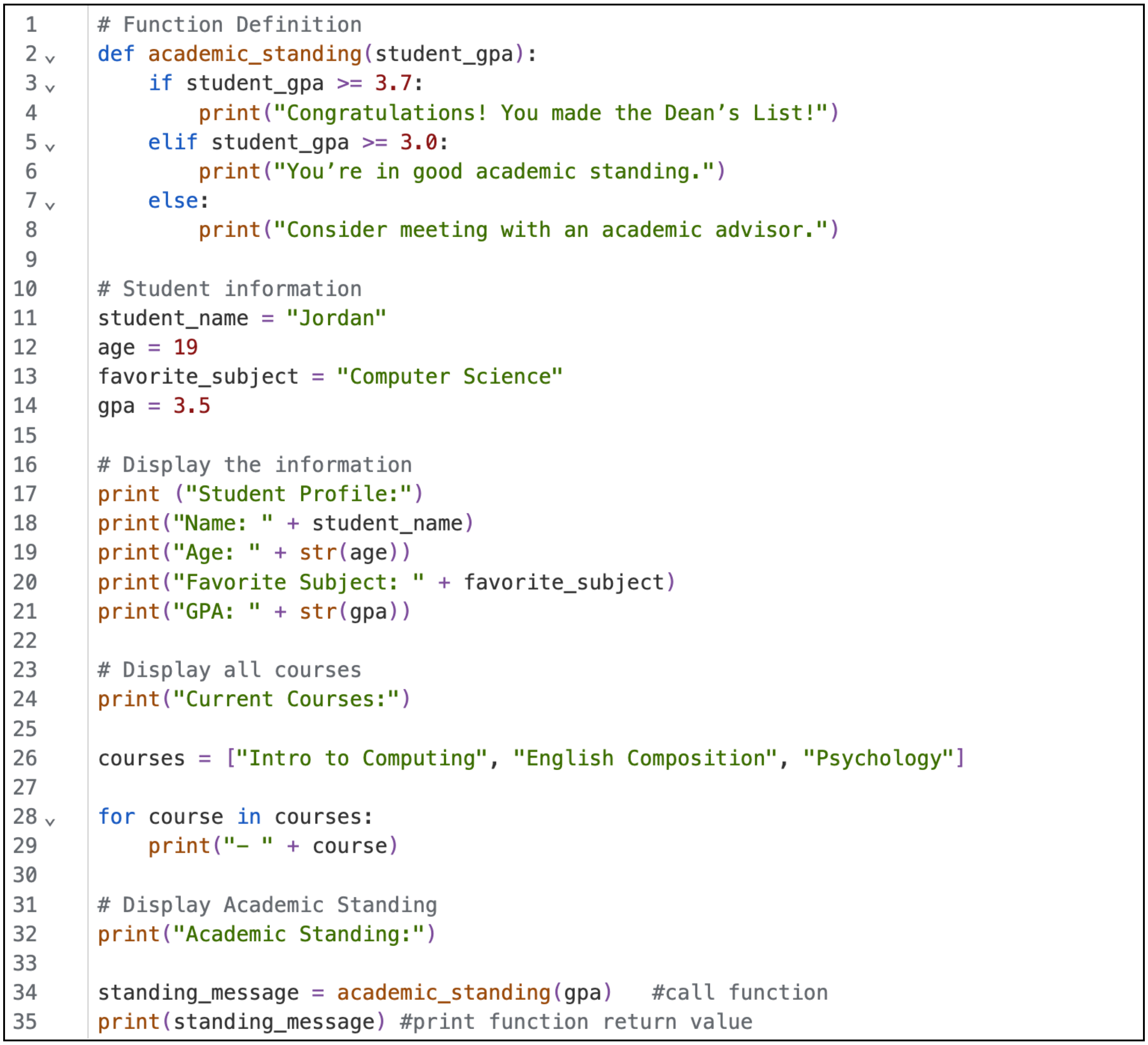
```
# Function Definition
def academic_standing(student_gpa):
     if student_gpa >= 3.7:
          print("Congratulations! You made the Dean’s List!")
     elif student_gpa >= 3.0:
          print("You’re in good academic standing.")
     else:
          print("Consider meeting with an academic advisor.")
```

- Replace the old if-elif-else Academic Standing section with a function call and print the return value, for example:

```
# Display Academic Standing
print("Academic Standing:")

standing_message = academic_standing(gpa)  #call function
print(standing_message)    #print function return value
```

- Your fully functioning program code should look similar to the example below:



```python
# Function Definition
def academic_standing(student_gpa):
    if student_gpa >= 3.7:
        print("Congratulations! You made the Dean’s List!")
    elif student_gpa >= 3.0:
        print("You’re in good academic standing.")
    else:
        print("Consider meeting with an academic advisor.")

# Student information
student_name = "Jordan"
age = 19
favorite_subject = "Computer Science"
gpa = 3.5

# Display the information
print ("Student Profile:")
print("Name: " + student_name)
print("Age: " + str(age))
print("Favorite Subject: " + favorite_subject)
print("GPA: " + str(gpa))

# Display all courses
print("Current Courses:")

courses = ["Intro to Computing", "English Composition", "Psychology"]

for course in courses:
    print("- " + course)

# Display Academic Standing
print("Academic Standing:")

standing_message = academic_standing(gpa)   #call function
print(standing_message) #print function return value
```

→ Test your program by changing the gpa values to make sure it still works.

→ Be sure to fix any bugs in your program and save it as you make changes.

→ Copy the URL of your working program and paste it below.

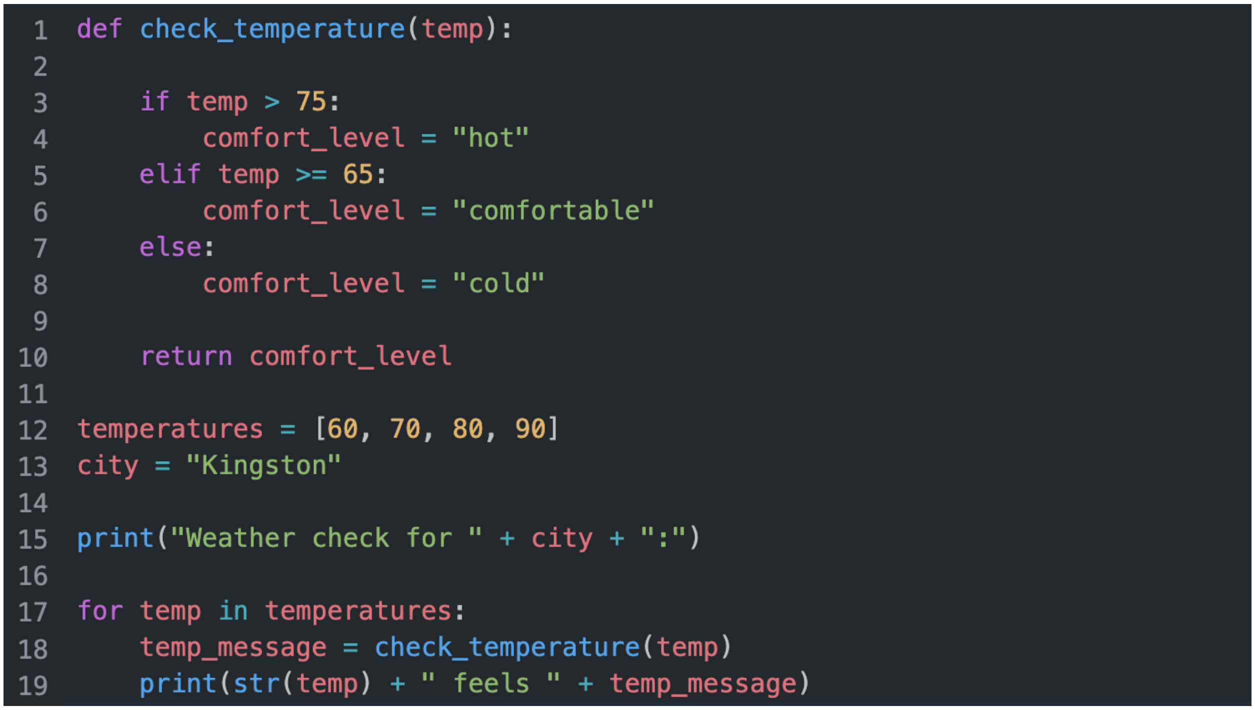
Link:	

Part 3: Program Deconstruction

You just created a more complex program by adding small pieces of code to a simple program until your program was finalized. Now, let’s deconstruct a program into all of the programming concepts we’ve learned about so far.

DO THE FOLLOWING:

- ➔ Review the sample code below:



```
def check_temperature(temp):

    if temp > 75:
        comfort_level = "hot"
    elif temp >= 65:
        comfort_level = "comfortable"
    else:
        comfort_level = "cold"

    return comfort_level

temperatures = [60, 70, 80, 90]
city = "Kingston"

print("Weather check for " + city + ":")

for temp in temperatures:
    temp_message = check_temperature(temp)
    print(str(temp) + " feels " + temp_message)
```

- ➔ Explain the following in your own words:

Identify where a list is used.	
Where is the function defined?	
What is the function name?	
What parameter does the function take?	
What value does the function return? Is the value an integer or a string?	
Where is the conditional statement used?	
Describe what the conditional statement does.	
Where is a loop used?	

Describe what the loop does.	

- What would happen if you change the city name to the city you were born in? Be specific in what the console would display when you run the program.

- Choose a temperature from 0-100. Explain what would happen when you add that temperature to the list. Be specific in what the console would display when you run the program.

- Choose an additional comfort level like “cool” or “freezing”. Explain in your own words how you would add the new level to your program.

Part 4: Vibe Coding

With just a few programming concepts now in your toolbox, you can begin to use AI to create your own programs and build upon your foundational programming skills. **Vibe coding** is when you describe what you want a program to do in plain English, and an AI assistant helps you write the code. It's like having a coding partner who can translate your ideas into Python! This is a powerful way to bring your creative ideas to life while learning how code works.

Use the AI assistant in Google Colab to create a simple Python program of your own design. You'll practice describing what you want clearly and understanding the code that gets generated.

▶️ Watch: Vibe Coding Demo []

DO THE FOLLOWING:

- Open Google Colab and click “New Notebook” in the prompt window.

This will create a new Colab Notebook in your Google Drive, where you can write and run code. Gemini AI is integrated into the notebook so that you can use AI to assist you as you code.

- Rename your file to Lastname_VibeCoding (where Lastname is your own lastname).
- Access the AI helper by clicking the Gemini toggle button on the bottom of the screen. ✦

Brainstorm:

- Think of a simple program you'd like to create. Keep it focused on ONE main task.

Example Ideas:

- Study Buddy Quiz Generator - Creates a random quiz from a list of questions and checks answers
- Daily Affirmation Generator - Displays a random positive message and asks if you want another one
- Recipe Scaler - Takes ingredient amounts and multiplies them based on number of servings
- Word Scrambler Game - Scrambles a word from a list and lets you guess the original
- Playlist Randomizer - Shuffles and displays songs from your playlist with song numbers

It's important to keep your program simple so that you can understand, tweak, and explain it. Below are some tips for constraining your program.

✅ Keep These Constraints:

- Uses 3-5 variables maximum
- Includes 1-2 loops (for or while)
- Uses 1-2 lists
- Has 2-4 conditional statements (if/elif/else)
- Includes 1-2 functions you define
- Runs in 20-40 lines of code (not counting comments)
- Completes in under 1 minute when running

❌ Avoid These (Too Advanced):

- Reading or writing to files
- Using libraries beyond `random` or `time`
- Complex math or algorithms
- Multiple functions calling each other
- Nested loops or complex data structures

Prompting:

- Describe your program to the AI assistant using the following template:

```
Create a simple Python program that [MAIN PURPOSE].
```

```
The program should:
- [Feature 1]
- [Feature 2]
- [Feature 3]

Use only these programming concepts: variables, loops, lists,
conditionals, and functions. Keep the code under 40 lines and
add comments explaining the code.
```

- Send your prompt to the AI assistant.

- Review the generated code to make sure it aligns with what you asked for, then Accept & Run the program.

- Test the program by trying different inputs. You can run the program again by clicking the play button. ▶

- If the program doesn't work properly ask the AI to fix specific problems (e.g. “The loop runs forever, can you fix it?”)

Understand & Modify

- Review the code to check your understanding. If there's something you don't understand, ask the AI assistant to explain it to you. Ask yourself:
    - Can I identify the variables, loops, lists, conditionals, and functions?
    - What does each part of the code do?
    - What happens if I change a number or text string?

- Make at least 2 small modifications. Some examples include:
    - Change a message displayed to users
    - Adjust a range of numbers
    - Add another feature
    - Personalize the data

- Share your Google Colab Notebook and make sure the access allows for “Anyone with the link can view”

- Provide your program information below:

Program Title:	
Program Description:	

Modification #1:	
Modification #2:	
Colab Notebook Link:	

Reflection:

➔ Consider how the concept of Vibe Coding may be used by anyone in the real world. Describe one positive aspect and one negative aspect for using AI to generate program code.
