HW 4 - Functions + Conditionals + Loops

Due February 25th, 2026 at 11:59pm

Overview

In this homework assignment, you'll practice with lists and dictionaries, build more functions, apply basic debugging strategies and begin branching with GitHub.

Note: If you need help fetching a branch from the `course_assignments` repository, please reference **Section 6: Bonus Content** in the [**Class 8**] slides on bCourses. Feel free to also look through **The GGG**.

1 Debugging

As you begin writing more code, you'll start encountering more complex errors. Having worked with GitHub, you're probably already familiar with how tedious debugging can be.

Here are some helpful debugging strategies:

- **Read the error message.** It may seem obvious, but reading the error carefully can tell you the line where the error occurred, the type of error, and sometimes even how to fix them explicitly.
- **Use `print()` statements.** If you're unsure what your code is doing, add `print()` statements to trace the problem. Especially around the line where the error occurred.
- **Start small.** Don't try to fix the whole function at once. Tackle smaller parts first and then build up to the whole function or script.
- **Take a break :)** Sometimes stepping away and letting your subconscious work on a problem in the background can be very helpful.

Throughout this homework, when you encounter an error, add a comment in your code explaining:

- What the error was.
- How you fixed it.

Include at least three of these error explanations. Here's an example:

```
print("Hello, World!")

"""
I encountered this error:
```

```
"SyntaxError: unterminated string literal (detected at line 1)
I originally wrote: print("Hello, World!)
I forgot the closing quotation mark. I fixed it by adding a " at the end.
"""
```

2 Create a Script

Inside your `yourname/` repository, create a folder named `homework4/`. Within that folder, create a file called `homework4.py`.

3 Lists

3.1 List Operations

Create a list of your **five favorite foods**:

1. Print the **second food** in the list.
2. Print the **last food** using **negative indexing**.
3. Add a **new food** to the end of the list using `.append()`.
4. Insert "apple" at the **start** of the list.
5. Remove the **third item** in the list using `del` or `.remove()`.
6. Print the **length** of the list with `len()`.
7. Loop through the list and **print each food in uppercase.**
   Hint: Use `.upper()`.
8. Create a **new list** containing only the **first and last** food (use slicing).
9. Use an if statement to check if "potato" is in the list. If it is, print `"A potato!"`. Otherwise, print `"No potato!"`.

3.2 Slicing and Striding

Create a list of numbers from **0 to 20**, inclusive, using the `range()` function. Store it in a variable called `numbers`.

Now, define and **call the following functions in sequence**, passing the output of one function into the next:

1. `get_first_15(numbers)`
   Takes a list of numbers and **returns the first 15 elements**.
2. `get_every_5th(lst)`
   Takes the list from `get_first_15()` and **returns every 5th element** from it.
   (So index 0, 5, 10...)
3. `reverse_and_stride(lst)`
   Takes the list from `get_every_5th()` and:
   - Reverses it

- Then returns **every 3rd element** of the reversed list

Example: Putting it All Together

```
step1 = get_first_15(numbers)
step2 = get_every_5th(step1)
step3 = reverse_and_stride(step2)
```

3.3 Nested Lists

A **nested list** is a list that contains other lists.

If a regular list is like a box, then a nested list is a box of boxes.

```
list_1 = [1, 2, 3]
list_2 = [4, 5, 6]
list_3 = [7, 8, 9]

numbers = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
```

3.3.1 Nested List Operations

Using the `numbers` list above:

1. Print the **third** row. *Hint: Use `list[row][column]` indexing.*
2. Print the **second item** in the second row.
3. Add `[10, 11, 12]` as a new row using `.append()`.
4. Write a function called `sum_nested()` that loops through each row, sums all numbers, and returns the total.

3.4 Create a 5x5 List

Write a function that uses **nested `for` loops** to create a 5x5 list of numbers from 1 to 25. Store the result in a new variable.

1. Write a function that replaces all **multiples of 3** with "`?`". Store the updated 5x5 list in a new variable.
2. Write a function that adds all elements **not equal to** "?" and returns the sum. Save the result in a variable.
   Hint: Use `!=` to skip "`?`".

Note: To clarify, edit the list with the first function and then edit it again with the second function.

4 Dictionaries

4.1 Dictionary Operations

Use the following dictionaries:

```
ages = {
    "Katie": 30,
    "Mariam": 42,
    "Safia": 25,
    "Mira": 48
}
```

1. Print "Katie"'s age.
2. Change Mira's age to 100.
3. Add ``Milana" with an age of 52.
4. Remove ``Mariam" from the dictionary.
5. Use a `for` loop to print out each person's name and age.

5 Running Your Code

5.1 VS Code

Please complete all the parts above before continuing.

5.2 In Your VS Code Terminal:

- Open VS Code.
- Open your `homework4.py` script.
- Pick your favorite function from this problem set.
- Include code at the bottom of your file that calls this function and prints the result.

- Name the screenshot: `hw4_vscode`.
- Save it in your `homework4/` folder.

5.3 On Your Terminal Application:

Please show the output of **all** of your functions.

- Open the terminal (Mac) or Git Bash (Windows).
- Navigate to your `homework4/` folder.
- Open a Python session with the command line.
- Import your file:

```
import homework4 as h4
```

- Run each function with example inputs.
- Take a screenshot of your **entire** script output.
- Name the screenshot: `hw4_commandline`.
- Save it in your `homework4/` folder.

6 Branching with GitHub

To access this question, pull the latest changes from the `course_assignments` repository. However, there is a twist. First, you will need to fetch a separate branch from the repository.

We have not covered **branching** in class. Please refer to the **Git and GitHub Guide** on bCourses for further information.

Let's walk through how to pull a branch from `course_assignments`.

1. Run `git branch`, you should see only `main`. This command shows you your local branches.
2. Run `git branch -a`, now you should see both local and remote branches.
3. Run `git fetch`, this downloads new remote branches.
4. Run `git branch -a` again, now you should also see `remotes/origin/surprise`.
5. Run `git checkout surprise`, now you're on a new branch
6. Run `git branch`, you should see a `*` next to `surprise`.
7. Take a screenshot of all these commands and name it `hw4_branch`.
8. Save the screenshot in your `homework4/` folder.

You are now on a different branch of the `course_assignments` repository. This is essentially a different version of the git repository. Then:

1. While on the `surprise` branch, go to `course_assignments/homework4` and find `surprise.py`.
2. Copy it to your `homework4/` folder.
3. Before you start working on the file in that folder, switch from the `surprise` branch back to the `main` branch. Start by calling `cd ..` to move back up to the parent directory.
4. Run `git branch`. You should see a `*` next to `surprise`.
5. Run `git checkout main`, now you have switched back to the `main` branch.
6. Run `git branch`, confirm you are back on the `main` branch by noting a `*` next to `main`.
7. Call `ls`. You should see the original `homework2/` and `homework3/` folders.
8. Edit `surprise.py`, run the code on the terminal and take a screenshot of the **entire** output.
9. Name the screenshot `hw4_surprise` and save it in `homework4/`.

7 Submitting Your Homework

After completing the above assignments and running it in VS Code and on the command line, follow the steps below to save your work and submit to Gradescope.

Steps:

1. Save your work by running the following commands inside your `yourname` repository.

   ```
   git add .
   git commit -m "done with hw4"
   git push origin main
   ```

   (Note: You may use any commit message you prefer.)

2. Take a screenshot showing your terminal of calling all of the above steps with their outputs.
3. Name the screenshot: `hw4_changes`.
4. Place it inside your `homework4/` folder.

5. Ensure all screenshots are saved correctly and your code runs without errors. Your `homework4/` folder should now contain:

   ```
   homework4/
   |--- homework4.py
   |--- surprise.py
   |--- hw4_branch.png
   |--- hw4_changes.png
   |--- hw4_commandline.png
   |--- hw4_surprise.png
   |--- hw4_vscode.png
   ```

6. Go to Gradescope and find the **Homework 4: Lists + Dictionaries** assignment.
7. Select the option to upload a GitHub repository.
8. Submit your `yourname` repository.

Great job!