COMP 110 - Problem Solving Assignment 1

Assignment Overview:

This assignment has one computational problem:

Correct a python program that has one or more turtles drawing your first name to do this correctly and to look professional.

Before you start coding, make sure you read through this whole document. If any of the instructions aren't clear, feel free to ask about it on CampusWire.

Important: You should not be using any coding concepts or language features that we have not yet discussed in class.

Pre-requisites

- Textbook Chapters 1 and 2, Sections 5.{1-4,6}, 7.1
- Successful completion of PSA0

Learning Outcomes

Upon successful completion of this PSA you will be able to…

- Debug autogenerated code
- Create an object and assign it to a variable
- Call a method from a given object
- Prompt a user for text input and use their response
- Use comments to help readers understand its functionality

Grading Synopsis

This assignment will be worth 9 points total, broken down as follows. More details about each of these points throughout the rest of this document.

1. **Uses Python Turtles:** (1 pt) - Uses python turtles to draw your name
2. **User input:** (1 pt) - Accept user input to set some aspect of the drawing
3. **Draw your name professionally:** (4 pts) - Correct the given code to draw your name (at least 5 letters) so it looks professional using the turtle module
4. **Style points:** (1 pts) - Add appropriate comments to the code to explain what is happening
5. **Share your drawing:** (1 pt) - Share a screenshot of your completed drawing on CampusWire or emailed to your instructor if you are not on campuswire
6. **Turn in your AI logs:** (1 pt) - Turn your your Copilot logs to show your chat for this project

Initial Setup

You'll need to start by using Git to get the starter code for this PSA.

1. If you haven't already done so, I recommend creating a folder either on your desktop or in your documents folder named "comp110". Throughout the semester, use this folder as the place where you put all of the code you write, including that for the PSA.

2. In VS Code, open the command palette and select the "Git Clone" option.

3. When prompted for the repository URL, enter the following.

   https://github.com/usd-comp110-assignments/comp110-starter-psa1.git

4. When prompted for where to save the repository, select the "comp110" folder you created (either on your Desktop or in your Documents folder). If you happen to get an error, make sure you click the "Git Log" button when prompted and look for the reported error message. Check CampusWire to see if anyone else got the same error message. If not, create a new question, copying and pasting the output of the Git log.

5. You will now have a folder named comp110-starter-psa1 opened in VS Code. In that folder, find the `name_drawer.py` file and open it. At this time, the file should be empty.

   If in the future you can't find this folder in VS Code, you can reopen it by going to "File" and selecting "Open Recent" in VS Code.

6. Ask Copilot to write out your name using turtles in python (1 pt). Please note that you need to include at least 5 letters in your drawing, so you may need to include your last name. Copy this code into the `name_drawer.py` file by applying it to the editor or copying and pasting it in.

7. Run the starter program by clicking the "Run" icon in VS Code. If you can't find that icon, instead open the VS Code command palette and select "Run python file in terminal". Make sure that the correct message was printed in the VS Code terminal window.

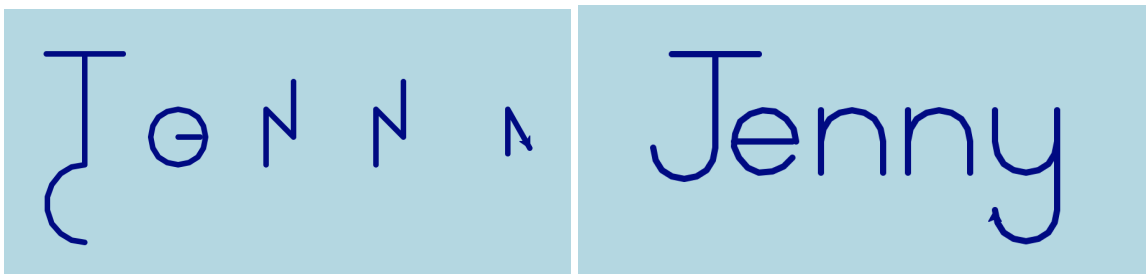
Note that for this PSA, you will be working individually to draw your own name. However, feel free to discuss your design with those around you and bounce ideas off of each other about how to make different letters.

Computational Problems

Turtle Name Drawing:

1. Inside of the `name_drawer.py` file, you have placed code written by a GenAI tool that writes your name. If you run this code, which you should do now, you will see that it is most likely not written correctly. This is a task that GenAI is not very good at!

   In this assignment, you need to fix this code to have it write your name correctly and in a way that looks professional (e.g., even spacing, consistent letter types, name capitalization). After running your program, your name should be drawn on the screen and look similar to the example shown below. You may **not** use the `write` method from the turtle class to write your name. (4 pts):



Example name drawing. The left is what was created originally by AI and the right is the fixed drawing.

   Note that the code you were given is using the turtle module. You cannot use a different module for this assignment.

   One additional requirement is that your program use the `input()` function to allow the user to customize the drawing to **change a color or size**. Your code currently does not do this and is something that you will need to add.

   For example, you might have them enter the color for the pen, the color of the window background, the size of the drawing, or some combination of these. You should give a clear input prompt so that the user knows what they need to enter and what it will affect. For example, "Enter the color to use for the pen" is good, while "Enter a color" is bad. (1 pt)

   While you are fixing your code, you may use resources within reason. You can ask Copilot to help to fix the code. You can look up resources on the internet. You can chat with a friend about the code and what may be going wrong. You CANNOT have someone else give you code or to make your corrections for you by either writing your code or telling you what to write. You also cannot use other Generative AI tools outside of Copilot. At the end of the assignment, you will need to turn in your chat logs (1 pt), so make sure you are saving them as you go. You can do this by going to command palette -> chat: export chat…

2. Properly comment your code and place the following information at the beginning (i.e. the "header") of your file.

    1. Your name.
    2. Your email address.
    3. The date you first created the file.

    Inside of your code, you should have a comment line right before each letter you draw, indicating that the following code will start to draw a specific letter (e.g. "# move to the correct spot and start drawing the letter 'A'"). You should be explaining the purpose of a section of code, not how it does it. This means you will lose points if you comment every line of code. (1 pt)

3. After you are done, run your program and take a screenshot of the Turtle window. Upload this to CampusWire under the "Share your Turtle Name Drawing" post. (1 pt)

Customizing Your Turtle's Shape and Color

By default, a Turtle has a black, arrow-shaped body and a black pen. If you wish to spice things up a bit, you can change both the look and color of your turtle(s).

Changing Colors

You can use the Turtle class' `color` method to change both the body and pen color. In its simplest form you can pass in a single parameter to this method that contains the name of a color (e.g. "pink").

```
# Assume that you have a turtle variable named bob.
# Sets bob's body and pen color to pink.
bob.color("pink")
```

Alternatively, if you have a very specific color in mind that doesn't have a standard name, you can use this online color picker to pick the color and get the "HEX" code of the color (e.g. "4282C2").

```
# Again, assumes bob is a turtle variable
# You must add a "#" in front of the HEX code
bob.color("#4282C2")
```

If you want to get extra fancy, you can give the `color` method two parameters, the first of which will be the pen color and the second of which will be the body color.

```
# Set's bob's pen color to red and body color to green
bob.color("red", "green")
```

Changing Body Shapes

If you want to make your turtle look like something other than an arrow, you can use the Turtle class' `shape` method. Unlike colors, there is a fairly limited set of built-in options, namely: "arrow", "circle", "square", "triangle", and… (wait for it…) "turtle". The following code demonstrates using this method.

```
# As usual, bob should already refer to a turtle object
bob.shape("turtle")
```

There's also a way to make the turtle use any image you want but it's more complex.

Submission Instructions

Important: To be safe, you should always run your code both on your computer and check the feedback from Gradescope.

Submit your code to the Gradescope assignment **psa-01,** please note that you should not be renaming this file

Now you need to download your chat log with Copilot to submit this. Open View, Command Palette, and search for "Chat: Export Chat…" Name this file `<username>-chat.json` (replacing <username> as appropriate). Save this in the PSA 1 folder.

Checking Your Results

Part of this assignment is automatically graded. You can view the results of this autograded part of the assignment by visiting Gradescope

Please note that this gradescope provides feedback on the complex parts of the assignment. **Having all green checkmarks does not guarantee a perfect score** as items like the quality of comments are checked manually and if you do not follow directions, you may be able to pass the tests but your code may not run or give the correct output when you test it. It is important that you always test your code first before checking the autograded results.