CS215: Introduction to Program Design, Abstraction and Problem Solving
(Spring, 2025)
Lab Assignment 1
(20 points)
Today's Date: Tuesday, January 14
Demonstration Due Date: the end of Lab2 class
Submission Due Date: Friday, January 24

The purpose of this lab assignment is

- to get familiar with Microsoft Visual Studio IDE.
- to compile and run your first C++ program.
- to follow Coding Style Guide criteria when writing code.
- to practice using input/output.
- to practice using variables, constants, fundamental data types and basic operations.
- to get familiar with assignment submission through Canvas.

Problem specification

If you are going to work on computers in Lab classroom or you have already installed Visual Studio Enterprise 2022 on your own laptop, you can skip **Part 0**.

Part 0: Install and activate Visual Studio Enterprise 2022 on your own Windows laptop

Get a free copy of MS Visual Studio C++ through Microsoft's **Azure Dev Tools for Education** .
For instructions for UK students, start with this URL:
https://www.cs.uky.edu/docs/users/msdnaa.html#microsoft-azure-dev-tools
Follow the instructions, install and activate Visual Studio Enterprise 2022 on your own windows laptop. ***During the installation, do apply "Product Key" as described in the above instructions, otherwise the software license will be expired in 90 days.***

Part 1: Visual Studio

1. Log in to the lab computer and start Visual Studio (**Start → All Programs → Visual Studio 2022**).

   Note if you see the popped window asking you to sign in to use the Visual Studio 2022, type your link blue account followed by @uky.edu and password to sign in, password is not needed from the lab computer since you log in to the lab computer with your link blue account. Unless you choose to sign out before exiting Visual Studio 2022, you do not need to sign in to use the IDE next time)
2. **Create a New C++ project** (see the screenshot at **Figure 1**)
   a. **File → New → Project...** or **Ctrl+Shift+N**
   b. Select **Empty Project** then press **Next**
   c. Configure your new project (see **Figure 2**)
      1) Project Name: In the **"Project Name"** text field, enter a name for your project, for example, "Lab1".

2) Location: you may leave it in the **…\repos** folder, or use another folder where you will keep all CS215 labs, projects, etc. **Remember this folder! You will need it to submit to Canvas.**
3) Check the "Place solution and project in same directory" box.
4) Click the **Create** button in the bottom/right.

3. Now your project will appear on the side of the window named "Solution Explorer" (see **Figure 3**), with folders for "Header Files", "Source Files", and so on.
4. **Add a new source file**:
   a. Right click on **Source Files** and select **Add → New Item...** (see **Figure 3**)
   b. Choose "C++ File (.cpp)".
   c. Enter a name for your file (for example, "Lab1.cpp" or "main.cpp" or leave it as the default "Source.cpp") and click **Add**. **Note: this is the file you will submit in Canvas.** Remember or make a note of the Location and File Name for future reference.

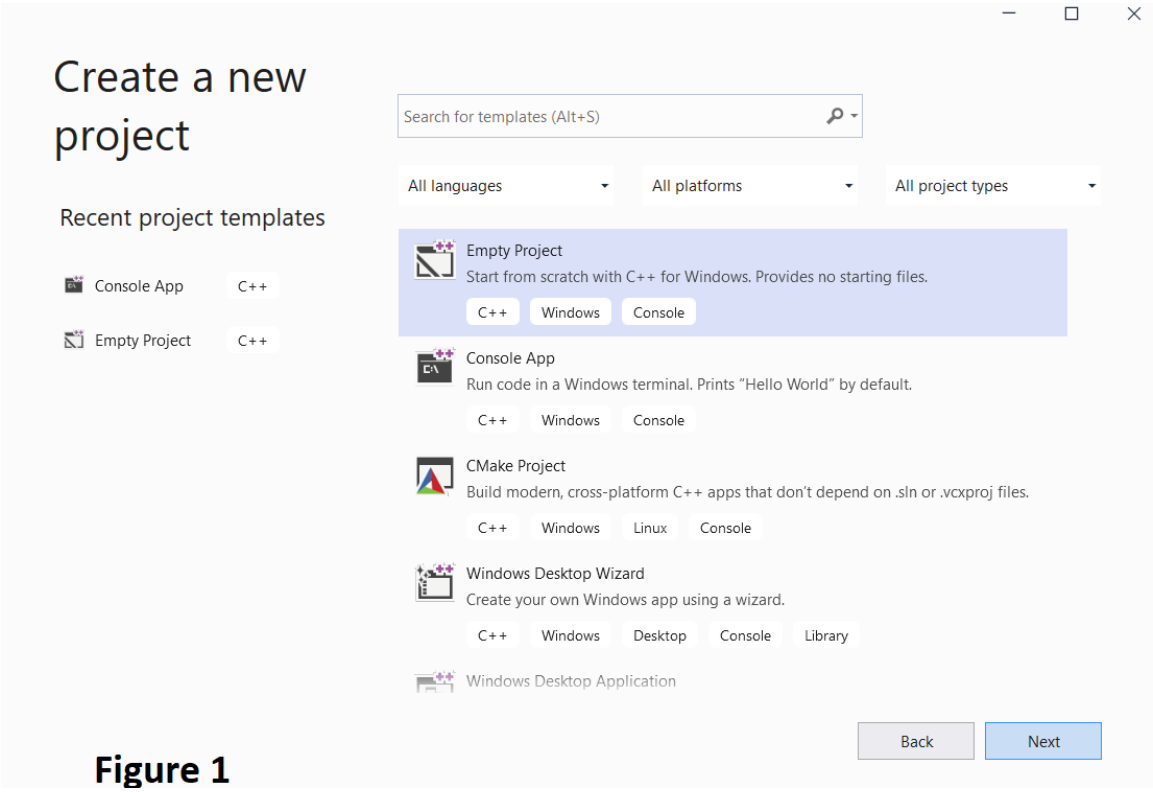

Figure 1

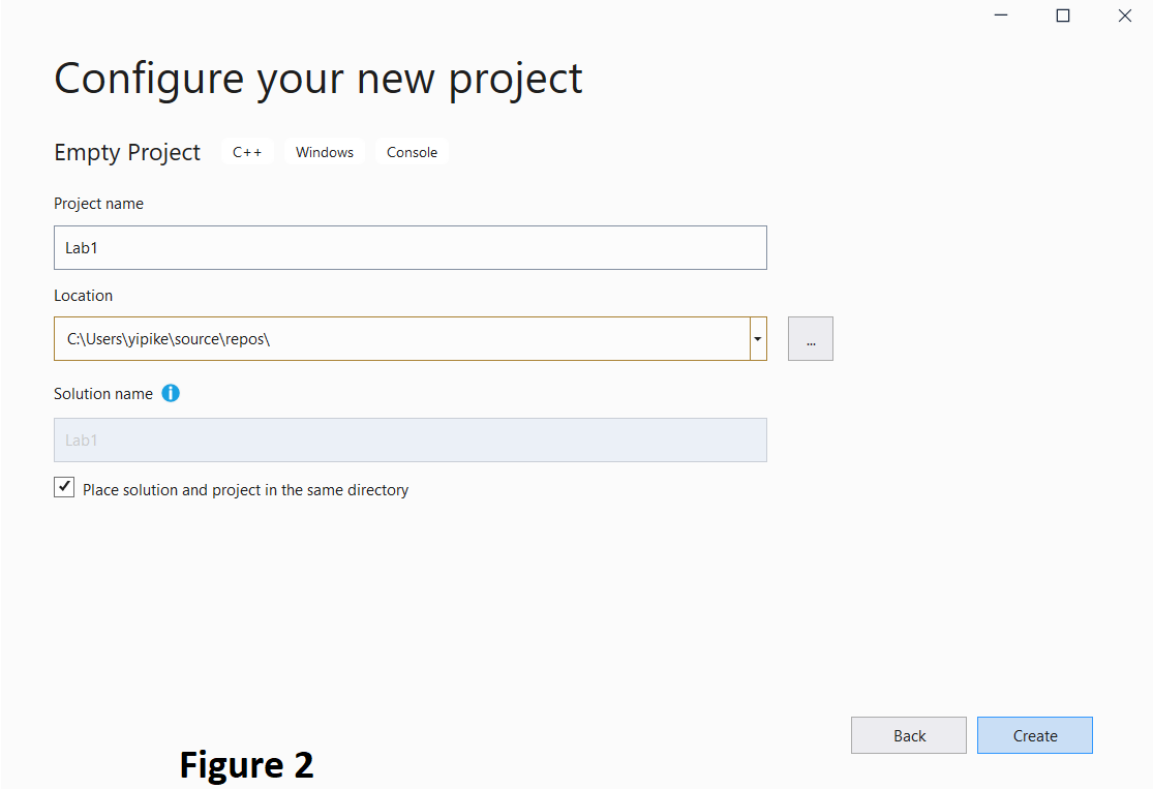

Figure 2

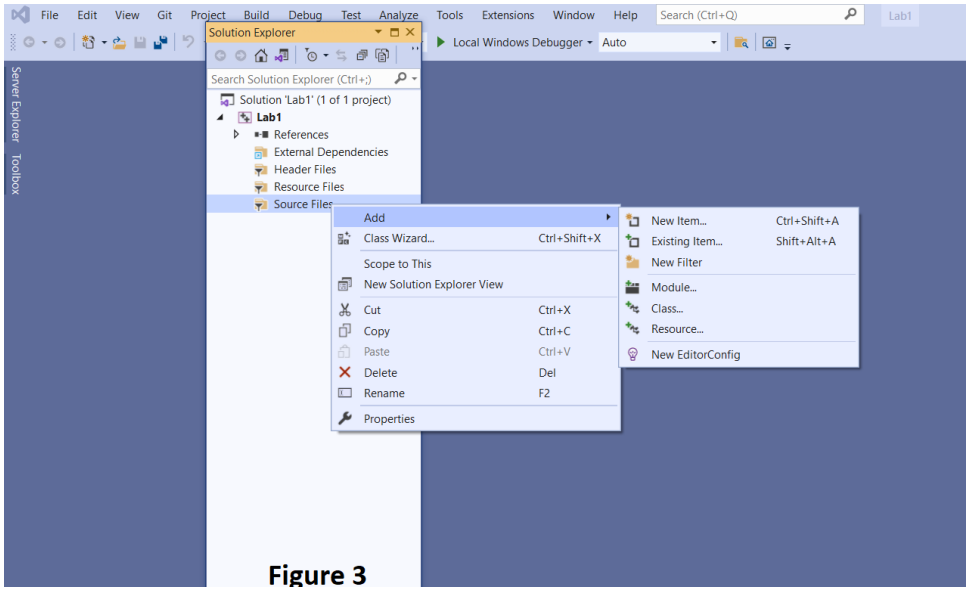

Figure 3

5. You will then by presented with an edit window. Enter your first C++ program to print “Hello, World!” to the computer screen and save.
6. Now let's try compiling:
    a. First, make sure you have saved your project (**Ctrl+Shift+S**).
    b. From the menu select **Build → Build Solution**, or press **Ctrl+Shift+B**. Check the "Output" tab at the bottom of the screen. It should say "1

succeeded, 0 failed". If not, check your program for errors, correct them, and try again.

7. Now run the program: from the menu select **Debug → Start Debugging**, or press **F5**. You will see the output of your program and ask you to press any key to close the window (shown in **Figure 4**).

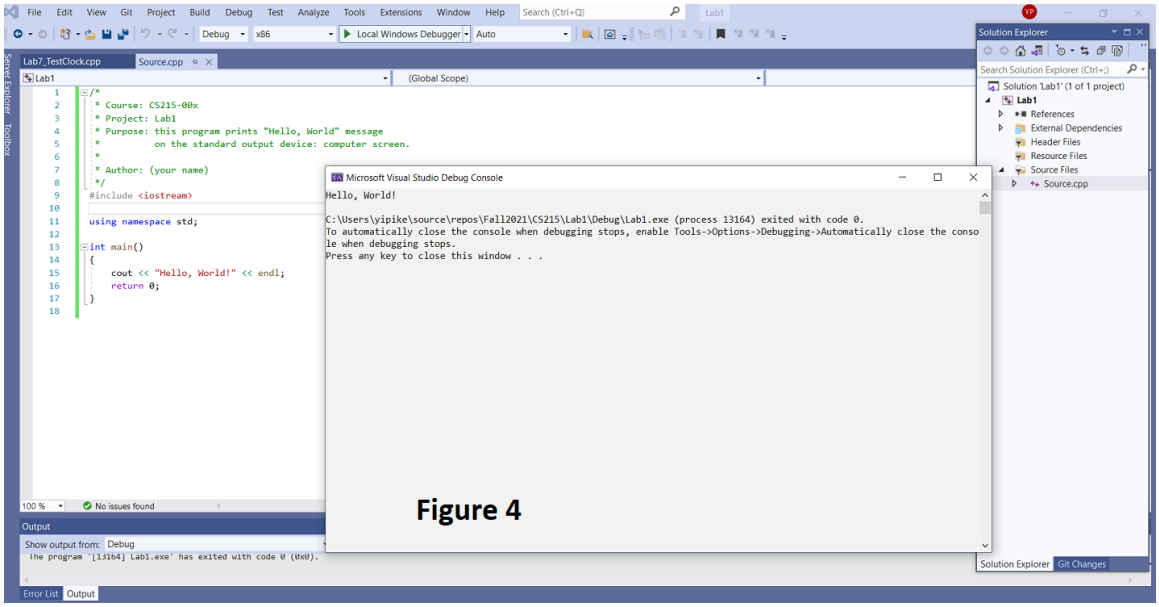

Figure 4

8. Go back to the edit window, and make some change to the .cpp file, so that your program should display three lines of message on the computer screen:
   Hi A,
   I am glad you are my TA.
   My name is B. Nice to meet you!
   Please note that you need to find out your Teaching Assistant' name to replace item A from the above example. Then use your name to replace item B. Remember to **save** your program.
9. You can try to run your program again by click "**Local Windows Debugger**" with a green arrow, then check if the required message has been displayed on the computer screen.

Part 2: Display some message and a snowman

1. Open the project you created from **Part 1**, modify the source file you created from **Part 1** so that your program will continue printing some message, together with a snowman as shown in **Figure 5**. (Note that you can print a different snowman with your own design)
2. Compile and run your program just as you do in **Part 1**.
3. Save your source file or simply press **Ctrl+S.**

A sample output is shown in **Figure 5**: (Assume your TA is Olaf, and your name is Yi Pike.)

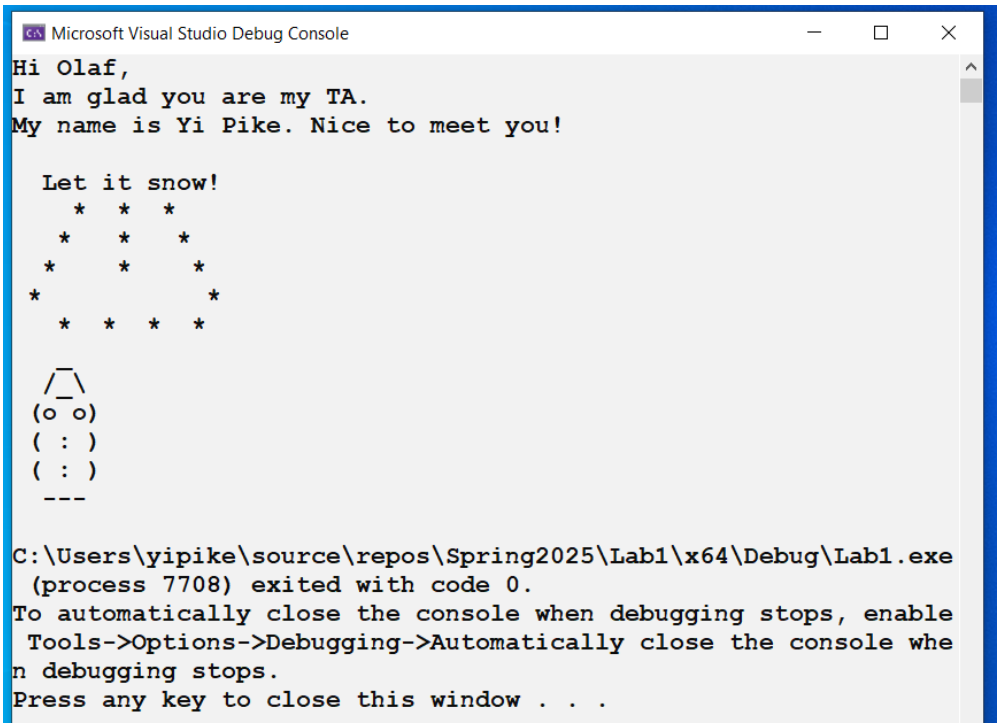

Figure 5

Part 3: Collect the user input and display the output.

1. Back to the edit window and display some user-friendly message to ask the user to type how many courses for today, then display the result.
2. Compile and run your program just as you do in **Part 2**.
3. Save your source file or press **Ctrl+S.**

A sample output is shown in Figure 6: (Assume the user types "2.5" as the input, then click the enter key.)

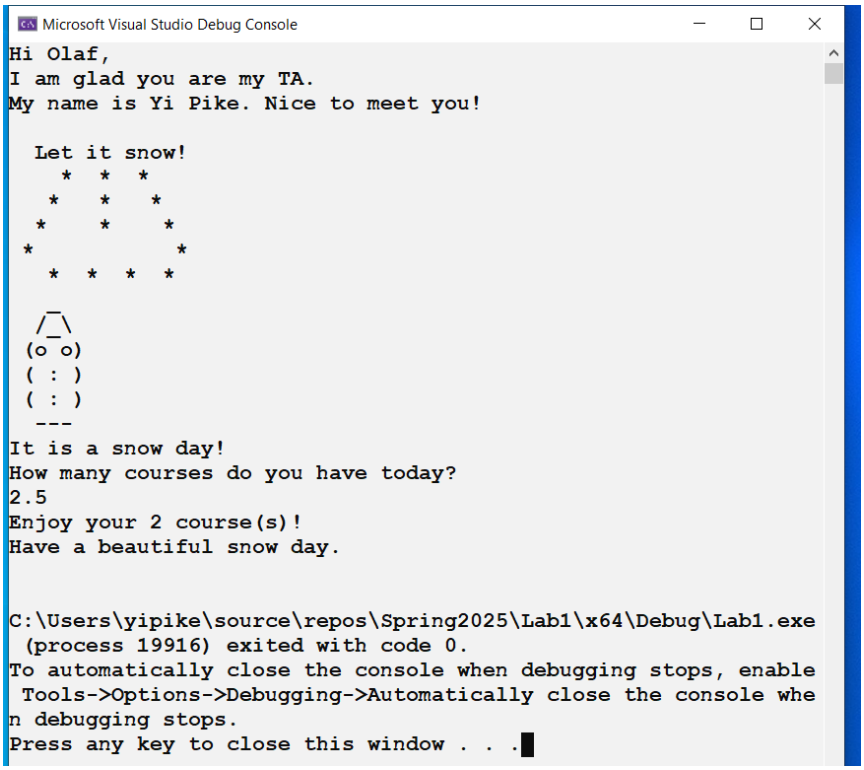

Figure 6

Part 4: Collect the user input again and do a little calculation.

1. Back to the edit window, and type the following code into your source file (insert this block before the return statement):

```
   //collect user input and calculate the time spent
   double hours = 0.0;      // declare a variable to store how many hours
   cout << "How many hours are you going to spend on CS215
every week?" << endl;
   cin >> hours;
   const int HOUR_TO_MIN = 60;  //declare a constant: 1 hour = 60 minutes
   const int MIN_TO_SEC = 60; //declare a constant: 1 minute = 60 seconds
   double seconds = hours * HOUR_TO_MIN * MIN_TO_SEC;
   cout << "Good Luck! You will spend " << seconds << "
seconds each week on CS215." << endl;
```

2. Compile and run your program just as you do in **Part 2**.
3. Save your source file or press **Ctrl+S.**

A sample output is shown in Figure 7: (Assume the user types “3” then “5.5” as the two inputs, then click the enter key.)

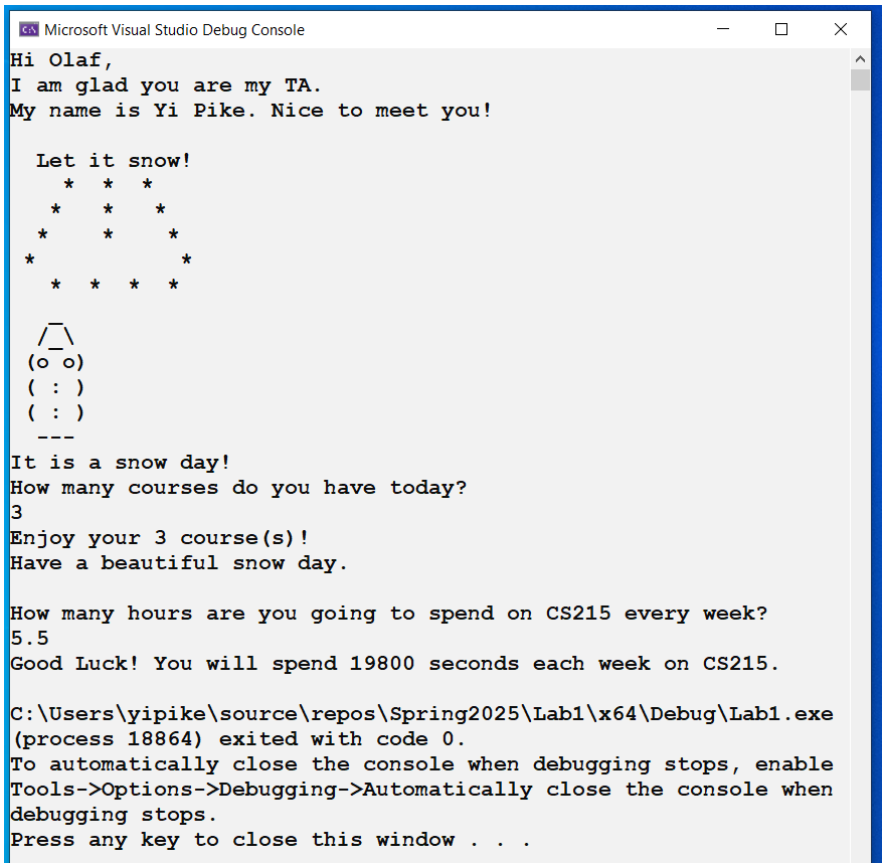

Figure 7

Congratulations! You are ready to demonstrate your program to your TA.

Part 5: Demonstrate your code to your TA, then submit it in Canvas

1. Each Lab assignment needs to demonstrate to your TA to be graded. You can demonstrate Lab1 during Lab1 class (with possible bonus 3 points) or no later than the end of Lab2 class (this is the **demonstration deadline** for Lab1).
If you finish Lab1 assignment during Lab1 class, you may demonstrate your program to your TA and answer your TA's questions, you can get up to 3 extra points for this lab assignment. (Note you can also demonstrate your program to your TA during Lab2 class. However, any demonstration later than the end of the Lab1 class cannot get bonus 3 points.)
If you need extra time, you can continue working on Lab1 assignment after the Lab class and try to finish it before the next Lab class. Then demonstrate your Lab1 during Lab2 class.
***If you do not demonstrate your code, even if you submit it in Canvas, you will receive a grade of 0!!** The TA may ask you to make some corrections. If so, make the corrections and demonstrate again…repeat until you have 100%!*
2. After the successful demonstration, submit the code on Canvas. Open the link to Course Canvas page (https://www.uky.edu/canvas), and log in to your account using your

LinkBlue ID and password. Please submit your file through link "**`Lab 1`**". See Part 1 step 4 (highlighted in blue) to find which file to submit.
Even if you successfully demonstrated it to the TA, if you do not submit it in Canvas by* the submission deadline*, you will receive a grade of 0!

Grading (20 points + Bonus 3 points)

1. Attend the lab session or have a documented excused absence. (**5 points**)
2. Demonstrate your program to your TA and submit it on Canvas. (**15 points**)

Demonstrate your program to your TA and answer TA's questions during Lab class when the same Lab assignment is given. (**Bonus 3 points**)

Programming tips and some observation:

- **Do Not Use Magic Numbers**

A magic number is a numeric constant that appears in your code without explanation. For example,

```
total_volume = bottle * 2;
```

Why 2? Instead use a named constant to make the code self-ducomenting:

```
const double BOTTLE_VOLUME = 2;
total_voume = bottle * BOTTLE_VOLUME;
```

There is another reason for using named constants. Suppose circumstances change, and the bottle volume is now 1.5 liters. If you use a named constant, you make a single change, and you are done. Otherwise, you have to look at every rule of 2 in your program and ponder whether it means a bottle volume, or something else. In a program that is more than a few pages long, that is incredibly tedious and error prone.

- **What happens if I input 2.5 for the first input of my program?**

Try to run your program again just like what you did in Part 3: after you type "2.5", observe what happens? The program is complete. Do you get a chance to type for the second time? What is the output? Does the output match what you expect? Trying to answer those questions. We will be able to explain why this happens when we explore more about variables and how they are stored in memory space, input statement in a few weeks.