

Exercise 7: Calculations and Rounding

Temperature Conversions

In this exercise, you'll be converting temperatures from Fahrenheit to Celsius and back again.

To convert from Fahrenheit to Celsius:

- Begin by subtracting 32 from the Fahrenheit temperature.
- Divide the answer by 9.
- Finally, multiply by 5 to get the Celsius temperature.

To convert Celsius to Fahrenheit:

- Begin by multiplying the Celsius temperature by 9.
- Divide the answer by 5.
- Finally, add 32 to get the Fahrenheit temperature.

Clone your repository

1. Click on the appropriate link and accept the assignment to create your repository for submitting your work:
 - a. Gallant AM: <https://classroom.github.com/a/XOYynMS4>
 - b. Gallant PM: <https://classroom.github.com/a/9iXMIDS0>
 - c. Nunn AM: <https://classroom.github.com/a/1VAfe7L>
 - d. Nunn PM: <https://classroom.github.com/a/fX6dM6v4>
 - e. Wijaya AM: <https://classroom.github.com/a/pccZ4OKt>
 - f. Wijaya PM: <https://classroom.github.com/a/Q-b7vLgR>
2. In GitHub Desktop, clone the repository you just created to your desktop.

Create your Unity project and prepare for GitHub tracking

3. Use Unity Hub to create a new 2D Unity project named **Exercise7**. Save the project in your new repository folder.
4. Once the project opens in Unity, go to File Explorer and move the **_UnityProjectRoot.gitignore** file into the Unity project folder and rename it to **.gitignore**
5. Go to GitHub desktop and commit your changes with the message: "Create initial Unity project" Make sure there are only about 30 files being committed.
 - a. If you have thousands of changed files, return to step 2 to make sure you've named the gitignore file properly and that it is placed at the root of the Unity project not in its original location. Ask for help if you are unsure.
6. Push your changes to the remote.

At this point you are ready to proceed with this assignment. We encourage you to make interim commits as you go. Use your commit message to indicate which step (e.g.: "Completed through step 5").

Problem 1 – Temperature Calculations with Int

7. In Unity Editor, rename **SampleScene** to **Scene0**.
8. Add a Scripts folder and add a new C# script named **ConvertTemperatures**.
9. Open the script, fill in the comment near the top of the script, and delete the **Update** method.
10. Add the following code to the **Start** method:
 - a. Declare three **int** variables to store an original temperature in Fahrenheit, a calculated temperature in Celsius, and a calculated temperature in Fahrenheit (when you convert from Celsius back to Fahrenheit).
 - b. Assign 0 as your original temperature in Fahrenheit and print that Fahrenheit value, convert to Celsius and print the Celsius value, then convert the Celsius value back to Fahrenheit and print the Fahrenheit value. Be sure to label each of your outputs rather than just printing out the numbers.
 - c. Assign 32 as your original temperature in Fahrenheit and print that Fahrenheit value, convert to Celsius and print the Celsius value, then convert the Celsius value back to Fahrenheit and print the Fahrenheit value. Be sure to label each of your outputs rather than just printing out the numbers.
 - d. Assign 212 as your original temperature in Fahrenheit and print that Fahrenheit value, convert to Celsius and print the Celsius value, then convert the Celsius value back to Fahrenheit and print the Fahrenheit value. Be sure to label each of your outputs rather than just printing out the numbers.
11. Attach the script to the Main Camera in the scene and run the game to see the output in the Console window.

You'll discover that the conversion from 0 to Celsius and back to Fahrenheit yields 5, not 0, for the final Fahrenheit temperature. That's one of the points of this exercise! Think about the data type you used and why you might get this result given that data type.

12. In GitHub Desktop commit your changes with the commit message: "Completed Problem 1"

Problem 2 – Temperature Calculations with Float

13. Add the following code to the **Start** method after the code you already added:

- a. Declare three **float** variables to store an original temperature in Fahrenheit, a calculated temperature in Celsius, and a calculated temperature in Fahrenheit (when you convert from Celsius back to Fahrenheit).
- b. Assign 0 as your original temperature in Fahrenheit and print that Fahrenheit value, convert to Celsius and print the Celsius value, then convert the Celsius value back to Fahrenheit and print the Fahrenheit value. Be sure to label each of your outputs rather than just printing out the numbers.
- c. Run the game to see the output in the Console window.

*Now the conversion from 0 to Celsius and back to Fahrenheit is closer to 0, but it's still not exactly 0. Remember, even though floating point numbers are approximations of the real numbers in the continuous domain, they give us more precision than whole numbers do. That's why we got a closer final Fahrenheit temperature when we used **float** variables instead of **int** variables.*

14. In GitHub Desktop, commit your changes with the commit message "Completed Problem 2"

Problem 3 – Temperature Calculations with Double

15. Add the following code to the Start method after the code you already added:
 - a. Declare three double variables to store an original temperature in Fahrenheit, a calculated temperature in Celsius, and a calculated temperature in Fahrenheit (when you convert from Celsius back to Fahrenheit).
 - b. Assign 0 as your original temperature in Fahrenheit and print that Fahrenheit value, convert to Celsius and print the Celsius value, then convert the Celsius value back to Fahrenheit and print the Fahrenheit value. Be sure to label each of your outputs rather than just printing out the numbers.

*Finally, the conversion from 0 to Celsius and back to Fahrenheit yields a 0 as you probably originally expected. Because the **double** data type uses more bytes than the **float** data type, we don't get the rounding error we got when we used **floats**.*

This exercise shows how data type really matters, especially when we do multiple calculations.

16. Go to GitHub Desktop and commit your changes with message: "Completed problem 3".

Submit Your Work

17. Make a final test of your code and copy the output from the terminal window.
18. If you need to make any additional changes to your code, make sure you commit them.
19. By committing and pushing your updates to GitHub you have submitted your assignment on GitHub Classroom.
20. Return to CodeHS. Paste your output into the code window to complete the assignment.