

CSCI 1301 – Lab 04

1 Reading From the User

1. Download the PersonalizedWelcomeMessage¹ project.
2. Extract it and open it in VS.
3. (If you are using VS on MAC, follow the instructions at <https://stackoverflow.com/a/49056993/> to have your project “Run on external console”. You will need to do that for every solution where the user is supposed to enter values.)
4. Compile and execute it.
5. You will be prompted with the message:

Please, enter your first name, followed by "Enter":

Enter your first name, followed by Enter . You just witnessed an interaction between a program and the user!

6. Read the source code carefully and make sure you understand all of it.
7. Change the code so that the program would also ask for the user’s last name and print both their first and last names.

2 Numerical Datatypes

For this part, it is recommended to have the cheatsheet² we discussed last time. Note that it contains numerous references at its end, you are encouraged to open those links if you have not already, to have a look at the official documentation, which should not scare you.

2.1 Literals and Variables

This part should be first carried out without using VS.

Assume we have the following statements:

```
int a = 21, b = 4;
float f = 2.5000000f;
double d = -1.3;
decimal m = 2.5m;
```

Answer the following:

- How many variables are declared?
- What are their datatypes?
- What are their values?
- What are their names?

¹<http://spots.augusta.edu/caubert/teaching/2020/fall/csci1301/weekly/04/lab/PersonalizedWelcomeMessage.zip>

²<http://spots.augusta.edu/caubert/teaching/2020/fall/csci1301/weekly/03/datatypes/>

2.2 Operations

- Consider the following expressions. For each of them, tell if they are legal and if so, give the result and its corresponding datatype. The first two are given as examples:

Operation	Legal?	Result	Datatype
a + d	Yes	19.7	double
m + f	No	—	—
a / b	—	—	—
b * f	—	—	—
d + f	—	—	—
d + b	—	—	—
a + m	—	—	—
f / m	—	—	—
d * m	—	—	—

You can check your answers using VS: create a new project, copy the variable declarations and assignments, and write your own statements to perform the calculations in the Main method. For instance, if you want to check that the result of a + d is of type double, write something like:

```
double tempVariable1 = a + d;
Console.WriteLine($"The value of d+f is {tempVariable1}");
int tempVariable2 = a + d; // This line should give you an error.
```

2.3 Cast Operator

Create a new project, and then do the following.

- Add in your program the following:

```
float floatVar = 4.3f;
int intVar = floatVar; // This statement will give you an error
```

You will get an error that reads

```
Cannot implicitly convert type 'float' to 'int'. An explicit conversion
↵ exists (are you missing a cast?)
```

Can you explain it?

- VS is suggesting that we use a “cast” to “force” C# to store the value of the variable floatVar into the variable intVar. To do so, replace the previous statement with the following:

```
int intVar = (int)floatVar; // This statement will compile
```

- Using a Console.WriteLine statement, observe the value stored in intVar. Can you tell if the value stored in floatVar was rounded or truncated before being stored in the variable intVar? Conduct further experiments if needed to answer this question.