User Input

https://csci-1301.github.io/about#authors

September 26, 2021 (11:12:02 PM)

Contents

1 Reading From the User **1**

2 Parsing Numeric Types **2**
2.1 Warm-Up 2
2.2 Variable Types: From String to Integer 2

3 Reading Numeric Datatypes From the User **2**

1 Reading From the User

1. Download the PersonalizedWelcomeMessage solution[1], extract and open it as usual.
2. If you are using Visual Studio on Mac or Monodevelop, you may have to perform an additional step for this program to run as expected.
   - For Visual Studio on Mac, follow the instructions at https://stackoverflow.com/a/49056993/ to have your project "Run on external console".
   - For Monodevelop, follow the instructions at https://stackoverflow.com/a/67185469/ to similarly have your projeect "Run on external console".

   You may have to perform this operation for every solution where the user is supposed to enter values.
3. Compile and execute it.
4. The user of your program (in this case, 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!
5. Read the source code carefully and make sure you understand all of it.
6. Change the code so that the program would also ask for the user's last name and print both their first and last names.

[1] PersonalizedWelcomeMessage_Solution.zip

2 Parsing Numeric Types

2.1 Warm-Up

1. So far, our user input has always returned a specific type. What type is it?
2. Without making changes to the code, execute it again but give a number as your first name. Does the type returned change if the user enters only numeric values?

2.2 Variable Types: From String to Integer

1. Create a new project.
2. Write two statements, one that declares a variable of type `int` named `intVar` and one that declares a variable of type `string` named `stringVar`.
3. Assign the value `3` to `intVar` and `"4"` to `stringVar`.
4. Display the values of `intVar` and `stringVar`.
5. Write a statement that assigns the value of `stringVar` to `intVar`. Why is the compiler complaining? Comment out the statement you just added (that is, add `//` in front of it, so that the compiler will not try to execute it).
6. Copy the following statement to "convert" the string value of `stringVar` into an integer value and assign it to `intVar`:

```
intVar = int.Parse(stringVar);
```

7. Using https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/types/how-to-convert-a-string-to-a-number, try to understand what just happened.
8. Change the value of `stringVar` to be `"Train"` and assign it to `intVar` using `int.Parse` as previously shown. What happened?

3 Reading Numeric Datatypes From the User

1. Looking back at the PersonalizedWelcomeMessage solution[2], what if you ask the user directly for an integer? How can you store it in an int variable?
2. Add the following to the code:

```
Console.WriteLine("Please enter your age in years as an integer.");

string ageInput = Console.ReadLine();
int age = int.Parse(ageInput);

Console.WriteLine($"Your age in months is {age*12}");
```

3. Re-compile and execute your code. Be sure to enter a whole number for your age.
4. Are the results what you expect?
5. Run the code again, this time with a negative number for your age. Then try again with `0`. Does the code still work?

[2] PersonalizedWelcomeMessage_Solution.zip

6. What if you were to enter a floating point number when asked for an integer? What if you entered the word “twenty”?

   Here you are purposely ignoring the prompt, but know that your user may purposely or accidentally give the wrong input type. Later in the course you will learn how to handle untrustworthy user input

7. Can you think of a change you can make to the code to accept ages of type `float` instead of `int`? Try making that change!

8. If you were to ask a user to enter an age without specifying its type, what `.Parse` should you use?