User Input

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

September 5, 2023 (05:12:43 PM)

Contents

This lab serves multiple goals:

- To understand how to let the user interact with your program,
- To understand how to read a `string` from the user,
- To understand the difference between an `int` and a `string` containing a number,
- To become able to convert a `string` containing a numerical value into a numerical datatype,
- To understand how to read a numerical value from the user.

1 Reading From the User

1. Download the PersonalizedWelcomeMessage solution[1], extract it, 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 project "Run on external console"[2].

   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:

[1] labs/UserInput/PersonalizedWelcomeMessage_Solution.zip

[2] If you are using Ubuntu and, after performing this step, you receive an error message

```
ApplicationName='/usr/lib/gnome-terminal/gnome-terminal-server', CommandLine='--app-id
↪ mono.develop.id14c27428bd5345f99daadebf684a2876', CurrentDirectory='', Native error= Cannot find the
↪ specified file
```

then follow the instructions at https://stackoverflow.com/a/65331098.

```
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. Add to the code so that the program would, in addition to asking for the user's first name, ask for the user's last name and display both their first and last names.

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?

Solution:

1. The `Console.ReadLine()` methods returns a `string`. In other words, regardless of what the user entered, it is initially treated as a `string` literal.
2. The number the user entered is still treated as a `string`, and it is important to understand the difference between an `int` (`42`) and a `string` that contains a number (`"42"`–pay attention to the quotes).

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 (i.e., 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?

Solution:

The program you end up writing looks like this:

```
int intVar;
string stringVar;

intVar = 3;
stringVar = "4";

Console.WriteLine("intVar is \t" + intVar);
Console.WriteLine("stringVar is \t" + stringVar);

// This statement returns the error message
// … error CS0029: Cannot implicitly convert type `string' to `int'
// intVar = stringVar;
// because we cannot store a string literal ("4") into an integer variable.

intVar = int.Parse(stringVar);
// This statement "converts" the string contained in stringVar
// into the number it contains. It "unwraps" the value contained in stringVar.

// Those statements return an error message that begins with
// System.FormatException: Input string was not in a correct format.
// stringVar = "Train";
// intVar = int.Parse(stringVar);
// This is because int.Parse cannot convert "Train" into an integer!
```

3 Reading Numeric Datatypes From the User

Looking back at the PersonalizedWelcomeMessage solution[3], one may wonder how we could ask the user directly for an integer. How we could store it in an `int` variable. This part explains exactly how to read nunmerical value from the user.

1. 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 at least {age*12}.");
   ```

2. Re-compile and execute your code. Be sure to enter a whole number for your age.
3. Are the results what you expect?
4. Execute the code again, but this time with a negative number for your age. Then try again with `0`. Does the code still work?
5. 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 be aware 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
6. Can you think of a change that you could make to the code to accept ages of type `float` instead of `int`? Try making that change!

[3] labs/UserInput/PersonalizedWelcomeMessage_Solution.zip

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

Solution:

We can convert the program to use `float` easily:

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

string ageInput = Console.ReadLine();
// We can parse the user input as a float:
float age = float.Parse(ageInput);

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

// We should use float by default just in case the user wants to specify the ".5" if they
 ↪  are 5 and a half years old (it matters to 5-years-old kids that they are mid-way to
 ↪  6!).
```