# Contents



# Strings (Solutions)

## Warm-up Exercises

1. Assuming that `myLastName` and `myFirstName` are two string variables that have been initialized, write a statement that concatenates them, with a space and a comma in-between, and assign the resulting string to a variable named `fullName`. For instance, if the value of `myLastName` is "Holbertonand", and the value of `myFirstName` is "Betty", then the value of `fullName` after your operation should be "Holbertonand, Betty".

   Solution

   ```
   string fullName = $"{myLastName}, {myFirstName}";
   ```

## Questions

1. What is string interpolation?

   Solution

   String interpolation is the use of specialized syntax (`$` and curly braces) to include variables within a string.

2. What is the difference, if any, between the `WriteLine` and `Write` methods?

   Solution

   The `WriteLine` method appends a newline character to the end of the argument passed into it while `Write` does not.

3. Assume we have a variable whose name is `myVariable`, whose type is `string`, and whose value is "My message". What would be displayed on the screen by the following statement?

   ```
   Console.WriteLine($"Here is my variable: {myVariable}");
   ```

   Solution

   "Here is my variable: My message"

4. Is the name myVariable the same as myvariable? If not, why?

   Solution

   The variable names myVariable and myvariable are different because variable names are case sensitive.

## Problems

1. Write down, on a piece of paper, a program that:

   (a) Declares a `string` variable named `userName`.
   (b) Displays on the screen: `Please, enter your name, followed by the enter key:`.
   (c) Reads a `string` value from the keyboard and assigns the value to the `userName` variable.
   (d) Declares an `int` variable named `number`.
   (e) Displays on the screen: `Please, enter your area code, followed by the enter ke`
   (f) Reads an `int` value from the keyboard and assigns the value to the `number` variable.
   (g) Declares a `string` variable named `id` and initializes it with the string referenced by the `userName` variable, followed by the number entered by the user. (Note: you can cocatenate a `string` and an `int` using the + sign.)
   (h) Displays on the screen: `Your id is`, followed by the content of the `id` variable.

   Here is an example of execution, where the user input is underlined, and hitting "enter" is represented by ↵:

```
Please, enter your name, followed by enter:
C l é m e n t ↵
Please, enter your area code, followed by enter:
8 2 8 ↵
Your ID is Clément828
Press any key to continue . . .
```

   Solution

```
string userName;

Console.WriteLine("Please enter your name, followed
↪  by enter.");
userName = Console.ReadLine();

int number;

Console.WriteLine("Please enter your area code,
↪  followed by enter.");
number = int.Parse(Console.ReadLine());
```

```
string id = userName + number;

Console.WriteLine("Your ID is " + id);
```

2. Write a series of statements that:

   (a) Declare a `string` variable named `favoriteColor`
   (b) Display on the screen a message asking the user his or her favorite color
   (c) Read the value entered by the user and store it in the `favoriteColor` variable.

   You can combine some of the statements if you want, but do not display on the screen any information that was not explicitly asked.

   Solution

```
string favoriteColor;

Console.Write("Please enter your favorite color: ");
favoriteColor = Console.ReadLine();
```