

Quiz #2, on Thursday, September 19, will consist of questions taken or inspired from Parts I and II of this homework and from the lab.

Part I — Questions

1. In C#, what is the “escape character”? Why is it useful?
2. Write a statement that *initializes* a variable named `myHeightInMeters` to your height in meters. What should be the datatype of `myHeightInMeters`?
3. Suppose you replace every `*` in your program with the `!` symbol. Are you completely sure that your program would still compile? Why, or why not?
4. Assume you have an `int` variable named `myAge` whose value is 24. What would be displayed at the screen by the following? `Console.WriteLine($"{myAge * 2}");`
5. Give the values of `c` and `d` after the following four instructions have been executed.

```
int c = 3, d;  
d = 2 + c;  
c = d * 2;  
d += 2;
```

6. Is there an error in the following code? Explain the error, or give the value of `b` after the second statement is executed.

```
float a = 3.7f;  
int b = (int)a;
```

7. If one of the operator's operand is of type `float`, and the other is of type `int`, what will be the type of the result of the operation?
8. What is the return type of the operation `12.4 * 3`?
9. Write an explicit conversion, from a `double` variable `myDoubleVar` to a `int` variable called `myIntVar`. You don't need to re-declare those variables. Assuming `myDoubleVar`'s value is 5.89, what value would be stored in `myIntVar`?
10. Write a statement that performs an implicit conversion between two different numeric datatypes.
11. Assuming that `myLastName` and `myFirstName` are two `string` variables that have been initialized, write a statement that *concatenate* 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"`.
12. In C#, what is the name of the method used to read from the user?

13. What is wrong with the following? Will the error appear at compilation time, or at execution time?

```
int age;
Console.WriteLine("Please enter your age:");
age = Console.ReadLine();
```

14. What, if any, is the difference between a parameter and an argument?

15. What is the purpose of the keyword `new`?

16. Do objects from the same class share their instance variable?

17. What does it mean to say that instance variables have a default initial value? How is that different from the variables we have been manipulating in the `Main` method?

18. Briefly explain what a local variable is.

19. Suppose we have a `Circle` class containing

```
public void SetRadius(double RadiusArgument)
{
    radius = RadiusArgument;
}
```

Write a statement that create a `Circle` object, and one statement that sets its radius to 3.5.

Part II – Problems

This time, the three exercises **do not** require a computer. Make sure you feel ready before starting them, try to do them with limited time and without notes, and, if you want, check your answer using VS.

Problem 1

This problem summarizes the content of Section 3.8 of your textbook, and ask you to answer various problems. There are 5 different arithmetic operations available in C#:

C# Operation	Arithmetic Operator	Algebraic Expression	C# Expression
Addition	+	$x + 7$	<code>myVar + 7</code>
Subtraction	−	$x - 7$	<code>myVar - 7</code>
Multiplication	*	$x \times 7$	<code>myVar * 7</code>
Division	/	$x/7$, $\frac{x}{7}$ or $x \div 7$	<code>myVar / 7</code>
Remainder (a.k.a. modulo)	%	$x \bmod 7$	<code>myVar % 7</code>

Computing operations involving one of them is straightforward:

Operation	Result
<code>3 + 4</code>	7
<code>3 - 4</code>	-1
<code>3 * 4</code>	12
<code>6 / 2</code>	3
<code>6 % 4</code>	2

But things get complicated when multiple operators are used, but not parenthesis are indicated. For instance, should

$$7 / 2 - 4 * 8 \% 3$$

be read as

$$\begin{aligned}(7 \div 2) - ((4 \times 8) \bmod 3) &= 3.5 - (32 \bmod 3) \\ &= 3.5 - 2 \\ &= 1.5\end{aligned}$$

or as

$$\begin{aligned}(7 \div (2 - 4)) \times (8 \bmod 3) &= (7 \div (-2)) \times 2 \\ &= (-3.5) \times 2 \\ &= -7\end{aligned}$$

? Certainly the result is not the same, and there are other possibilities!

Actually, C# uses the following rules:

1. $*$, $/$ and $\%$, called the *multiplicative operations*, are always evaluated before $+$ and $-$, called the *additive operations*. So that, for instance,

$$2 - 4 * 8$$

will be evaluated as $2 - (4 * 8) = -30$.

2. If there are multiple operations of the same type, they are evaluated from left to right. For instance,

$$4 / 2 * 8$$

will be evaluated as $(4 \div 2) \times 8 = 16$, and

$$4 - 2 + 8$$

will be evaluated as $(4 - 2) + 8 = 10$.

3. Parenthesis can be used to force a particular order of evaluation, so that $2 * (3 + 4)$ will be evaluated as $2 \times (3 + 4) = 2 \times 7 = 14$, and not as $(2 * 3) + 4 = 6 + 4 = 10$ as it would without parenthesis.

Answer the following:

1. Which one(s) of the following C# operations compute the arithmetic expression $(x \times (3 \bmod 5)) - (y \times 7)$?

- (a) $x * 3 \% 5 - y * 7$
- (b) $x * (3 \% 5) - y * 7$
- (c) $(x * 3) \% 5 - y * 7$
- (d) $x * 3 \% (5 - y * 7)$
- (e) $(x * 3 \% 5) - (y * 7)$
- (f) $(x * ((3 \% 5) - (y * 7)))$

2. State the order of evaluation of the operators in each of the following C# operations, and compute the resulting value:

- (a) $8 - 39 * 1 / 12 + 5$
- (b) $12 + -23 / 12 \% 3$
- (c) $90 * 23 / 34 - 12 - 13$
- (d) $12 \% 83 - 2 * 3$

3. (Optional) Check your answers using Visual Studio. You can use statement of the form

```
Console.WriteLine($"8 - 39 * 1 / 12 + 5 is {8 - 39 * 1 / 12 + 5}");
```

Problem 2

Write down, on a piece of paper, a program that

1. declares a **string** variable named `userName`,
2. display at the screen “Please, enter your name, followed by enter:”,
3. reads a **string** value from the keyboard and assigns the value to the `userName` variable,
4. declares a **int** variable named `number`,
5. display at the screen “Please, enter your number:”,
6. reads a **int** value from the keyboard and assigns the value to the `number` variable,
7. 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 (you can concatenate a string and an int using the `+` sign),
8. display at the screen “Your id is” and 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.
Clément ↵
Please, enter your area code, followed by enter.
828 ↵
Your id is Clément828
Press any key to continue . . .
```

Problem 3

Write down, on a piece of paper, a fully compilable program that initializes an **int** variable named `persons` with the value 5, an **int** variable named `bottles` with the value 3, and a **double** variable named `litterPerBottle` with the value 1.5. What should be the type of a variable `litterPerPerson` to be able to be assigned the number of litters every person is going to get, if we split equitably? Write the correct initialization of that variable, and a statement that display its value.

Place a delimited comment with a your name and the time at which you wrote the program at the top of the program.

