

Please read 3.6 – 3.7, 4 of the textbook and then answer the following, trying not to look at your notes or at the textbook. Quiz #2, on Thursday 1st February, will consist of questions taken or inspired from the Part I of this homework and from the lab. Project #2, presented in Part II, 2, is due Thursday 8th February, before 11:30 PM.

## Part I — Questions

1. Write a statement that *initializes* a variable named `myAge` to your age. What should be the datatype of `myAge`?
2. What is wrong with the following? Will the error appears at compilation time, or at execution time?

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

3. In C#, what is the name of the method used to read from the user?
4. In C#, what is the “escape character”? Why is it useful?
5. Suppose I replace every `*` in my program with the `!` symbol. Am I completely sure that my program will still compile? Why, or why not?
6. Assuming I have an `int` variable named `myAge` whose value is 24. What would be printed by the following?  
`Console.WriteLine($"My age is {myAge * 2}");`
7. Give the values of `a` and `b` after the following four instructions have been executed.

```
int a, b;  
a = 23;  
b = a * 2;  
a = a - 1;
```

8. What, if any, is the difference between a parameter and an argument?
9. What is the purpose of the keyword `new`?
10. Do objects from the same class share their instance variable?
11. 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?
12. Briefly explain what a local variable is.
13. Why would one want to define a constructor for a class?
14. Write a get method for a `total` instance variable of type `int`.
15. Assuming `name` is a `string` instance variable, there is problem with the following setter. Fix it.

```
public int SetName(string var){  
    name = var;  
}
```

16. Draw the UML diagram of a class named “Student” with a single attribute, “name”, of type `string`, and two methods, `SetName` and `GetName`.

## Part II — Programming Exercises

This part will help you in assessing your level of understanding of this lecture, help you to study a new topic, and give you an idea of the kind of problem you will be asked to solve during the exams. I'll assume that you will have successfully completed the first problem by the time Homework #3 is released (Thursday 1st February), so don't wait and let me know if you had difficulties solving it.

1. This problem summarizes the content of Section 3.8 of your textbook, and ask you to answer various problems. Looking at Figure 1, page 3, you'll notice that there are 5 different arithmetic operations available in C#: +, -, \*, / and %. Computing operations involving one of them is straightforward:

Operation	Result
3 + 4	7
3 - 4	-1
3 * 4	12
6 / 2	3
6 % 4	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  $(4 - 2) + 8 = 10$ .

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:

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>

Figure 1: Arithmetic Operators in C#

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

1. `x * 3 % 5 - y * 7`
2. `x * (3 % 5) - y * 7`
3. `(x * 3) % 5 - y * 7`
4. `x * 3 % (5 - y * 7)`
5. `(x * 3 % 5) - (y * 7)`
6. `(x * ((3 % 5) - (y * 7)))`

(b) State the order of evaluation of the operators in each of the following C# operations, and compute the resulting value:

1. `8 - 39 * 1 / 12 + 5`
2. `12 + -23 / 12 % 3`
3. `90 * 23 / 34 - 12 - 13`
4. `12 % 83 - 2 * 3`

(c) (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}");
```

2. In this project, you will create and use a simple class in Visual Studio. This time, you can name your project, solution and files the way you want, but remember to write your name and the date in a delimited comment at the very beginning of every .cs file in your project.

Create a Room class, with three instance variables: one to hold the name of the room, one for the length of the room, and one for the width of the room. Chose an appropriate datatype to be able to store the length and the width of the rooms (in meter) using floating point numbers. You need to create 8 methods:

- A method to set the value of each instance variable (“setters”),
- A method to get the value of each instance variable (“getters”),
- A method that returns the area of the room in square meters,
- A method that returns the area of the room in square feet.

To test your Room class, create a “test” program with a main method. In it, you should create a Room object and ask the user for its name, length and width, and then print the name of the room, followed by its area in square meters and square feet.

Once your project is completed, compile without error nor warning, and can be executed, upload the folder containing the project on Box (<https://augustauniversity.app.box.com/>, use your augusta credentials to log-in). Rename the folder to “lname\_fname”, where “lname” (resp. “fname”) is your last name (resp. first name), click on “Share”, “Invite Collaborators”, and then enter my email address (caubert@augusta.edu)

and give me the “Viewer” status (under “Invitee Permissions”). You need to invite me before **Feb 8, 2018, 11:30 PM**, and should not edit your files once you invited me.

Further details will be given in class, but make sure you follow those instructions as closely as possible.

