

Instructions: Study this homework as we progress if you want to feel ready for the first exam, on 02/17.

1 Short 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. Give the values of `a` and `b` after the following four instructions have been executed.

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

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

```
decimal a = 1.6M;  
int b = (int)a + a;
```

8. If one of the operator's operand is of type `int` and the other is of type `float`, what will be the type of the result of the operation?
9. What is the return type of the operation `12.5 * 2`?
10. Write an explicit conversion from a `double` variable `myDoubleVar` to an `int` variable called `myIntVar`. You do not need to re-declare those variables. Assuming `myDoubleVar`'s value is 5.89, what value would be stored in `myIntVar`?
11. Write a statement that performs an implicit conversion between two different numeric datatypes.
12. Assuming that `myLastName` and `myFirstName` are two variables that have been initialized, write a statement that *concatenates* them with a space and a comma in-between, and assign the resulting 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".
13. In C#, what is the name of the method used to read input from the user?
14. What is wrong with the following? Will the error(s) appear at compilation time or at execution time?

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

15. Will those statements, if placed in a proper Main method, compile? Could this program crash at execution time? Justify your answer.

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

16. Write a series of statements that: a) Declare an variable named , b) Display on the screen a message asking the user to enter his or her age, c) Read the value entered by the user and store it in the variable. You can add statement(s) performing intermediate steps if you want.

2 Problems

The following 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.

1. This problem restates differently the content the [Order of Operations](#) section of the lecture notes, and ask you to answer various problems.

There are 5 different arithmetic operations available in C#:

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

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 no 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 possible ways this calculation may be performed!

Actually, C# uses the following three rules:

- (a) $*$, $/$, 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 $2 - (4 * 8) = -30$.

- (b) 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 $(4 - 2) + 8 = 10$.

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

Answer the following:

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

- i. $x * 3 \% 5 - y * 7$
- ii. $x * (3 \% 5) - y * 7$
- iii. $(x * 3) \% 5 - y * 7$
- iv. $x * 3 \% (5 - y * 7)$
- v. $(x * 3 \% 5) - (y * 7)$
- vi. $(x * ((3 \% 5) - (y * 7)))$

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

- i. $8 - 39 * 1 / 12 + 5$
- ii. $12 + -23 / 12 \% 3$
- iii. $90 * 23 / 34 - 12 - 13$
- iv. $12 \% 83 - 2 * 3$

- (c) (Optional) Check your answers using Visual Studio. You can use a statement of the form:

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

2. Write down, on a piece of paper, a fully compilable program that initializes an variable named with the value 5, an variable named with the value 3, and a variable named with the value 1.5. What should be the type of the variable to be able to be assigned the number of liters every person is going to get, if split equitably? Write the correct initialization of that variable and a statement that displays 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.

3. 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 enter:”.
- (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 number:”.
- (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 (you can concatenate a string and an int using the $+$ sign).
- (h) Displays on 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.

Jaylah↵

Please enter your area code, followed by enter.

49391↵

Your id is Jaylah49391

Press any key to continue . . .