

Michelle M. Khalifé

Assignment 2 – datatypes and variables, statements, informative naming, notation (camelCase, Pascal)

#1 [datatypes and variable declaration]

There are many datatypes in C#. See:

<https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/built-in-types>

Ranges of Integral types:

<https://www.infoworld.com/article/2076075/core-java-learn-java-from-the-ground-up.html>

The table shows some of the language's primitive (value-based) data-types and the string reference type.

Byte (byte), 1bytes Int16 (short), 2bytes Int32 (int), 4bytes Int64 (long), 8bytes	Single (float), 4bytes Double (double), 8bytes	Char (char), 2bytes	Boolean (bool), 1byte	String, 4bytes address
For whole numbers, e.g.: -7, -5, 0, 3, 7, 11, 999	For decimal/real numbers, e.g.: -5.5, 0.00, 10.9999	For single letters, digits, special characters and/or symbols, e.g., 'a', 'b', 'c', 'X', 'Y', 'Z', '1', '2', '3', '@', '#', '!', '?' <i>Character values are surrounded by single quotes</i>	Truth values: True, False	For a sequence of characters (one or more) e.g.: "algorithms and programming", "123456789", "Joe Smith", "Jan-20" <i>String values are surrounded by double quotes</i>

Declaration Syntax

datatype varName; // declaration, reserves space in memory for that variable

or

datatype varName = some_value; // declaration and initialization, reserves space and stores the value

Convention:

Choose an informative name for your variable – typically, you would be looking at one or more nouns.

C# favors camelCase notation for naming local variables. Start with a lower-case letter, then every subsequent word starts with an upper-case letter. E.g.: myVariable, myVar1, day, weekDay, dayOfTheWeek, hourlyWage, rate, ...

Declare a variable for each of the following:

- the current temperature
- a letter grade
- your final AP average
- the number of students in a class
- your favorite colour
- the temperature required for baking a banana cake
- your middle initial
- your 7-digit student ID
- your 10-digit mobile number
- PI
- the name of your bank
- a yes/no answer to a question

To do so, you must think of any possible value that represents this variable. In the case of the temperature, one would think: -20, 0, 10, 25, but also 18.5, 7.5, ... This will allow you to determine the type. Once you've got the type and the name, you may go ahead assign values to these variables and print them.

#2 [explicit casting]

Ref.:

<https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/conversions#101-general>

Casting is when the value of a variable changes type. For example, in `int a = 5`, the 5 is an integer. If we wish to turn the 5 into 5.0, i.e., a float, then we need to prepend it with the desired type between parenthesis: `(float)5`. Seeing that the 5 is contained in variable `a`, this becomes `(float)a`. This results in a float and should be assigned to a float variable.

Casting syntax:

type1 var1 = some_value; // first, for an assigned &/or scanned value, declare a variable to contain it.

type2 var2 = (type2) var1; // then, cast this variable into another type, & save the result in a variable of that new type

a. Declare a(n) _____ variable and cast it into a(n) _____ variable. Print the latter. What do you observe in each case?

- i. integer, float
- ii. float, integer
- iii. float, double
- iv. char, integer
- v. char, float
- vi. integer, char
- vii. float, char

b. Repeat the exercise: instead of explicit casting, use the `Convert.To...()` syntax. What is the difference between explicit casting and conversion?

#3 [implicit casting]

Ref.:

<https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/conversions#1023-implicit-numeric-conversions>

"Prompt the user" translates into a `Console.WriteLine()` function call, which allows you to communicate to the user what you want them to enter, followed by a `Console.ReadLine()` function call, which allows you to read their input. Remember, the `Console.ReadLine()` returns a string.

- a. Prompt the user for a float, save it into a float var using `Convert.ToSingle()`, and then save the var in an integer. Print the integer. What do you observe?
- b. Prompt the user for an integer, save it into an integer var using `Convert.ToInt32()`, and then save the var into a float. Print the float. What do you observe?
- c. Prompt the user for an integer or a float. This time, save it in the right variable type. Print the variable. Anything?

#4 [logic]

A statement is either true or false, but it cannot be both at the same time. For instance:

Statement	Truth Value
S1: 1 is strictly less than 5	True
S2: 0 is equal to 100	False
S3: "January is considered a winter month"	True
S4: "We meet for algorithms on Fridays"	False
S5: "2 is even and prime"	True

The table on the left depicts the resulting truth value when two statements are joined with logic operators:

- && (and), and
- || (or)

The table on the right depicts the truth value of a statement (or a logic operator) that is being negated.

S1	S2	&&	
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

S	!S (not S)
T	F
F	T
&&	
	&&

- a. Come up with 4 or 8 pairs of statements that would allow you to understand how && and || operate. For example, the pair below illustrates the first row in the table:

S1: 2 is even

S2: 2 is prime

S1 && S2 => true and S1 || S2 => true

- b. Throw a third statement S3 in the mix. Let S3 be true, then let it be false. What happens?

#5 [function naming]

main() is a function. It is called *main* because it is the main point of entry & exit of the program. Other functions that programmers create will be defined outside *main()*. Unlike variables, functions do not carry information: they DO something. They perform an action. Therefore, they are best named with a verb followed by a noun. C# favors PascalCase notation for function names. The first letter of every word is capitalized while the rest of the word is in lower-case. E.g.: *MakeBananaMangoSmoothie*, *MakeCappuccino*, *IsSunny*, *RunAnalytics*, ...

How would you simply name a function that:

- adds a bunch of numbers
- computes the value of a number raised to some power
- tells you the day of the week
- prints a new line
- lets you know if a number is prime or not
- converts a numeric average to the corresponding letter grade
- calculates your BMI