Naming Rules and Conventions

C, C++, C#, Java,

Dan McElroy

Hello programmers! Dan McElroy here with a discussion about rules and conventions for naming things inside a program. This discussion does not cover file names. When writing code it is important to give meaningful names to variables, constants, subroutines, classes and objects. A rule is a requirement of the programming language that will cause a syntax error if violated. A convention is not a violation of the programming language but is like an agreement among programmers on how things should be done. As you write a lot of code, you may find that some of the rules might be a little different between different programming languages. You also may find that some of the conventions may be a little different even within the same language depending on where you work.

Rules

You can use the following characters when creating a name for something inside a program:

A to Z, a to z, 0 to 9, and _ (underscore)

Some languages like C, C++ and Java permit the use of the dollar sign $ as part of a name.

You can use the following characters when creating a name for something inside a program:
big A to big Z, small a to small z, the digits 0 to 9, and the _ (underscore) character.
Some languages like C, C++ and Java also permit the use of the dollar sign $ as part of a name.

More Rules

IMPORTANT: Names can not start with a digit (0-9). Starting with the underscore _ is discouraged.

ALSO IMPORTANT: You can't use a keyword, also known as a reserved word that belongs to the language.

Here are some more rules
Names cannot start with a digit (0-9). Starting with the underscore _ is discouraged.
ALSO IMPORTANT: You can't use a keyword, also known as a reserved word that belongs to the language.

Legal and Illegal Names

Legal	Illegal	Why?
firstName	1stName	can't start with a digit
last_name	first name	space is illegal
	name#1	non-valid character #
	while	reserved word

Here are some more rules
Names cannot start with a digit (0-9) and starting with the underscore _ is discouraged.
ALSO IMPORTANT: You can't use a keyword, also known as a reserved word that belongs to the language.

Multiple human language words can be used to form a name but spaces can't be used. Here are some examples of legal names:
firstName, last_name.
firstName combines two English words together to form one variable name. It is easy to read because the word Name has its first letter capitalized. second_name is also easy to read because instead of using a space as a separator between the English words, the underscore _ character is used.
1stName, starting with a numeric 1 is illegal because the first character is a numeric digit
first name is illegal because a space character is used to separate the two English words. A programming language would look for a variable named first and then would not know what to do when it found the word name.
name#1 is illegal because the # pound-sign or hash mark is not a legal character for variable names
while is not a legal name because the word while is one of the reserved words that belong to the language itself. Although not all programming languages use the exact same set of reserved words, some of the most common are control words such as if, else, do, while, for, end and data types such as int, Integer, double, char, string. The editors in most Integrated Development Environment systems (IDE) give a special color to reserved words. You will learn the meaning of the reserved words as you study each programming language.

Case Sensitivity

Many languages like C, C++, Java, C#, and Python are case sensitive. Other languages like Visual Basic are not. When we say a language IS case sensitive, that means the capital letters and small letters are treated as though they have no relation to each other. It is like there are 52 different letters. Although discouraged, in C++, you could have three completely different variables named **counter, Counter, COUNTER** – but in Visual Basic, all three names would refer to the exact same variable.

Many languages like C, C++, Java, C#, and Python are case sensitive. Other languages like Visual Basic are not. When we say a language IS case sensitive, that means the capital letters and small letters are treated as though they have no relation to each other. It is like there are 52 different letters. For example, in C++, you could have three completely different variables named

counter, Counter, COUNTER – but in Visual Basic, all three names would refer to the exact same variable.

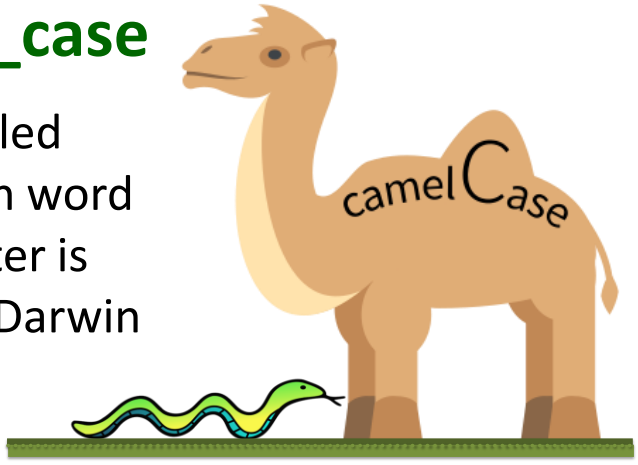
Naming Conventions

Names can be as long as needed to make them meaningful. And abbreviations can also be used to shorten names. Just make sure they are meaningful to other people.

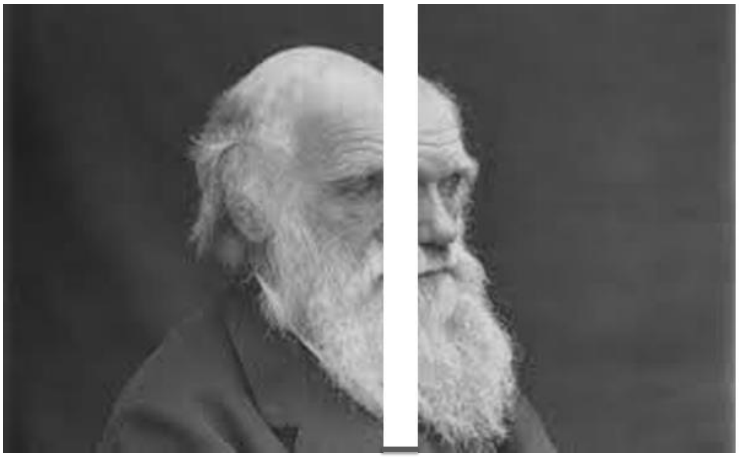
Names can be as long as needed to make them meaningful. And abbreviations can also be used to shorten names. Just make sure they are meaningful to other people.

camelCase vs. snake_case

Capitalizing each word is called ‘camelCase’. Separating each word with the underscore character is called either snake_case or Darwin case.



When combining multiple human words to form a name in a program, you can either capitalize the first letter of each word in the middle, or separate each word using the underscore _ character. Capitalizing each word is called ‘camelCase’. Separating each word with the underscore character is called either snake_case or Darwin case.



Dar_win

But Darwin?

Variable and Subroutine Names

By convention, start the names of variables, subroutines, functions and objects with a lower case letter:

```
regularHours
overtimeHours
pay = computePay(regularHours, overtimeHours);
```

Function call

By convention, start the names of variables, subroutines, functions and objects with a lower case letter:

```
regularHours
overtimeHours
pay = computePay(regularHours, overtimeHours);
```

computePay is the name of a function, also called a method in object oriented programming. The computePay function is being passed two parameters, regularHours and overtimeHours. The function should be written to accept these two parameters, compute and return the value for the paycheck so that it can be stored in the variable named pay using the = assignment operator. More on that later.

‘Class’ Names

Object Oriented Programming is based on the concept of **objects**, which can contain data and executable code. The structure and form of an object is defined by a piece of code called a **class**. An analogy between a class definition and objects built from the class is similar to a blueprint (class) and buildings (objects). Names of classes should start with a capital letter while names of objects should start with a small letter.

Object Oriented Programming is based on the concept of **objects**, which can contain data and executable code. The structure and form of an object is defined by a piece of code called a **class**. Names of classes should start with a capital letter while names of objects should start with a small letter.

Names of Constants

Variables and constants are similar except that once defined, the value of a constant can not be changed by the program when it is running. By convention, constants are defined using only capital letters and the underscore character.

C	#define TAX_RATE 0.085 // 8.5%
C++	const double TAX_RATE = 0.085; // 8.5%
Java	final double TAX_RATE = 0.085; // 8.5%

Variables and constants are similar except that once defined, the value of a constant can not be changed by the program when it is running. By convention, constants are defined using only capital letters and the underscore character.

Names of Constants

Variables and constants are similar except that once a constant can not be changed by running. By convention, constants are defined using only capital letters and the underscore character.

Do not use an equal sign = or a semicolon ; when defining a constant in the C language

C	#define TAX_RATE 0.085 // 8.5%
C++	const double TAX_RATE = 0.085; // 8.5%
Java	final double TAX_RATE = 0.085; // 8.5%

Here are examples of defining constants in C, C++ and Java. As you can see, the syntax is different for each language. The C-language uses the #define statement and does not use the equal-sign = or a semicolon at the end of the statement. C++ starts a constant definition with the word **const** and uses the assignment operator = and a semicolon at the end. Java is similar to C++ but uses the keyword **final** instead of **const**. C++ and Java are also similar in that a data type is associated with the constant, in this case **double** but constants can be defined with any data type. A data type is not associated with the constant definition in C. Both C++ and Java are more advanced than C on the 'evolutionary' scale for computer languages in that data types are part of the definition for a constant.

Hungarian Notation

Hungarian notation starts a variable or object's name with a group of lower case letters that identify the data type. For example, an integer may start with the letters int (intCounter), or a double with the letters dbl (dblPayRate).

Hungarian notation starts a variable or object's name with a group of lower case letters that identify the data type. For example, an integer may start with the letters int (intCounter), or a double with the letters dbl (dblPayRate).

Hungarian Notation

RECOMMENDATION: follow the naming convention already established if working on an existing project.

Wikipedia has an excellent article

http://en.wikipedia.org/wiki/Hungarian_notation

RECOMMENDATION: follow the naming convention already established if working on an existing project.
Wikipedia has an excellent article
http://en.wikipedia.org/wiki/Hungarian_notation

This is the end of the discussion on creating names for things that are used as part of a program. And most important of all is to make names meaningful, both to you and any future person who may read your code. One time I worked with a programmer who named the variables – junk1, junk2, junk3, junk4, junk5, etc. The programmer was temporarily assigned to help out on a different project for a couple of months and when returning to the original project had no idea what each of the junk variables was used for. Don't let that happen to you. It may be you who needs to look at something you yourself have written not too long ago. It might even be a lab project you want to refer to because you need to create something similar. Using good names can help make programming more meaningful and more fun for you both now and in the future.
-Bye for now. See you on the next video.

Enjoy working with classes and objects in Java.
Bye. See you later. Dandalf signing off for now.