ANSI C

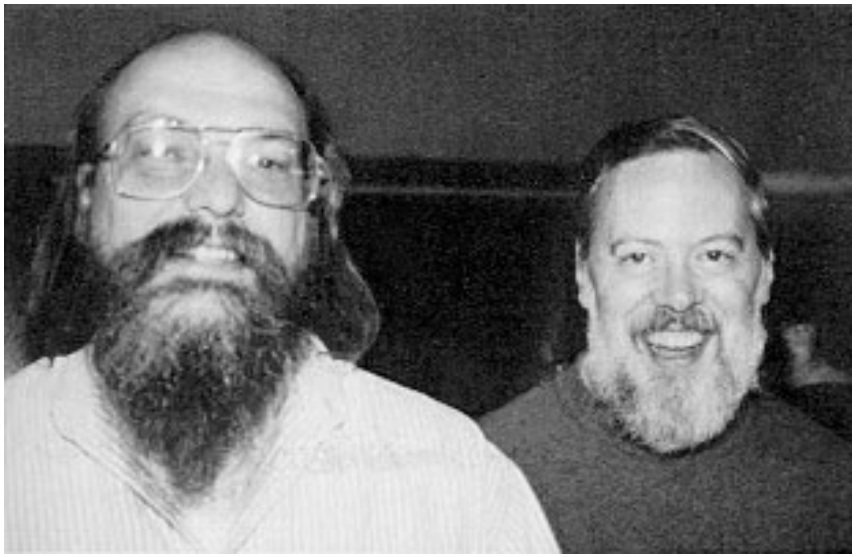
Data Analysis in Geophysics

Demián D. Gómez

November 2013

ANSI C

- Standards published by the American National Standards Institute (1983-1989).
- Initially developed by Dennis Ritchie between 1969 and 1973



Dennis Ritchie

http://en.wikipedia.org/wiki/C_(programming_language)

Why is it a good idea to write your program in C?

- It gives you the ability to program in a very solid and portable language.
- It is very flexible and fast.
- If you want to create a very efficient application, it is the way to go.
- Lots of ready to use libraries.
- If you program in C you can program almost in any computer language.

Why is it NOT a good idea to write your program in C?

- It can be tedious if you don't know exactly what you are doing.
- If your are writing a small App that you are going to run just once, then is probably not worth it to spend time trying to put together a C++ program. Just use Matlab and you'll be fine…
- But, when you need speed and efficiency, maybe you should think about writing your code in C.

How is C and C++ compared to other languages? (I)

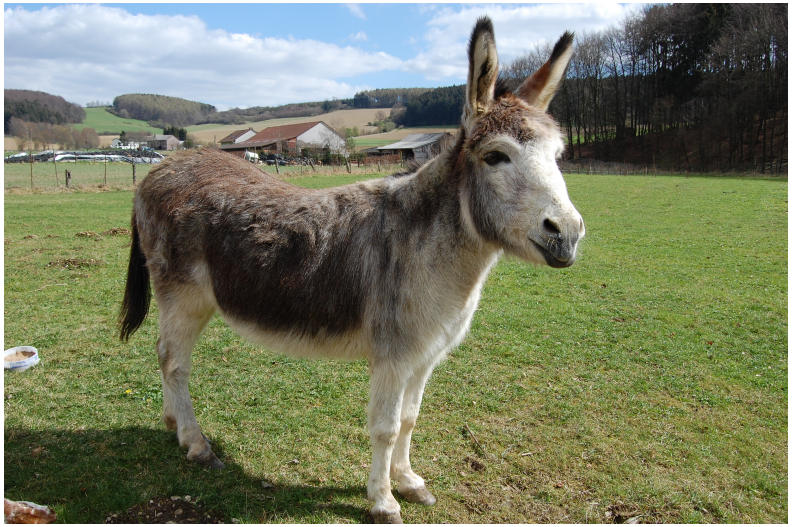
- In terms of preconfigured features:

Matlab, Visual Basic, Java

ANSI C, C++

How is C and C++ compared to other languages? (II)

- In terms of speed and flexibility



Matlab, Visual Basic

ANSI C, C++

Are we going to learn C today?

- In today's class, we will try to give you an insight into how to use and interpret a C++ program, but we don't expect you stop using Matlab.

- To really learn how to program in C takes more than just one class, but maybe you would like to get started… so, here we go…

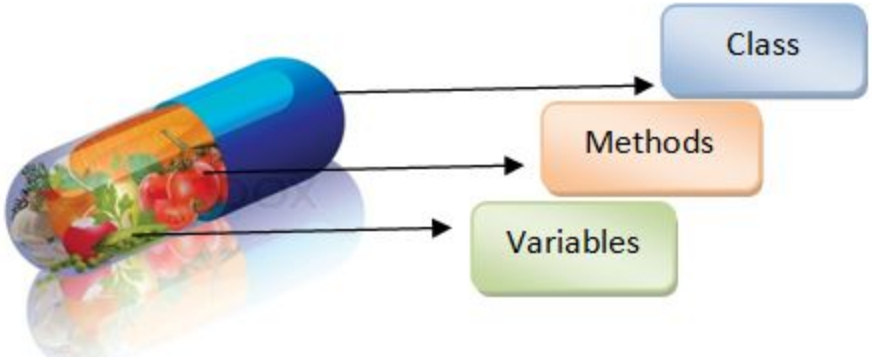
What is the difference between C and C++?

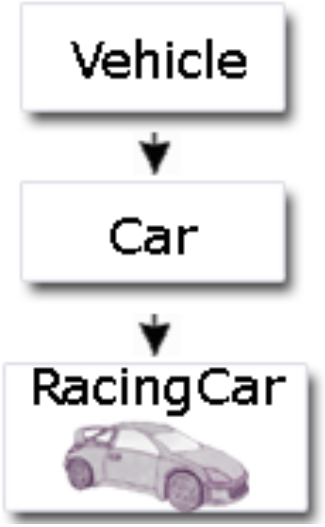
- The language structure is the same, but C++ introduces the Object Oriented Programming (OOP) concept.
- OOP introduces the following:
  - Objects (classes) that have associated properties and methods.
  - It is possible to create as many “instances” of an object as there is memory in a computer. This means, that you can create “copies” of your objects an let them interact with other objects, procedures, functions, etc.

OOP in 2 minutes…

- Encapsulation: restriction to access certain procedures and properties.
- Inheritance: one object can have other objects inside and can “inherit” their properties without having to write more code.
- Polymorphism: create procedures for objects whose exact type is not known until runtime. A very usefull extension of polymorphism is overloading.

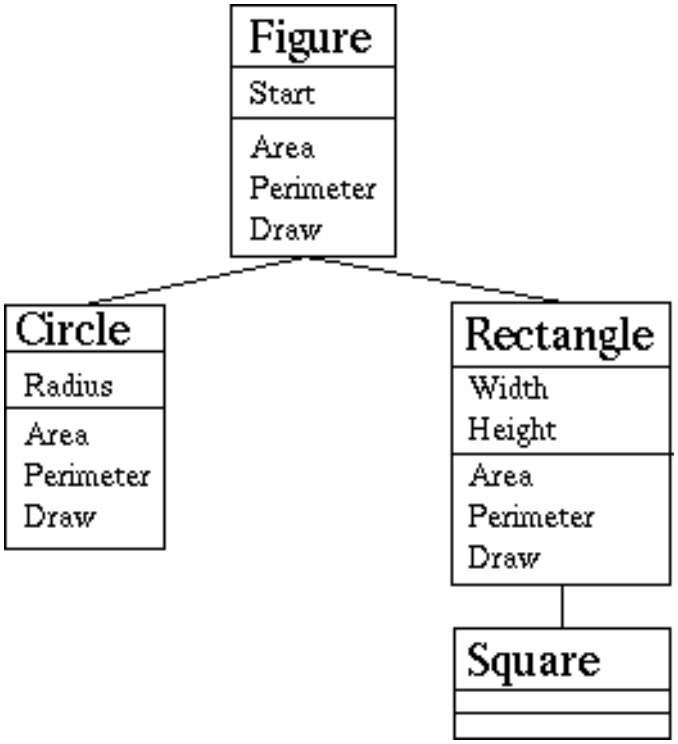
Encapsulation & Inheritance





http://www.c-sharpcorner.com/UploadFile/e881fb/learn-object-oriented-programming-using-C-Sharp-part-5/

Polymorphism



Function Overloading

- Some languages (like C and C++) allows us to overload a function or operator depending on what arguments we pass.

```
void eat(Apple apple1)
{
    bite(apple1);          // take a bite of the apple
    chew();                // chew
    swallow();             // then swallow.
}

void eat(Banana banana1)
{
    banana1.peel();        // peel the banana first
    bite(banana1);         // take a bite of the banana
    chew();                // chew
    swallow();             // then swallow.
}
```

A simple C program

```
#include <stdio.h>

int main()
{
        int i=0;
        for(i=1; i<5; i++)
                printf("Hello World\n");

        return 0;
}
```

- To compile, just type: gcc main.c –o main.o

A simple C++ program

```
#include <iostream>

using namespace std;

int main()
{
        int i=0;
        for(i=1; i<5; i++)
                cout << "Hello World" << endl;

        return 0;
}
```

- To compile, just type: c++ main.c –o main.o

Libraries and Includes

- There are lots of free libraries to do EVERYTHING in C and C++.
- These include: matrix multiplication, FFT, image processing, audio handling, etc.
- To include a library can be non-trivial, but if everything works OK, it should be as simple as typing:

```
#include <library>
```

There is a library for everything

- If you want to do math operations (like, take a sine, cosine or square root) you will need to include math.h
- The language does minimal operations (like + - *) but the compiler doesn't know how to interpret sin(x) unless you include the right library.

C, C++ operators

- i++; means i = i + 1;
- i--; means i = i - 1;
- i += j; means i = i + j;
- i *= j; means i = i * j;

- i = pow(j,2); means i = j^2. It requires math.h
- All statements are terminated with a ;

Overloaded Operators

- In C, you can overload almost any operator.
- For example, << means left shift. But, including iostream it can mean “output”.
- Another example: A*B can be a normal multiplication or a matrix multiplication or a scalar time matrix multiplication depending on what A and B are.

C, C++ Decision Blocks

```
int i=0;

[other code in here…]

if (i == 2)
{
        cout << “i =” << i << endl;
}else{
        cout << “i is not 2” << endl;
}
```

C, C++ For Loops

```
int i=0;
double result=0;

[other code in here…]

for(i=1; i<5; i++)
{
        [Other operations in here…]
        cout << “Result ” << result << endl;
}
```

Doing Something Useful

- Copy to your work dir the directory named armadillo from gaia/ddgomez/public/
- Create a new file named main.cpp in the same directory and open it in **xcode**
- Type the following…

Using Armadillo to Multiply Matrices

```
#include <iostream>
#include "armadillo/include/armadillo"

using namespace std;
using namespace arma;

int main()
{
        mat A=randu<mat>(3,3);
        mat B=eye(3,3);
        cout << "Matrix A:" << endl << A << endl;
        cout << "Matrix B:" << endl << B << endl;
        cout << "Multiplication" << endl << A*B << endl;
        return 0;
}
```

- To compile use: c++ main.c -o main.o
- To use Armadillo visit: http://arma.sourceforge.net/

Solving a Linear System of Equations

- Create a matrix called A like this

  mat A="1 2 4; 2 0 1; 1 2 -2";

- Create a matrix L like this

  mat L="7; 3; 1";

- Use the inv() function to solve the system Ax = L and display the output (x).

- To compile this program, use:

  c++ main.cpp -o main -llapack

Your code should look like this...

```
#include <iostream>
#include "../armadillo/include/armadillo"

using namespace std;
using namespace arma;

int main()
{
        mat C="7; 3; 1";
        mat D="1 2 4; 2 0 1; 1 2 -2";
        .......
        .......
```

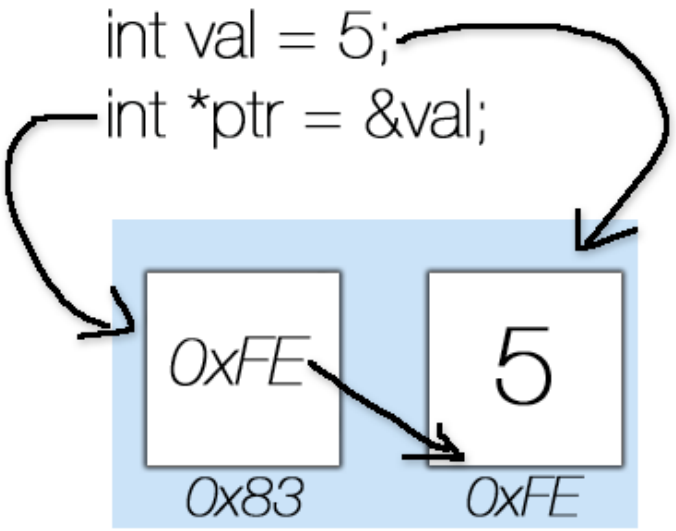
Advanced Stuff (I)

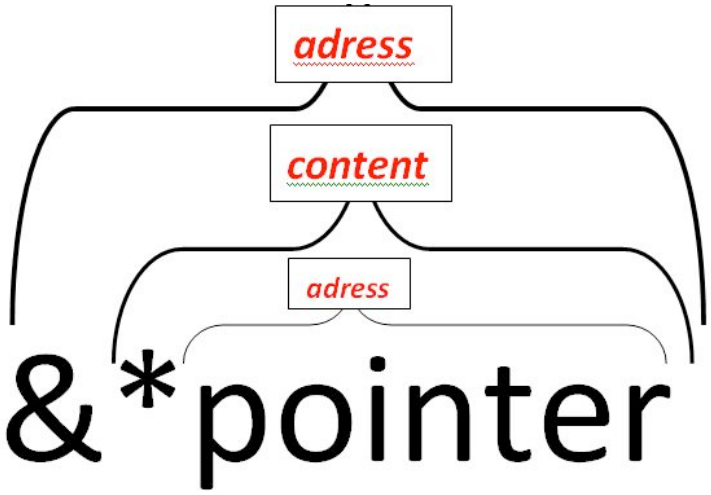
- Pointers: indirect addressing technique to read and write data from the computer memory. They are declared with an *

```
int *A=0;
double *vector=0;
```

Advanced Stuff (II)

- Address of operator:





Pointer to ***char*** Example

```
#include <iostream>
using namespace std;

int main()
{
	const char *str="Hi there! This is a pointer test.";

	for (int i=0; i<strlen(str); i++)
	{
		cout << str[i];
	}
	cout << endl;

	cout << "This string was stored in memory address: " << &str << endl;

	return 0;
}
```

C++ Reference Website

- Reference guide
  - http://www.cplusplus.com/reference/
- Tutorial:
  - http://www.cplusplus.com/doc/tutorial/