# Cpluspan – Start C++ with one code

By Alex Grompone

## Introduction

In this work I propose a simple, innovative and effective method to learn a new programming language in just few steps using **one comprehensive code**. The Cpluspan project is a free and open-source C++ algorithm. It is a fully commented educational tool that allows the student to see, test and experiment all the main functionalities of C++ within a single algorithm.

The idea derived by my constant need to learn, refresh or teach different programming languages in few days. As most of my colleagues might agree, the best way to learn a programming language it is by using it and reverse engineer existing codes, just like to learn a foreign language it is better to go in the foreign country instead of just relying on grammar book.

I found it very useful for myself and for my students to develop a single code meant to be a benchmark of information and examples. This code is useful to climb the steep learning curves of software programming without relying solely on memory and time. Afterwards it becomes easier for the user to fill the gaps with other books and tutorials, without getting lost in the jungle of conventions and fine tuned coding.

**This method is a fast and painless way to get the basic tools required to start reading and writing code in C++ using the fundamentals of Object Oriented Programming.**

This is a fast method to get out there and start coding using all the required functionalities of C++, but for deeper understanding and knowledge of the programming language it is important to follow up with some class or text book. A list of good references is provided at the end of this document.

This document provides a step by step tutorial on how to setup your Windows or Linux machine to start programming and a description of the entire code, of the topics and of the examples contained in it.

## C++ Installation Tutorial

### Setup a C++ editor and compiler in Windows 7 64bit

A valuable option to have a good and free C++ programming environment is to combine the editor Eclipse with the compiler MinGW.

1. Verify the installation of, or download and install, **Java JRE** and the updates of Windows
2. Download and install **MinGW**, choosing “C++ Compiler” and “MSYS Basic”
3. Download and install **Eclipse for C++ Developers**
4. In Eclipse, create a **new C++ Project** (from menu File->New) and select **MinGW as tool chain** (for existing projects this can be also done from the Project Explorer, right clicking on the Project Name and choosing Properties-> C/C++Build-> Tool Chain Editor)
5. From the Project Explorer, right click on the Project Name and go to Properties-> C/C++Build-> Settings-> Tool Settings-> MinGW C++ Linker-> Miscellaneous.
6. Add “-static-libgcc -static-libstdc++” as Linker flags and Apply.
7. If the error “std could not be solved” shows in your program, from Project Explorer right click on the Project Name-> Properties -> C/C++ General -> Paths and Symbols -> Includes -> Languages and select GNU C++. Modify the code and save it.

### Setup a C++ editor and compiler in Ubuntu 14.04

The editor Eclipse is a valid option also for a Linux environment using gpp. Here I provide a step by step installation procedure valid for Ubuntu 14.04 .

1. Update the repositories

>> sudo apt-get update

1. Install Java

>>sudo apt-get install openjdk-7-jdk

1. Download Eclipse from Eclipse website http://www.eclipse.org/downloads/?osType=linux
2. Move the downloaded files to the /opt directory

>>sudo mv Downloads/eclipse-sandard-kepler-SR2-linux-gtk-x86\_64.tar.gz /opt/

1. Unzip the file

>>cd /opt

>>sudo tar –xvf eclipse-standard-kepler-SR2-linux-gtk-x86\_64.tar.gz

1. Create a new file **eclipse.desktop** in **/usr/share/applications/** and add the below code.

[Desktop Entry]

Name=Eclipse

Type=Application

Exec=/opt/eclipse/eclipse

Terminal=false

Icon=/opt/eclipse/icon.xpm

Comment=Integrated Development Environment

NoDisplay=false

Categories=Development;IDE;

Name[en]=eclipse.desktop

1. Now install that desktop file using the below command.

>>sudo desktop-file-install /usr/share/applicatons/eclipse.desktop

1. Create a link file

>>sudo ln –s /opt/eclipse/eclipse /usr/local/bin/

## The Algorithm Structure

The algorithm is composed by a main source file “Cpluspan.cpp”, two cpp function files and a header.

The main file organizes the program in topics through a ‘switch case’ menu that allows the user to select the Topic and recognize within the code the different sections of the algorithm. The user can test a topic simply by selecting the number of the test he wants to perform. Every topic is a collection of basic example lines written in a way to be easily understood, tested and modified. The topics are organized in a propaedeutical way, so that every command used for the first time is commented on the same line or explained in the Topic documentation of this chapter

The algorithm has been structured as represented in the following scheme:

Main Script

Introduction Output

Switch Case (for Topic Selection)

…

Topic 1

Topic 2

Topic 3

Include Headers and Libraries

Define Functions

Define Classes

Algorithm Information and Comments

The Topics of the algorithm are the following:

1. Standard Headers, Libraries and Comments
2. Strings, Outputs and Inputs
3. Mathematical operations and Conditions
4. Defined Functions and Headers
5. Conditional Structures and Logical Operations
6. Variables and Standard Data Types
7. Arrays
8. Pointers
9. Classes
10. Object Oriented Programming
11. Memory Management (under construction)
12. Templates (under construction)

## Strings, Outputs and Inputs

The first topic of the algorithm shows some basic strings operations, how to define strings, and how to input and output information from the code.

## Mathematical operations and Conditions

Topic 2 is a basic example on how to use the simple operations, input values and output results.

## Defined Functions and Headers

Topic 3 shows how to define a function above the main() string or define and call a function through and header or another source.

## Conditional Structures and Logical Operations

Topic 4 provides basic examples of the Conditionals trough the main logical operations **if**, **switch**, **while** and **for**.

## Variables and Standard Data Types

In this topic the main Types are listed and a if statement is used to collect the input value in the correct data type. Further work in this topic will show the effects of data type conversion (casting) and effect on operations between different types.

## Arrays

This topic is an example on how to define, initialize and use an array.

## Pointers

Topic 7 provides basic examples to show what pointers are, how to declare and dereference a pointer, and how to use them as arguments.

This topic also shows the relationship between pointers and arrays.

## Classes

The basic example of a classes has been defined in the algorithm to describe a class of mammals like in the figure below:

To have a better idea on how it works, the same schematic with values is provided in the following scheme:

## Object Oriented Programming

For the Object Oriented Programming Topic, some of the main techniques of handling classes are discussed. The example of inheritance from the “Mammals” Class to “Humans” is represented in a schematic form below:

And with example values we have:

Another technique discussed in this topic is the Constructor, to provide “function like” properties to the class defining in this case an input and output, in schematic form can be visualized as below:

## Under Construction Topics

Future work is required to implement some basic information regarding Memory Management and Templates. For the time being I suggest to take advantage of the tutorials and references available for free on the internet. Some good references and links are provided below.

## References

MIT OpenCourseWare <http://ocw.mit.edu/index.htm>

How to install eclipse in ubuntu 14.04 <http://www.krizna.com/ubuntu/install-eclipse-ubuntu-14-04/>

C++ Programming, <https://en.wikibooks.org/wiki/C%2B%2B_Programming>