Introduction to this Course

Shahram Rahatlou

Computing Methods in Physics

`http://www.roma1.infn.it/people/rahatlou/cmp/`

Anno Accademico 2019/20

SAPIENZA
UNIVERSITÀ DI ROMA

Why C++?

- I am a particle physicist
- I am not a
  - programmer
  - C++ guru
  - Python expert
- Why this course?
  - In past 20 years C++ has become the primary programming language in High Energy Physics
    - Large experiments adopted C++ in 1990s moving away from Fortran
    - Many data analysis tools and numerical libraries rewritten in C++
      - Detector simulation
      - Signal extraction from detector and data acquisition
      - Data analysis
      - Multidimensional fits and multivariate techniques to determine parameters of the Standard Model from large number of experimental results
    - Since early 2000 Neural networks and machine-learning techniques have been used in High Energy physics
      - First easy implementation of these tools in ROOT with C++
    - Machine Learning has played a key role in discovery of the Higgs boson

Why Python?

- In past 10 years an explosion of Machine Learning (ML) and Artificial Intelligence (AI) also in commercial areas thanks to smartphones
  - Tagging photos
  - Shazam
  - Playlists on Spotify and Netflix
  - Voice recognition and digital assistants
- At the same time ML heavily used for data analysis at CERN Large Hadron Collider (LHC) in many areas
  - detector response
  - signal background discrimination
  - physics data analysis such as Higgs discovery
- Python is a very popular and versatile language for analysis of Big Data in both industry and physics
  - Many many freely available tools and libraries, tutorials, courses, HOW-TOs

About this course

- working assumption is that
  - you are familiar with the content of courses in Laboratorio di Calcolo and Fisica Nucleare e Subnucleare (Triennale)
  - You know why we write programs and applications to do Physics
  - You know what compilation means and how to use libraries

- Objectives of this course
  - Understand the advantage of object oriented programming (OOP) and using C++/Python for implementation
  - Being able to write simple applications in C++/Python to solve physics problems, perform data analysis, do numerical simulation
  - Learn to use external libraries and toolkit in your applications
    - For example ROOT in C++
  - Understand the differences between python and C++ and their pros and cons
  - Learn to use popular Machine Learning tools like scikit-learn

C++ in Physics

ROOT
Data Analysis Framework

Google Custom Search

Download Documentation News Support About Development Contribute

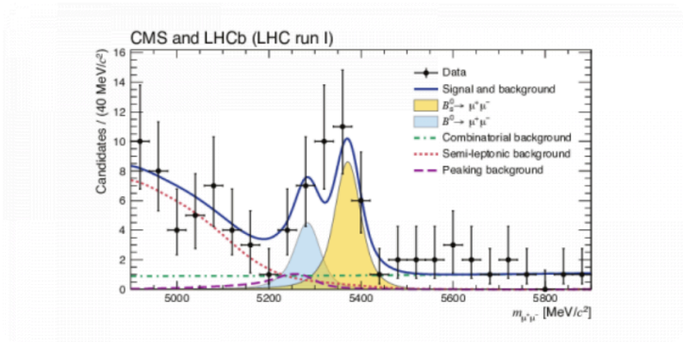
Getting Started | Reference Guide | Forum | Gallery

ROOT is ...

A modular scientific software toolkit. It provides all the functionalities needed to deal with big data processing, statistical analysis, visualisation and storage. It is mainly written in C++ but integrated with other languages such as Python and R.

Start from examples or try it in your browser!

Download or Read More ...



Previous Pause Next

Collaborator Login

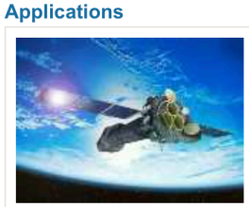
GEANT4
A SIMULATION TOOLKIT

Download | User Forum
Contact Us | Gallery

Overview

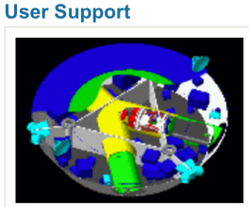
Geant4 is a toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science. The three main reference papers for Geant4 are published in Nuclear Instruments and Methods in Physics Research A 506 (2003) 250-303, IEEE Transactions on Nuclear Science 53 No. 1 (2006) 270-278 and Nuclear Instruments and Methods in Physics Research A 835 (2016) 186-225.

Applications



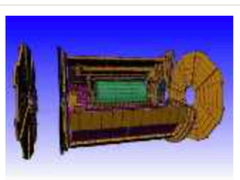
A sampling of applications, technology transfer and other uses of Geant4

User Support



Getting started, guides and information for users and developers

Publications



Validation of Geant4, results from experiments and publications

Collaboration

Who we are: collaborating institutions, members, organization and legal information

printer-friendly version

News

- *29 Jun 2018*
  Release 10.5-BETA is available from the BETA Download area.
- *25 May 2018*
  Patch-02 to **release 10.4** is available from the Download area.
- *12 Mar 2018*
  2018 planned developments
- *20 Oct 2017*
  Patch-03 to release 10.3 is available from the source archive area.

About lectures

- I will not go into details of all possible operators, functions, commands, and syntax of C++ and Python
  - excellent online resources and web pages with examples and tuorials
  - lectures will focus on use case and examples, and application is problems to solve

- Will discuss aspects of OOP with C++ that make it better and more versatile than C

- Many specific examples to exercise with OOP in C++

- Feel free to try free online tutorial for more technical aspects of the language

Learning a Language

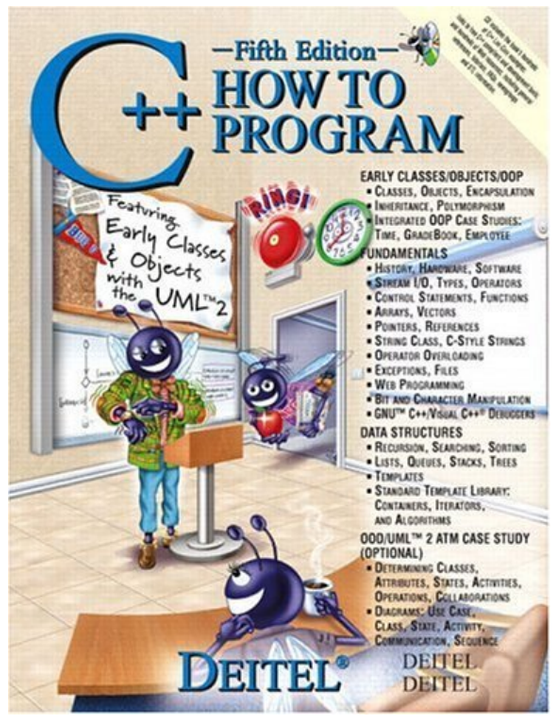
- Learning a programming language similar to learning a new spoken language
  - Learn by trying and making mistakes
  - Theory is useless if
    - your code does not compile
    - compiles but does not run
    - runs but you do not understand its output
- Writing trivial and simple applications is vital to understand features of OOP and C++/Python
  - You learn to understand and fix compilations errors
    - Extremely important when you will have to use application with $O(10^6)$ of cose and not written by you!
  - By repeating the same steps and basic operations in C++ trivial errors will go away and you will focus more on advanced features of the language

Course organisation

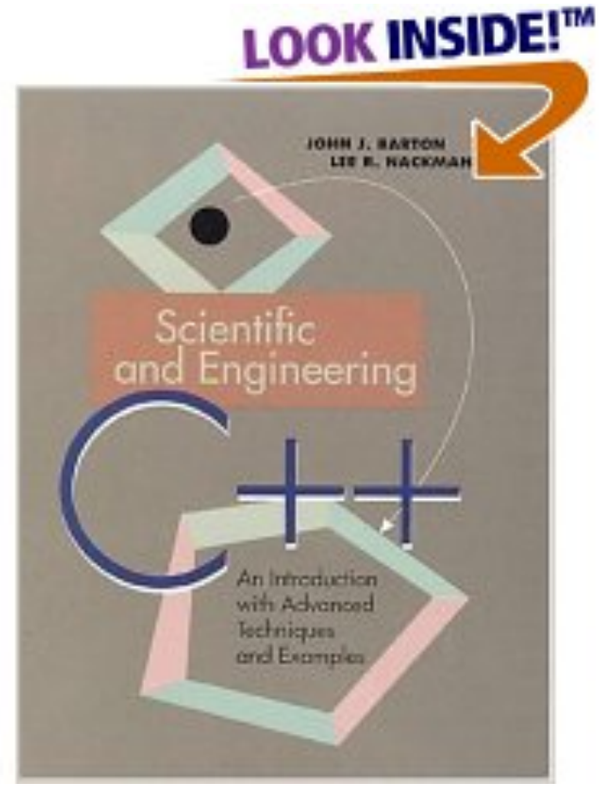
- In classe lectures on Tuesday and Friday
- Lectures in PDF format with many examples discussed
  - Material presented and files of all examples available on the course website after the lecture
- Exercise is necessary to learn
- This year introducing assisted lab sessions of 2 hours on Monday
  - suggested by students last year
  - you work in groups of two on problems or examples discussed in class
- Exams
  - Dates already available on infostud
  - a PDF file with 2 problems (usually 1 in C++ and 1 in python) available on course web site the day of the exam
  - You are given 48 hours to submit your project online
  - Oral discussion of the code and your choices to determine the grade
  - Grades range from 18 to 30 depending on number of mistakes
  - Depending on number of students attending the course we might have a midterm exam

Simple reference books

- Deitel & Deitel, **`C++ How To Program 5th Edition`**, editore Pearson - Prentice Hall
  - Very good for beginners with many examples and step by step instructions. Useful tips for both basic and advanced users
  - Italian version probably available in the library
  - Recently released 6th edition

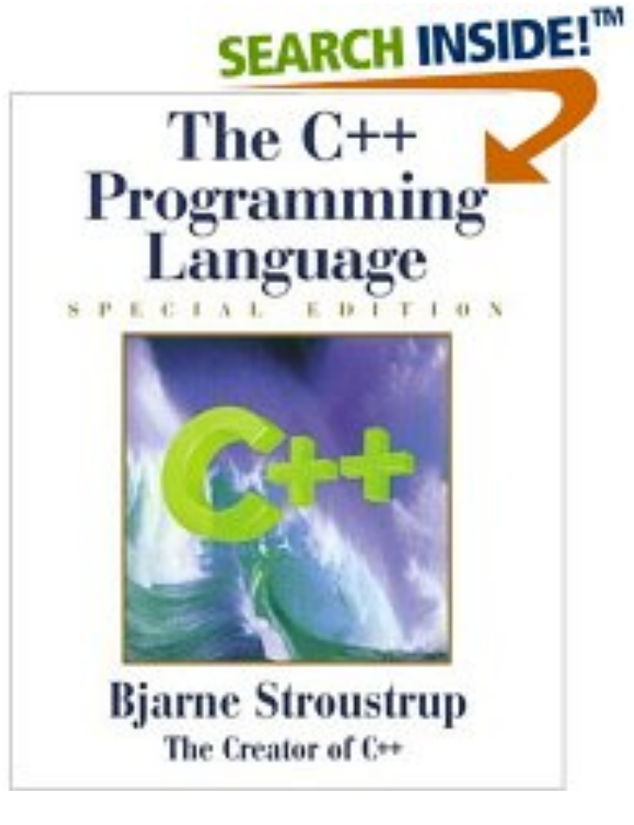


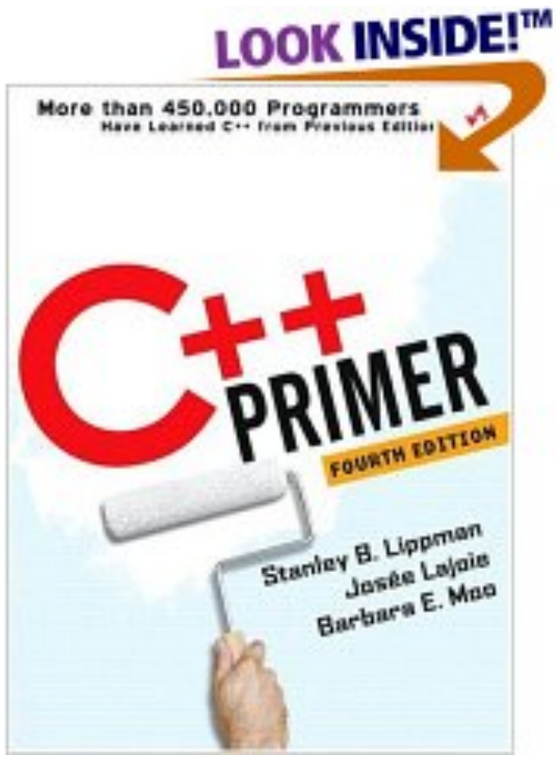
- J. Barton & L. Nackman, `Scientific and Engineering C++`, editore Addison-Wesley
  - Simple and a bit minimal
  - Covers all important features but very little text and difficult to use as a text book



More advances textbooks

- B. Stroustrup, C++ Programming Language
  - Bible of C++ written by the original author of C++
  - Very dense and very complete covering everything!
  - Not a teaching text book but a very good reference for detailed aspects of the language
  - Similar to Landau & Lifshits for Mechanics!
- Lippman, C++ Primer
  - Another very complete reference book lightly easier than the previous one

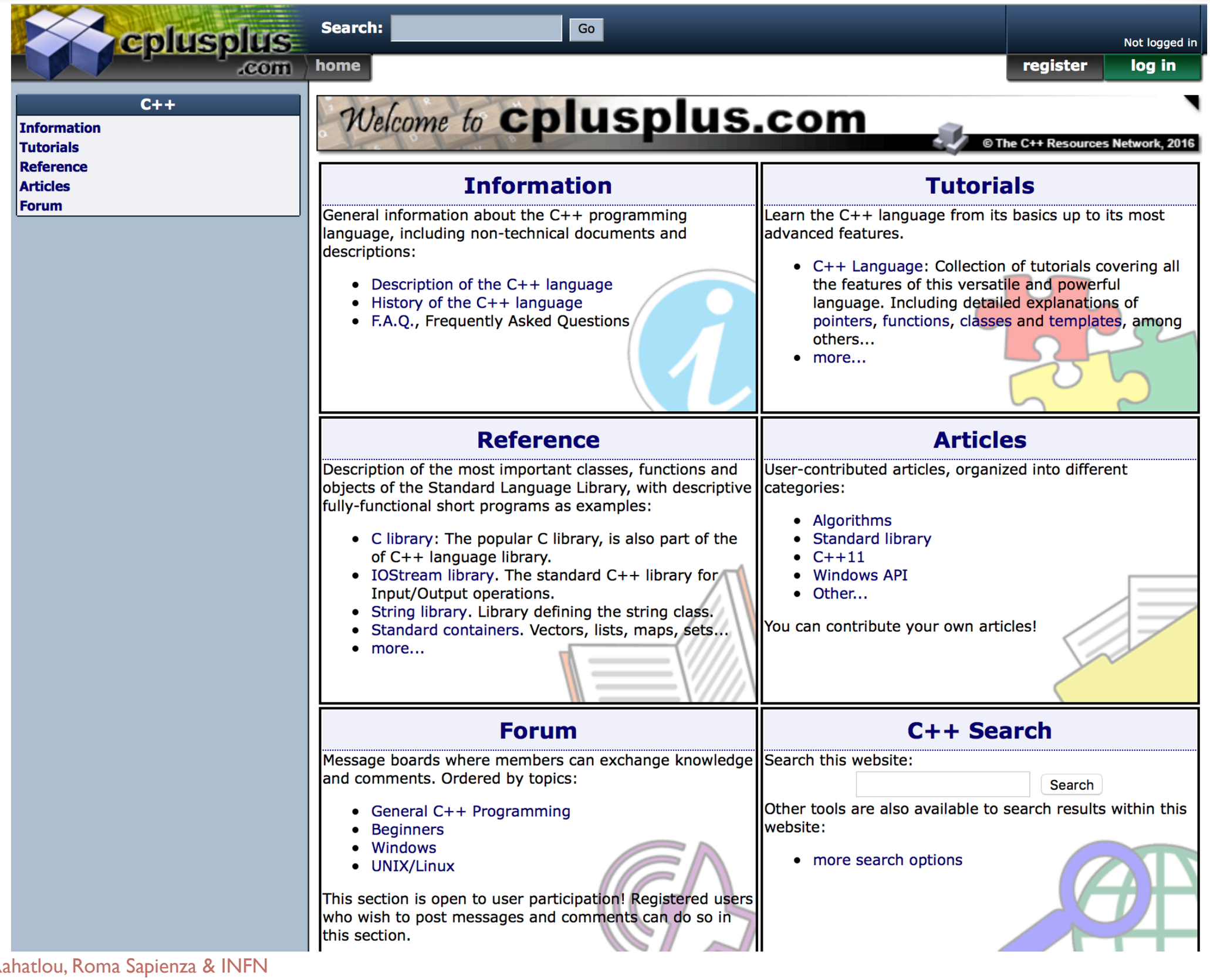




C++ topics covered in this course

- Elements of C++
  - Common features and differences with C
- Introduction to object oriented programming (OOP)
- Class and objects
- Polymorphism and Inheritance
- Abstraction: virtual class and interfaces
- elements of generic programming with *templates*
- error management with *exceptions*
- If we have enough time
  - makefiles and file organisation
  - Libraries and dependencies between objects

Many resources online



Some useful sites

- www.cplusplus.com
  - Tutorial, lezioni, manuali di riferimento, FAQ
- http://www.cppreference.com/
  - Un vero e proprio manuale online