

PHYSICS AT THE TERASCALE
Helmholtz Alliance

Advanced Programming Concepts
23 - 27 June 2014
Max-Planck-Institut für Physik
München

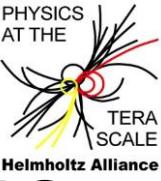
APC2014

Topics:
- Reminder: C++, object orientation, UML, and all that
- Refactoring
- Software patterns and techniques
- OO philosophy
- Performance and design
- Meta-template programming
The lecture programme will be accompanied by in-depth exercises and tutorials.

Participation is limited to 30 participants.
Registration deadline: 13 June 2014.
Fee: 50 Euro
For more information and registration go to:
www.terascale.de/apc2014

Organizing Committee:
Stefan Kluth (MPI),
Maria Grazia Pia (INFN Genova),
Thomas Schoerner-Sadenius (DESY),
Peter Steinbach (MPI-CBG)
Contact: anscon@desy.de

How do particle physicists learn the programming concepts they need?



Stefan Kluth¹, Maria Grazia Pia², Thomas Schoerner-Sadenius³, Peter Steinbach⁴, Thomas Velz⁵
¹MPI für Physik, München, ²INFN Sezione di Genova, ³DESY, Hamburg, ⁴MPI-CBG, Dresden, ⁵Univ. Bonn

The ability to **judge, use and develop** code **efficiently and successfully** is a **key ingredient in modern particle physics.**

Nevertheless, despite the consensus on this appraisal in the HEP community, many concepts and techniques that are essential to working effectively in HEP experiments are not covered in university studies, and limited opportunities are available to students to learn them after

4 workshops in 4 years,
138 participants
across experiments and
HEP related disciplines

The **Advanced Programming Concepts** workshop addresses the needs of programming knowledge that **“normal”** young particle physicists experience in their **everyday activity** in HEP experiments, such as physics analysis and detector development. It differs from other computing schools for physicists, as it bridges the yet unfilled gap between beginner courses and core developer training.

The program introduces basic and advanced **programming techniques**, along with elements of the **software development process** and **project management** skills. It encompasses lectures, extensive hands-on practice and open discussions. Focus is on **concepts and methods**, not only on technology. Emphasis is given to working effectively with existing code and to building a basis for further self-improvement in the field.

PHYSICS AT THE TERASCALE
Helmholtz Alliance

Advanced Programming Concepts
8 - 12 October 2012
DESY, Hamburg

The 2012 school and workshop on „Advanced Programming Concepts“ continues the series started in Dresden in 2010, and it supplements the educational programming part of the GridKit school 2012 <<https://indico.desy.de/conference-Display.py?confId=5219>>.

The school will turn the participants into members of two developer teams working in a highly coordinated way, on a programming project. In the course of the work, basic and advanced programming techniques (class design, design patterns, revision control, build systems etc.), but also project management skills (goal planning, software lifecycle, management styles etc.) will be taught.

The participants of the school are expected to have good knowledge of the C++ programming language and, preferably, also of Python.

Registration deadline: 21 September 2012
The school fee is 50 Euro.
Please register via the school webpage.
Contact: anscon@desy.de

Organizing Committee:
Benedikt Hegner (CERN), Stefan Kluth (MPI München), Thomas Schoerner-Sadenius (DESY),
Hermann Schulz (U. Heidelberg), Peter Steinbach (U. Dresden)

<http://www.terascale.de/apc2012>

1 Jet Finder

- Design classes for a jet finder package
- Input is `vector<FourVector*>`
- User can query
 - Number of jets for given y_{cut}
 - Value of y_{cut} when # of jets changes $N-1 \rightarrow N$
 - Association of input 4-vectors with jets
 - Jet 4-vectors for given y_{cut}

OOAD Examples Stefan Kluth

Software Design

It plays a fundamental role in the software development process and is instrumental to many critical aspects in the life-cycle of an experiment. Transparency of software design enables the validation of physics results, contributes to the effective use of human and computational resources, facilitates the evolution and the maintainability of the software. Nevertheless, young physicists seldom receive an education in software design.

- Principles of Object Oriented Design
- UML
- Design Patterns

Dealing with Existing Software

The workshop addresses the situation that students most commonly face in HEP experiments: using the code of others. They learn methods and techniques to work effectively with legacy code and to improve it.

- Refactoring

Programming Techniques

Some advanced programming techniques are introduced with the support of live coding sessions.

- Design and Performance
- Meta-template programming

3.3 Class Design Principles

- Single Responsibility Principle (SRP)
- Open/Closed Principle (OCP)
- Liskov Substitution Principle (LSP)
 - a.k.a. Design by Contract
- Dependency Inversion Principle (DIP)
- Interface Segregation Principle (ISP)

OO Class Design Principles Stefan Kluth

Refactoring

...is a disciplined technique for improving the design of an existing code

In the ideal world there would be hardly any need for refactoring

In the real world of HEP (and related fields) most software needs to be refactored

By learning refactoring you also learn writing code that minimizes the need to be refactored

Maria Grazia Pia, INFN Genova

“This workshop should be on everyone’s curriculum in Particle Physics.”

A participant of the 2014 APC School