



ORGANISATION EUROPÉENNE POUR LA RECHERCHE NUCLÉAIRE

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

Laboratoire Européen pour la Physique des Particules

European Laboratory for Particle Physics

February 2000
J. Franco-Turner

THE 2000 CERN SCHOOL OF COMPUTING

Marathon, Greece

17 September - 30 September, 2000

Bulletin No. 1

The 2000 CERN School of Computing is organised by CERN in collaboration with the Institute of Nuclear Physics NCSR "Demokritos", Athens, Greece.

1. Date and place of the School

The twenty-third CERN School of Computing will be held at the Hotel Golden Coast in Marathon, Greece, from Sunday 17 September to Saturday 30 September 2000.

The Hotel complex lies on the beach of Marathon, 4 km from Nea Makri, 12 km from Rafina Port and 35 km from Athens. It consists of a main building and bungalows scattered in the gardens and built in traditional Greek architecture. See <http://agn.hol.gr/hotels/goldencoast/location.htm>

2. Participation

The School is open to postgraduate students and research workers with a few years' experience in elementary particle physics, in computing or in related fields. The number of participants will be of the order of 80, mostly from the CERN Member States or from laboratories closely associated with CERN, but a few may come from countries which are not Member States of CERN.

Personal contacts and informal discussions among the participants during leisure time are an important aspect of the School. For this reason, participants are asked to note that they should not be accompanied by family members or friends.

3. Accommodation

Participants will be accommodated in the Hotel Golden Coast complex which lies on the beach of Marathon. All students participating at the School will be lodged in comfortable double rooms with private bathrooms.

The Hotel Golden Coast offers the following facilities:

Three open-air fresh water pools*, one with Jacuzzi, and Pool Bar. Pianist in evenings at the Symposium bar. Atlantis pool bar (mid May - end of September). Sandwiches, ice creams, juices, at Zephyros bar (open-air). Evening Entertainment and disco dancing at Disco Nefeles. Laundry Facilities, Mini Market, Souvenir Shop - Post Card, Newspapers, First Aid Articles, Toiletries. Pottery Exhibition and Traditional Souvenirs. Four tennis courts*, floodlit at night, volley and basket ball court*, mini golf*, table tennis*, water ski and windsurfing, canoes, pedaloos on the beach, jet skis and bicycles for hire.

*free of charge

Financial Support

It is normally expected that funding to cover travel and the cost of the School will be provided by the student's institute. This expectation is explicitly the case for students coming from institutes in CERN Member States¹, or in other industrial nations.

Citizens of Developing Countries may apply for a limited number of SCHOLARSHIPS awarded by UNESCO or by other International Agencies to young physicists or engineers who have not been to previous CERN Schools of Computing. The scholarships cover the cost of the course fee, including board and lodging and in certain cases travel. It is primarily intended for applicants from countries that are not members of CERN. Priority will be given to applicants from developing countries and those not requiring travel assistance.

Applications for support must be submitted on a special application form by **30 May 2000**, together with the formal application to the School, the formal letter of reference and work summary, as mentioned under the section "Application". An application form is enclosed with this Bulletin. Additional forms for Scholarships may be obtained from Mrs. J. Franco-Turner, CERN School of Computing, CERN, 1211 Geneva 23, Switzerland, or via Computing.School@cern.ch. Please note that the application for Scholarship **must** be submitted together with the School application form. **Requests submitted at a later date will not be considered.**

1. Cancellation

The Advisory Committee reserves the right to refuse reimbursement of part or all of the fee in the case of late cancellation. However, each case of cancellation would be considered individually.

2. Replacement

In all cases of withdrawal or cancellation, whether last-minute or otherwise, the choice of a replacement, if any, will lie entirely with the Advisory Committee and not with the laboratory concerned.

3. Advisory Committee

W. Carena	CERN, Geneva	
S. Cittolin	CERN, Geneva	
M. Delfino	CERN, Geneva	
F. Etienne	CPPM, Marseille	(Chairman)
F. Flückiger	CERN, Geneva	
L.O. Hertzberger	University of Amsterdam, Amsterdam	
A.J.G. Hey	University of Southampton, Southampton	
R. Jacobsen	University of California, Berkeley	
R. Jones	CERN, Geneva	
G. Kellner	CERN, Geneva	
J. Franco-Turner	CERN, Geneva	(School Administrator)
C. Vandoni	CERN, Geneva	(School Director)
D.O. Williams	CERN, Geneva	

¹CERN, the European Laboratory for Particle Physics, has its headquarters in Geneva. At present, its Member States are Austria, Belgium, Bulgaria, the Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Italy, Netherlands, Norway, Poland, Portugal, Slovakia, Spain, Sweden, Switzerland and the United Kingdom. Israel, Japan, the Russian Federation, the United States of America, Turkey, the European Commission and Unesco have observer status.

4. Scientific programme

The School is based on the presentation of approximately 37 lectures and on 20 hours of related practical exercises on PCs or workstations.

The programme of the Schools is organised round three themes:

Distributed Computing
OO Design and Implementation
Storage and Software Systems for Data Analysis.

The following lectures are now confirmed. Any additional lectures will be announced later.

Distributed Computing - this series of lectures will be complemented by exercises

The requirements of HEP computing for the next generation of experiments at CERN and in the other major HEP laboratories will demand a world-wide access to very large amount of data and massive aggregate computing capacity.

High performance networking, WEB access, distributed computing models (such as Corba) and metaphors such the "GRID" are necessary ingredients of the final solution.

This track will review the status-of-the-art in high performance networking and introduce concepts such as the GRID, as well as OO methodology such as CORBA.

A series of examples taken from the GLOBUS tool kit and exercises will complete the course.

Lectures:

G. Aloisio and P. Falabella, University of Lecce

"The Use of Computational and Data Grids for HEP"

These lectures will provide a practical introduction to "computational grids" and to the technologies required for building applications on heterogeneous and distributed environments. Issues related to the secure access to the grid, the "intelligent" management of distributed resources, the access to remote files, the starting and steering of remote applications will be discussed in the theoretic part of the lectures. During the practical part, the presentation of a case study will allow the attendees to learn gradually the problems that arise in a real grid-enabled application. Special emphasis will be given to the problems that are more common in the HEP applications such as the management of large datasets.

F. Fluckiger, CERN

"Recent Advances in Networking Technologies"

S. Kolos, PNPI and CERN

"CORBA".

This tutorial will give a practical introduction to the Object management Group's Common Object Request Broker Architecture (CORBA). After a short introduction of OMG and history of CORBA the speaker will describe CORBA architecture and vocabulary. He will then describe how to implement programs in CORBA and will conclude with a survey of current CORBA activities and future plans. The course will be completed by practical exercises.

B. L. Tierney, LBL

"An Overview of Grid Computing and the Data Grid"

Introduction

What is a Computational Grid?

What is a Data Grid?

Grid Middleware

- Globus (<http://www.globus.org>)
- Legion (<http://legion.virginia.edu/>)
- Storage Resource Broker (<http://www.npaci.edu/DICE/SRB/>)

Sample Current Grid Components

- Globus Grid Security Infrastructure (GSI)
- Globus Grid Information Service (GIS)
- Globus Resource Management (GRAM)

Current Grid Research and Development

- Data Grid Services
- Grid Monitoring Service

Suggested Reading List:

The Grid: Blueprint for a New Computing Infrastructure, I. Foster, C. Kesselman (Eds), Morgan Kaufmann, 1999 in particular, see chapters 2, 4, 5, and 11.

"A Data Intensive Distributed Computing Architecture for Grid Applications", Tierney, B., Johnston, W., Lee, J., Thompson, M., Future Generation Computer Systems, Elsevier Journal, April, 2000. Available from: <http://www.didc.lbl.gov/publications.html>

The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Data Sets, A Chervenak, I Foster, C Kesselman, C Salisbury, S Tuecke, Internet II Network Storage Symposium, Oct. 1999, <http://dsi.internet2.edu/netstore99/>

OO Design and Implementation - this series of lectures will be complemented by exercises

Object-oriented concepts are now a fundamental part of the design of many applications. The basic concepts of data encapsulation, polymorphism, and inheritance provide an elegant way to separate the specification of how we interact with a computation from the way that computation is implemented. The course will introduce this new paradigm on the design of and the implementation of scientific software packages. The first part of the track will focus on the merit of C++ and Java with respect to the OO paradigm and the second part of the course will demonstrate how Geant4* and JAS** exploits the advantage of those languages in the context of scientific/engineering computing.

*Geant4: An Object-Oriented Toolkit for Simulation in High Energy Physics

**Java Analysis Studio: JAS is the tool aimed at the easy, intuitive and powerful analysis of the HEP data using Java language.

Lectures:

M. Asai, Hiroshima Institute of Technology

Lecture 1

"Overview of Object orientation"

The first hour of this track is dedicated to the general concepts of Object orientation which are common to all modern programming languages. The most basic and important three concepts, encapsulation, abstraction and polymorphism, are explained with examples mostly taken from simulation and analysis domains. It will be stressed that the concepts of Object-orientation are much more important than syntax of language.

Lectures 2-3

"OO design and implementation with C++"

This lecture explains the way of designing and implementing C++ programs which take full advantages of Object orientation. The use of the so-called OO language does not guarantee the benefits of object-oriented programming. This lecture does not go explore every syntax of the C++ language but introduces the systematic design and implementation based on the concepts of Object orientation.

Lecture 4

"Geant4 - Overview"

This lecture introduces the global structure of Geant4, an Object-oriented toolkit for simulation in high energy physics written in C++, with emphasis on how the concepts of Object orientation are adopted on its design. This lecture also covers the general way of designing the Geant4-based user's simulation application.

Lecture 5

"Geant4 - Designing and implementing a detector"

This lecture introduces the details of designing and implementing a detector for a Geant4-based simulation program. The ways of describing a detector for both its geometry and its sensitivity are covered. The concepts of Object orientation will be fully adopted to the detector description.

Lecture 6

"Geant4-based simulation application"

This lecture covers various key issues that must be considered for the user's simulation program based on Geant4. The issues include how to select the proper physics processes, how to shoot primary particles, how to get and analyse useful information from the detector, and how to visualise the detector and simulated results.

Geant4 Exercises

A simple detector set-up is given to the students and then they must design, implement and execute its simulation program based on Geant4. The detector set-up contains some of the most popular components widely used in HEP experiments.

- 1) Design and implement a detector description; not only its geometry and material description but also its sensitivity must be considered.
- 2) Design and implement several mandatory and optional user classes for Geant4-based simulation program.
- 3) Execute the simulation program. Get simulation results and also visualise the implemented detector set-up and simulated results.

Geant4 + JAS Exercise

Using the simulation program made at the Geant4 exercises, the students are requested to produce simulated data and to analyse it with JAS. Through this exercise, the students will learn how to store simulated data made by Geant4-based simulation program and how to analyse it using JAS.

- 1) Implement a class which stores simulated data with an appropriate format that JAS requires. The class must be derived from an abstract base class defined in Geant4.
- 2) Analyse the data with JAS.

A.S. Johnson, SLAC

"Overview of JAS"

Lecture 1:

This talk will give an overview of the Java language, Java virtual machines, and the large set of standard libraries and extensions available for Java. We will describe how the emphasis in Java on using abstract interfaces, and dynamic code loading, affects program design, and how the dynamic nature of Java is ideally suited to building distributed systems.

Lecture 2:

An introduction to Java Analysis Studio (JAS), a tool written in Java, and in which Java is the language used to perform data analysis. We will emphasise how the unique features of Java introduced in the previous lecture influenced the design of the system, and give examples of the use of JAS in several experiments. We will also explore how OO techniques have been used to build a system from modular components including visualisation, fitting and data-access components that can be used together, or on their own.

Suggested Reading.

Thinking in Java, Bruce Eckles (available online at <http://www.bruceeckel.com/javabook.html>)
Java in a Nutshell, David Flanagan (or any of the many O'Reilly Java Books, at <http://java.oreilly.com/>).

Useful Links:

Java Home Page: <http://java.sun.com/>

Sun's Java Tutorial: <http://java.sun.com/docs/books/tutorial/>

Java Analysis Studio: <http://www-sldnt.slac.stanford.edu/jas>

LCD JAS Tutorial: <http://www-sldnt.slac.stanford.edu/jas/documentation/lcd/start.html>

Storage and Software Systems for Data Analysis - this series of lectures will be complemented by exercises

The track addresses the challenge of storing event data and designing analysis systems for future High Energy Physics experiments. The demand for large amounts of reliable and high-availability storage continues to increase more significantly each year. Today everyone can have the same computer, the same hardware, the same network appliances, but no one can have your data. It is the data itself that is the DNA of today's leading-edge organisations. The lectures on storage systems will address current and long-range directions in data storage highlighting key and emerging technologies, storage architectures, and storage-intensive applications. The lectures on analysis systems provide an overview of the LHC++ and ROOT analysis frameworks, and cover those aspects of software engineering most relevant for HEP software development. The track combines software engineering lectures with exposure to the software technologies and packages relevant for LHC experiments. It shows, in a practical sense, how software engineering can help in the development of HEP applications based on the various data analysis software suites and also gives a taste of working on large software projects that are typical of LHC experiments.

Lectures:

R.G. Jacobsen, University of California

"Introduction"

This lecture explains the purpose and goals of the track and outlines the schedule and organisation. We then discuss how HEP physics data is taken, processed and analysed, with emphasis on the problems that data size and CPU needs pose for people trying to do experimental physics. The role of software engineering is discussed in the context of building large, robust systems that must at the same time be accessible to physicists. We close with some examples of existing systems for physics analysis, and raise some issues that will be addressed in the rest of the track.

R. Jones, CERN

"Software Engineering - Introduction"

This lecture introduces the subject of software engineering. It includes an overview of OO methods, in particular UML, and the software process. The tasks involved in requirements gathering and problem analysis are addressed.

A. Pfeiffer, CERN

"Overview of LHC++"

The lecture gives an overview of the various components of the LHC++ suite and their relationship. The lecture will use some simple concept of UML modelling.

R. Brun, CERN and F. Rademakers, GSI
"Overview of ROOT"
This lecture gives an overview of the ROOT system, the design philosophy, the basic architecture, the different components and their relationships.

R. Jones, CERN
"Software Design"
The lecture presents the design of software as three levels: architecture, mechanistic and detail. Various UML diagram types are introduced and the benefits of patterns are highlighted. The transition from design to implementation is described.

M. Nowak, CERN
"Data Storage and Retrieval in LHC++"
The persistency components of LHC++ are explained thoroughly, with emphasis on the decoupling of logical/physical models, use of OO modelling techniques and advanced persistency tools such as Objectivity/DB and Espresso.

A. Pfeiffer, CERN
"LHC++ Fitting Components"
The lecture introduces students to the basic concepts of minimization and fitting and then explains in detail how LHC++ components implement those features. Some basic knowledge of maths is required (e.g. derivatives, gradient).

R. Jones, CERN
"Software Testing"
The lecture explains why programs have defects (bugs), classifies defects, and explains why debugging is different from testing. Popular testing techniques and tools are presented.

R. Brun, CERN and F. Rademakers, GSI
"ROOT Persistency"
One of the most important issues that need to be solved by a data analysis framework is the problem of data storage and retrieval. The lecture gives an overview of how this is solved in the ROOT environment, from simple object storage to the advanced TTree object containers. The lecture also explains how very large databases can be built and managed using a combination of object-oriented and relational database management techniques.

R. Brun, CERN and F. Rademakers, GSI
"ROOT Data Analysis Facilities"
This lecture gives an overview of the different data analysis and visualization features of the ROOT system. It describes the use of histograms (1D, 2D and 3D), profile histograms, functions, fitting, random number generators and the way they can be visualized.

A. Pfeiffer, CERN
"Visualization for Data Analysis in LHC++"
The lecture introduces students to the graphics packages used in the context of LHC++ (e.g., Qt, OpenInventor), discusses their domain of application and shows how they integrate with the rest of the framework.

R. Brun, CERN and F. Rademakers, GSI
"ROOT GUIs"
This lecture describes the ROOT Graphical User Interface classes with special emphasis on Event Displays accessing remote data bases in a distributed environment. Threads and Sockets classes are also explained in this context.

R. Jones, CERN
"Long Term Issues of Software Building"
The lecture starts by identifying some of the major reasons why software projects fail. The advantages of an iterative development cycle are explained and the importance of release management tools is highlighted.

Wrap up (all lecturers)
This session gives the opportunity for students to give feedback and to ask questions about problems and issues they have discovered during the exercises and lectures.

F. Moore, Horizon Information Strategies
"Current and Future Trends in Data Storage Applications"
These lectures address the key factors in the data storage industry for today and tomorrow. The Internet alone is presently driving storage demand at over 90% per year. Over half of the data created today is born digital. From a storage technology perspective, the 1990's decade witnessed more advancement than there had been in the entire history of the data storage industry. The student will better understand the fundamental strategic issues needed to cope with the dramatically increasing demand for digital storage. The lectures will include the following sections:

- 1) Trends and projections for high-performance and high-availability systems
- 2) Magnetic Disk directions and technology
- 3) Magnetic tape, library architectures and mass-storage systems
- 4) Storage-Intensive Applications, the demand drivers
- 5) Advanced technologies (a look into the labs)
- 6) Storage Networking, the model for the 21st century.

Handouts: Each student will receive a copy of the paper "Storage Panorama 2000"

LHC++ Exercises:

A problem statement is given to the students that they must analyse, then design, implement in C++ and test using the LHC++ suite. Essentially, the case study will require students to develop four programs in succession during the practical exercises:

- 1) Populate an event database according to a defined object model and retrieve some summary information (e.g. data quality estimators).
- 2) Use of LHC++ minimization/fitting packages to solve typical problems in data analysis (e.g. determination of an unknown physics quantity with its errors).
- 3) Interactive Data Analysis and visualization exercise. Students will be asked to carry out a data analysis task that will involve the use of various visualization techniques.
- 4) Mini-project building on the three previous exercises.

ROOT Exercises:

In the exercises the students will learn how to use ROOT to analyse typical physics data. We plan to use, as much as possible, real data from existing experiments (BaBar).

- 1) Code a few C++ classes, integrate them into the ROOT framework via a dynamically loadable shared library. Exercise the classes interactively via the C++ interpreter and store and retrieve objects to and from a ROOT database.
- 2) Exercise with Root Trees. This will show how to use an Event tag tree to select events in a chain of large Trees. The Trees serial and split mode will be used.
- 3) Perform some basic data analysis (histogramming, fitting, etc.) using a ROOT database containing recent experimental data. The selection mechanisms using the mouse, the interpreter or code dynamically compiled and linked will be part of this exercise.
- 4) Build a simple Event Display GUI combined with the data analysis tools.

5. Language

The working language of the School will be English. There will be no simultaneous translation. Participants should therefore have sufficient understanding of English to enable them to benefit from the School.