Introduction to Programming and Computing for Scientists (MNXB01 / NAFY018)

Introductory meeting

DIVISION OF PARTICLE PHYSICS (www.hep.lu.se)
teachers:

OXANA SMIRNOVA (course coordinator)
EINAR ELÉN & ERIK WALLIN (C++)
FLORIDO PAGANELLI (Linux, GIT, IT)
BALÁZS KÓNYA (distributed computing)
ELSE LYTKEN (ROOT)

Attendance list

- Please fill the attendance list at:

  https://survey.mailing.lu.se/Survey/42552

1. Learning outcomes

- Knowledge of computing and programming is **essential** in modern sciences
- Acquired knowledge and understanding:
  - Understanding and use of the concept of **program libraries**
  - Knowledge of processing, analysing and modelling **scientific data** using custom software
  - Basic understanding of **code optimisation** issues, implications of machine accuracy
  - Awareness of **frameworks** that use programmatic interface
- Acquired skills and abilities:
  - Ability to work in UNIX-based operating systems, particularly **Linux**
  - Ability to write computer programs using **C++**
  - Ability to **compile** from source, build and debug computing programs
  - Ability to develop and document program code in a **collaborative** environment

2. Required knowledge

- The course is aimed towards **<u>beginners</u>** who have little or no knowledge of computers
  - For those who already know programming some parts will still be interesting
- Some essential knowledge is still required:
  - Good knowledge of mathematics, including basic calculus
  - Basic knowledge of statistics is a bonus
  - Good knowledge of English

3. What this course is not

- This is not a C++ programming course
  - You will learn basics of C++ , needed to be used with the ROOT data analysis framework
  - You will not learn various algorithms, rather, you will learn how to use existing program libraries
  - You will learn how to understand other’s code, and how to make others understanding yours
  - You will learn how to work with scientific data in the Open Science age

- For a deep knowledge of C++, programming algorithms, methods and tools, please refer to dedicated courses

4. Course plan and logistics

- We are using the Canvas learning management platform, log in here:

  CANVAS https://canvas.education.lu.se

- We will use computers at the LUNARC data center for all practical work, including the project
  - Make sure you have set up the account there (see the announcement in Canvas)
- The course uses Active Learning approach, with theory introductions followed by practical problem-solving classes and homework
  - Theory lectures typically every Wednesday at 15:00
    - With some exceptions
  - Problem-solving tutorials on Thursdays and Fridays
  - Your *Stil-login* should work for Canvas and SUPR; if not, please get in touch with LU Servicedesk
- In total, 8 lectures and 16 tutorials
  - Presented by 6 teachers (best experts!)
  - Plus homework, which is mandatory
  - Calendars in Canvas and Timeedit are in sync time-wise
- It is important not to miss classes!

5. Assessment

Home assignments

- **50%** of the final grade
- 12 graded assignments and a questionnaire
- **Mandatory** to submit, even if incomplete
- Each assignment is graded on the scale from 0 to 5

Project

- **50%** of the final grade
- Done in teams of 3-4 students
- Task: use ROOT and C++ to write code that opens text files with data, performs analyses and plots the results
- Project reports must be handed in towards the end of the first half-term and presented by the teams (preliminary date: Nov. 5)
- Projects are graded on the scale from 0 to 5; every team member receives the same grade, unless the contribution is outstanding either way

6. More about the project

- The project goal is to analyse data from climate observations using C++ and ROOT:
  - Data in plain text files (typically comma-separated lists)
  - Develop C++ code to read data within ROOT
  - Use ROOT to plot data and develop simple models (e.g. an extrapolation or a fit)
  - Use Github to share code among team members
    - Example: one student develops code to read data, another develops code to plot it, and others develop models
  - Write the final report (few pages) in LaTeX
    - Can use Overleaf, but keep the result in Github
- Students will work in groups of 3-4
  - Sign up to a project team in Canvas, deadline **September 30**
    - Small teams will be merged after that date
  - Each team should create an own Github project area
    - A team can appoint a code manager
- Detailed instructions, examples, data sets and basic templates are available in Canvas
- Project consultations in end-October: come with a work plan and some initial code
- Project presentations: October 28 (to be confirmed)
- Report can be finalised after the presentation, but you should have at least a good draft by October 28

7. Course literature

- Main material: slides and references therein
  - Referred manuals are available on-line for free
- Official textbook: *“Problem solving with C++”, W. Savitch*, any edition
  - Covers ~50% of the course