University of Notre Dame

PHYS 60070: Computing and Data Analysis for Physicists

Course Description and Syllabus - Fall 2015

Author: Anna Simon

August 25, 2015

Course Improvements, Syllabus Modifications, and Updates

Computing and Data Analysis for Physicists is a living course which evolves with time. This means the syllabus may change to better achieve our ever changing academic goals. You will be notified of any changes either verbally or literatively, and will be expected to adhere to all changes pre and post hoc.

---

1 Class Schedule

Venue: Room 412 Jordan Hall
Class Days: Wednesday and Friday, Class Time: 2.00pm-3.15pm
Session Type: Lecture, Task-based learning

2 Contact Information

Anna Simon, Rm. 221 NSH, email: simon.instructor@nd.edu
Emails send outside of the hours of 9:00am through 5:00pm on Monday through Friday, may not receive a response until those following hours.

Office hours: Monday?

3 Course offered for

Graduate students, undergraduate junior/seniors

4 Requirements

Basic knowledge of Linux/Unix, basic programming
CRC account

5 The Course

5.1 Course Goals

The primary goal of this course is to provide an overview of experimental-data analysis utilizing the ROOT libraries through C++ in a lecture-laboratory setting. The course will lead with a focus on program design including: control structure, variables, functions, file I/O, and classes. Then it will continue with an introduction to ROOT including its data structure and conversion, and graphics output. It will finish with the use of macros and incorporation of ROOT libraries into C++ programs for the purpose of building data analysis structures tailored to a specific experiment or system.

Upon successful completion of the course the students:

- will be familiar with elements of the C++ language,
- will be able to read and explain pieces of C++ code,
- will be able to utilize ROOT environment for data analysis,
- will write a complete data analysis code using C++ and ROOT libraries,
- will be able to create publication ready scientific figures using ROOT.

5.2 Course resources

All the materials will be provided via Sakai or during the class.

6 Grading

The various components of the course contribute to the final grade as follows:

In-class tasks and homeworks	65%
Attendance	5%
Final projects	30%

6.1 Assesment types

6.1.1 In-class tasks

The main part of the course is designed as task-based learning. After a short introduction to a new topic, during the scheduled class time students are required to complete part of the tasks from the list provided by the instructor. The remainder of the tasks will be assigned as homework due by the end of the day Tuesday of the following week (i.e. all the homework from week one, both Wednesday and Friday, is due by 5pm Tuesday of the second week of classes).

6.1.2 Final project

There will be a final project assigned to the students at the end of the course that will require writing a code that fulfills all the specifications of the project. It will combine both C++ and ROOT, and will be a self contained executable designed to analyze a data set provided by the instructor.

6.2 Requirements for successful completion of the assignments and projects

Each code submitted for grading has to compile and/or run either on the computers in the classroom or on the CRC cluster. Codes that do not compile will not be graded! Each programming assignment will be graded based on the following criteria:

- Clarity of the code
  - The code has to include a header comment explaining its purpose, author and date of creation

  - Comments have to be present within the code explaining briefly what a given part of the code does
  - The code has to use indentation to make it easy to read (it is up to the student to choose their preferred style)
  - Variable names should be meaningful (avoid using letters of alphabet as variable names)
- Use of the appropriate structures
  - When appropriate (or indicated in the instructions) header files and preprocessor declarations have to be used
  - The code has to use the language structures specified in the instructions
- Functionality of the code
  - The code has to perform all the tasks listed in the instructions
- Student's understanding of the code
  - Student has to be able to explain any part of the code, its purpose and outcome to prove their understanding and ownership of the code
  - Student has to be able to modify their code upon instructor's request

6.3 Grading Scale

Total percentage achieved for any assignment or project, is converted to a letter grade using the table below.

upper limit	grade	lower limit
	A	≥ 94
$94 >$	A-	≥ 91
$91 >$	B+	≥ 88
$88 >$	B	≥ 82
$82 >$	B-	≥ 79
$79 >$	C+	≥ 76
$76 >$	C	≥ 70
$70 >$	C-	≥ 67
$67 >$	D	≥ 64
$64 >$	F	

7 Course Rules

7.1 Cell Phone Policy

As soon as class begins all electronic devices, e.g. smart phones, tablets, laptops (unless used for the course), etc., must be turned off and put away.

7.2 Safety

As the classroom is both a discussion area as well as a laboratory, students should be mindful of certain dangers; e.g. tripping hazards. Students should keep backpacks and other items stowed away.

The course will be held in a room with computer equipment, thus no open-container drinks will be allowed in the classroom.