Outline 2017/06/05

- Summary of key points
- Examination: an example
- Exercise:
  - solve an old examination test

Examination, dates proposal

2017 JUNE

SUN	MON	TUE	WED	THU	FRI	SAT
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

www.free-printable-calendar.com

2017 JULY

JN	MON	TUE	WED	THU	FRI	SAT
						1
	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

www.free-printable-calendar.com

Important notice!!

Starting from 29/05/2017, there is a new version of OS installed on INFIS PCs. Concerning the C++ course you should care about these major differences:

1) the system architecture is now **64 bit**. This means that any program you have compiled should (must) be recompiled in order to work properly.

2) the default ROOT installation as been upgraded to **ROOT 6**

3) the default **GEANT4** installation has **not** been **re-compiled** in 64bit and it is not working properly (at the moment).

I have re-compiled GEANT4 and ROOT5 in my own directory.
You should be able to access my installation by sourcing the proper setup file (one for both ROOT5 and GEANT4):

```
source /home/mocchiut/setup_infis.sh
```

Once sourced this file you should be able to work as we have done till now.
If you have any problem or notice anything strange please contact me!

Single pdf with all slides

I'll put a single pdf with all slides on moodle, however remember:

- slides are not a textbook (indeed things are compressed and simplified to be put on slides)
- if you have doubt or question about the course just contact me
- there are books and manuals on C++, ROOT and GEANT4 which describe in details much more than it was described in this course. Use them if you want to learn more or you need a better/more deep explanation of details:
  - http://www.learncpp.com
  - "*The C++ programming language*" Bjarne Stroustrup, Addison-Wesley Professional, 3 edition (1997), ISBN: 978-0201889543 , http://www.amazon.com/C-Programming-Language-3rd/dp/0201889544/ref=sr_1_4?s=books&ie=UTF8&qid=1349104542&sr=1-4
  - https://root.cern.ch/root/htmldoc/guides/primer/ROOTPrimerLetter.pdf
  - http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/fo/BookForAppliDev.pdf

What you should have learnt

Basic knowledge

- **C++** compiling and running an executable
  - slides 278-283, 423-428
  - slide 506
- **C++** creating loops and conditional statements
  - slides 111-121
  - exercise 3, slide 693
- **ROOT** opening, reading and writing a TFile
  - examples 1 and 2, slides 537-546, ex. 7 slide 559, ex. 8-12 slides 605-607
  - exercise 2, slides 691-692
  - class as data containers 562-571
- **ROOT** creating, filling and saving histograms
  - exercise, slides 548-554
  - exercise 7, slide 559
  - collection classes 596-602, exercises 12 slide 607
- **ROOT** fitting distributions and getting access to results
  - exercise 2, slide 692
  - exercise 4, slides 694

Basic knowledge

- **GEANT4** compiling and running a simulation
  - slides 770-773
  - slide 807-817, 871
- **GEANT4** adding and removing volumes
  - slides 775
  - exercise 1-3, slide 781
- **GEANT4** changing primary distributions and particles
  - slides 799-802
  - example 803-805
  - exercise 4 slide 842
- **GEANT4** getting data and writing to a TFile
  - slides 838, ex. 5 slide 843
  - slide 852-866
  - exercises 5 slide 872

Basic knowledge

About opening, reading and writing a TFile:

- obviously it is possible to open more than one TFile per time in the same program, for example the code that follows in the next page copy one variable from a file to another one

Basic knowledge

```
#include <TFile.h>
#include <TTree.h>

#include <PamCalo.h>

void copy(TString inputFile, TString outputFile){

  // Open the input file
  TFile *file = TFile::Open(inputFile,"READ");
  TTree *tree = (TTree*)file->Get("pamcalotree");
  PamCalo *pc = new PamCalo();
  tree->SetBranchAddress("PamCalo",&pc);

  // Open the output file
  TFile *outfile = TFile::Open(outputFile,"RECREATE");
  outfile->cd();
  TTree *outtree = new TTree("outtree","PamCalo copy");
  Float_t energy = 0.;
  outtree->Branch("energy",&energy,"energy/F");

  // loop over events
  for ( Int_t i=0; i<tree->GetEntries(); i++) {
    if ( i%10000 == 0 ) cout << " get entry " << i << "\n";
    tree->GetEntry(i);
    energy = pc->energy;
    outtree->Fill();
  }
  file->Close();
  outfile->cd();
  outtree->Write();
  outfile->Close();
}
```

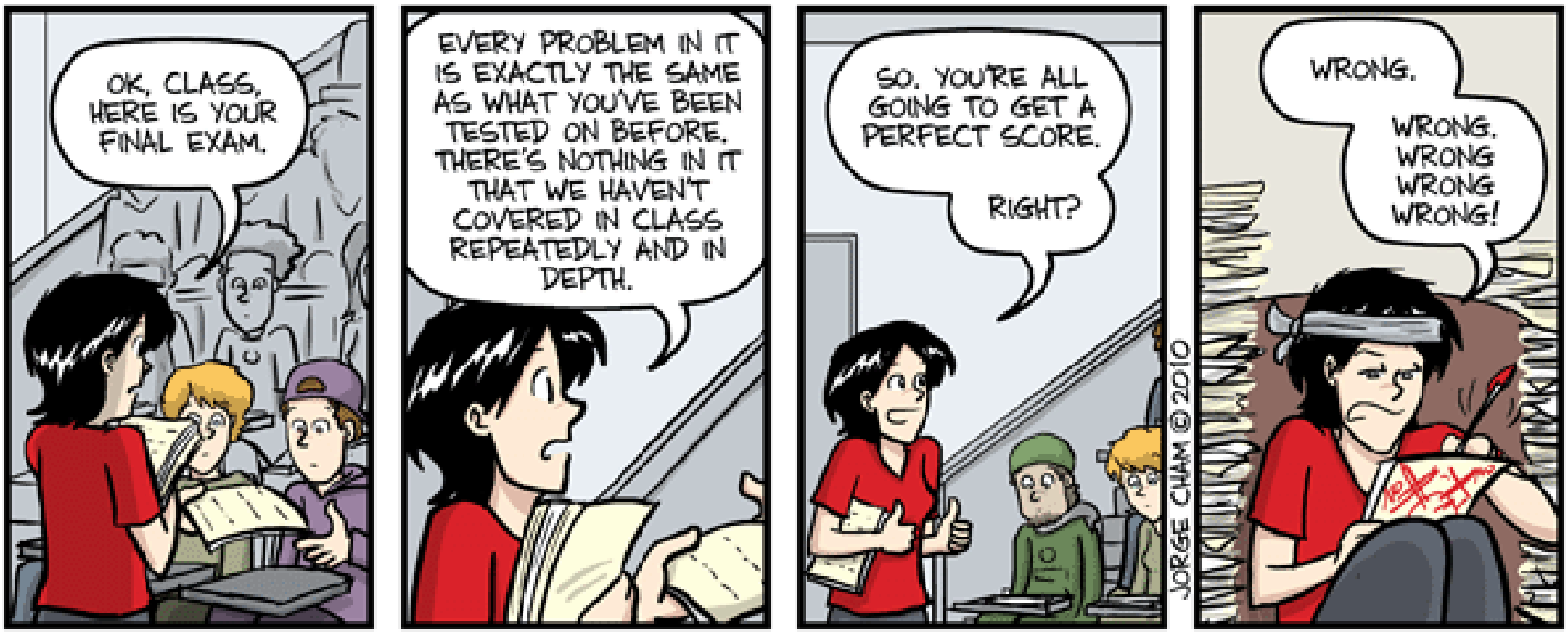
Important knowledge

- Proper use of terms when discussing the code (pointers, instantiations, objects, methods, members, etc.)
- Proper formatting of output histograms (axis, readability, labels, titles, colors, symbols, etc.)
- Proper use of comments inside source code
  - slides 82-88
- Coherent and well structured programming style
  - slides 89-96
- Capability of correctly translating real life requests into a properly written logical source code
  - e.g. slide 288 (just for logical request not overloading)
- CINT, ROOT, GEANT4 general usage knowledge

Important knowledge

- ***Read and understand WARNINGs and ERRORs!!***
- Read carefully, spellcheck, cross-spellcheck, spellcheck again...
- Computers do not have feelings, do not hate or love... be logic, read the code line per line as a computer would (almost) do.
- Do not take anything for granted
- You are smarter than a PC!!

Programming test, example



http://www.ts.infn.it/~mocchiut/C++2015/examination_example.pdf

Examination: example

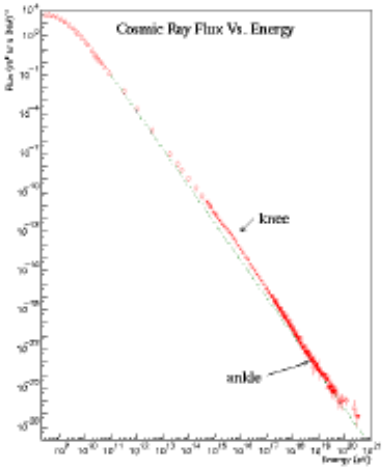
Particle fluxes and spectrum

Often in physics we have to deal with particle spectra. A particle spectrum is defined as the variation of the particle fluxes as function of the particle energy. Analytically, a particle spectrum is represented by the equation

$S(E) = N / (T \times G \times E \times \varepsilon)$ [s cm^2 sr GeV]$^{-1}$,

where N is the number of particles of energy E [GeV] passing through the solid angle G [cm^2 sr] in the integrated time T [s] and detected with efficiency ε.

Experimentally, precise determination of particle spectra requires an excellent knowledge of the detector – that is of the acquisition time (T), geometrical factor (G), energy determination and efficiencies – plus a clean selection of a data sample.



On the right is shown, as an example, the spectrum of cosmic rays. Notice the double log scale and notice that, in the case of cosmic rays, the spectrum can be roughly described with power law functions, that is : $F(E) = A\ E^{-\gamma}$,

where A is the normalization factor and γ is usually called "spectral index".

Spectra determination is of crucial importance in many fields of physics for example to determine sources composition, looking for new particles, study the amount of traversed material and its distribution.

Examination example: solution

em_main.cpp

```
#include <TFile.h>
#include <TTree.h>
#include <PamCalo.h>
#include <math.h>

int main(){

 TFile *file = TFile::Open("/home/mocchiut/pamela/data/pamsimu2013.root","READ");
 TTree *tree = (TTree*)file->Get("pamcalotree");
 PamCalo *pc = new PamCalo();
 tree->SetBranchAddress("PamCalo",&pc);

 TFile *ofile = TFile::Open("miofile.root","RECREATE");
 TTree *otree = new TTree("outtree","il mio tree");

 Float_t energy = 0.;
 Float_t qtotene = 0.;
 otree->Branch("energy",&energy,"energy/F");
 otree->Branch("qtotene",&qtotene,"qtotene/F");

 for (int j=0;j<tree->GetEntries();j++){
   tree->GetEntry(j);


   if ( pc->energy < 0. ){
     energy = fabs(pc->energy);
     qtotene = 0.;
     if ( pc->energy != 0. ) qtotene = pc->qtot/fabs(pc->energy);
     otree->Fill();
   }

 }
 file->Close();
 ofile->cd();
 otree->Write();
 ofile->Close();
 return 0;
}
```

approximate!
you can do better!!

Examination example: solution

em_main.cpp | exercise2.C

```
void exercise2(TString inputfile){
 TFile *file= TFile::Open(inputfile,"READ");
 TTree *tree = (TTree*)file->Get("outtree");
 Float_t energy = 0.;
 Float_t qtotene = 0.;
 tree->SetBranchAddress("energy",&energy);
 tree->SetBranchAddress("qtotene",&qtotene);

 TH2D *h2 = new TH2D("h2","qtotene vs energy;energy [GeV]; qtotene",1000,0.,20.,1000,0.,1000.);
 TH1D *h1 = new TH1D("h1","flux;energy;events",1000.,0.1,20.);

 Float_t x[9] = {1.,2.,3.,4.,5.,7.,9.,13.,18.};
 TH1D *h1b = new TH1D("h1b","flux;energy;events",8,x);

 for (Int_t i=0; i<tree->GetEntries(); i++){
        tree->GetEntry(i);
        h2->Fill(energy,qtotene);
        if ( qtotene>150. && qtotene <400. ){
         h1->Fill(energy);
         h1b->Fill(energy);
        }
 }
 Float_t epsilon = h1->GetEntries() / h2->GetEntries();
 cout << " epsilon = " << epsilon << endl;
```

approximate!
you can do better!!

Examination example: solution

em_main.cpp | *exercise2.C

```
cout << " epsilon = " << epsilon << endl;

TCanvas *c = new TCanvas("c","histo");
c->Divide(1,2);
c->cd(1);
c->GetPad(1)->SetTicks();
h2->Draw("colz");
c->cd(2);
c->GetPad(2)->SetLogx();
c->GetPad(2)->SetLogy();
c->GetPad(2)->SetTicks();
h1->Draw();
c->SaveAs("exercise2.pdf");

TCanvas *c2 = new TCanvas("c2","histo2");
c2->Divide(1,2);
c2->cd(1);
h1b->Draw();

Float_t xx[9];
Float_t yy[9];
Float_t eyy[9];
for (Int_t i=0; i<9;i++){
  yy[i] = h1b->GetBinContent(i+1) / (epsilon * h1b->GetBinWidth(i+1) * 1000. * 21.);
  eyy[i] = sqrt(h1b->GetBinContent(i+1)) / (epsilon * h1b->GetBinWidth(i+1) * 1000. * 21.);
  xx[i] = h1b->GetBinCenter(i+1);
}
TGraphErrors *tg = new TGraphErrors(9,xx,yy,NULL,eyy);
TH2D *sfondo = new TH2D("sfondo",";GeV;flusso",1000,0.5,20.,1000,1e-4,2.);
c2->cd(2);
c2->GetPad(2)->SetLogx();
c2->GetPad(2)->SetLogy();
c2->GetPad(2)->SetTicks();
sfondo->Draw();
tg->SetMarkerStyle(20.);
tg->Draw("Psame");

TF1 *fun = new TF1("fun","[0]*pow(x-[1],[2])",0.1,20.);
fun->SetParameters(6000.,-4.,-5.);
 tg->Fit(fun,"emr","same");
cout << " C = "<< fun->GetParameter(2) << endl;
c2->SaveAs("exercise3.pdf");
```

approximate!
you can do better!!

I hope you liked this course.

Remember to fill in the course questionnaire and to be honest...

Any direct feedback is welcomed – even after the end of the course and/or examination. Feel free to contact me!


CIAO CIAO
CIAO CIAO CIAO
CIAO
CIAO
CIAO
CIAO
CIAO
CIAO
CARILLI