

Training & Tutoring In All Things Computing for Science



D. Menasce (INFN)

The Working Group

A Working Group has been set up in HSF to coordinate activities related to Training & Tutoring (the T&TWG).

Aim of this WG is twofold:

- Provide a **forum** and a **reference** point for anything connected with scientific computing: people should find there documentation, examples, lessons suggestions and comments about issues related to computing in all their HEP (but not only HEP) incarnations.
- **Even more important:** provide suitable training to young students to *bridge the existing gap between University and Research for what concerns computing skills* (basically only web-based training due to time/budget constraints: excellent schools already exist)

The basic principles

Whatever the goal, basic to any activity in this field is the availability of a suitable web-based platform to publish links, documentation and whatever useful for tutoring & training.

We didn't want to be stuck to a specific platform from the beginning, but we soon realized we needed a quick jump-start (few volunteers means a very diluted activity with scarce manpower):

- Adopted WikiFM (no definitive commitment right now, only experimental R&D to investigate and assess possibilities). Why:
 - ✓ It already exists, it is maintained, hosted by KDE.org and is actively developed
 - ✓ Gentlemen's agreement: the developers help us with maintenance and development trading for broader (international) visibility. All young volunteers.
 - ✓ If the project turns out successful, we could make more formal agreements (so far all is just experimental and best-effort).
 - ✓ Can host links to external, already existing, material but also host our own home-brewed specific lessons (focused on scientific computing specifically, in whatever format, PPT, Word, PDF, YouTube...)

Been there, done that

Activity has just barely begun, but a few things have been accomplished:

- Held a first T&TWG meeting: a handful of people participated.
 - ✓ Encouraging, but the more the better (we need evangelism...)
 - ✓ Discussed what we would like to accomplish in the **short** term (something to show around, create visibility and attract more people), what will be needed in a **longer** term and finally *tentative* milestones.
- Independently from the WG, in Milano we did some prior R&D, using wikiFM, to explore the possibilities offered by that platform
 - ✓ Based on wikimedia (allows easy editing of material by anyone)
 - ✓ Specifically tailored for training and tutoring (originally in Italian and for Physics only, later on extended to other disciplines and lately to HSF)
 - ✓ Used so far mainly by Italian University students
 - ✓ Equipped with interesting features aimed at answering specific needs of students (eg, creation of customized pdf books with available material)

The wikiFM platform

<http://www.wikifm.org/>



WikiFM

Italiano

«Il sapere si accresce solo se condiviso»

English

«Knowledge only grows if shared»

Français

«Le savoir grandit seulement s'il est partagé»

Deutsch

«Nur wenn Wissen geteilt wird kann es wachsen»

Español

«El conocimiento solo crece cuando es compartido»

This is the original content of the site (basically aimed at University students)

This is where HSF training material will end up.



Layout



WikiFM

Page

Discussion

View

Edit

History



Main Page

WARNING: This page, like the english subsite, is very novel and still under heavy construction - please see [the Italian homepage](#) for a more complete website in the meantime (many novel features work there but not yet here)

Navigation

- Main page
- Recent changes
- Random page
- Help

Tools

- What links here
- Related changes
- Upload file
- Special pages
- Permanent link
- Page information

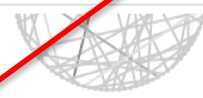
The Departments



Bio-sciences



Chemistry



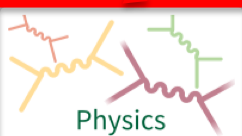
Computer Science



Engineering



Mathematics



Physics

Experimental HSF material is under the heading Physics (this could change)

The subject tree



Tools

- What links here
- Related changes
- Upload file
- Special pages
- Permanent link
- Page information

Print/export

- Create a book
- Download as PDF
- Download as Plain text
- Printable version

Personal tools

-  English
- Menasce
-  7
- Talk
- Preferences
- Watchlist
- New messages (none)
- Contributions
- Log out

Basic Courses

- [Classical Mechanics](#)
- [Electromagnetism](#)
- [Mechanics](#)
- [Quantum Mechanics](#)

Related Courses

- [Basic Calculus](#)

Astrophysics

- [Cosmology](#)

Biophysics

Particle Physics

- [Elementary Particle Physics](#)

Theoretical Physics

- [Statistical Mechanics](#)

HEP Software Foundation Courses

Disclaimer

This section is currently under construction. As such, teaching material can appear and disappear without further advance notice. Thanks for your understanding

- [Philosophy of this site](#)

University courses (old, pre-existing material for Physics students)

HSF specific material (new)

The subject tree

HEP Software Foundation Courses

Disclaimer ×

This section is currently under construction. As such, teaching material can appear and disappear without further advance notice. Thanks for your understanding

- [Philosophy of this site](#)

Basic training

- [Elements of Operating Systems](#)
- [Compiled languages](#)
- [Scripting languages](#)
- [Statistical tools](#)
- [Analysis tools](#)
- [Advanced mathematical techniques](#)
- [Simulation \(Geant IV\)](#)
- [Multithreading techniques](#)
- [Design Patterns](#)
- [Graphical User Interfaces techniques](#)
- [JavaScript and WEB techniques](#)
- [Data Acquisition Techniques](#)

Training by examples

- [The Standard Template Library](#)

Crucial is an early establishment of a navigation schema, to **first** allow contributors to publish their training material in the appropriate/relevant section and **second** to help students identifying the material they need.

This still needs ample discussion and improvements

For the time being we consider the addition of links to existing material a trivial exercise and we ignore this subject here (there are issues to be considered for this...)



WikiFM

Navigation

- [Main page](#)
- [Recent changes](#)
- [Random page](#)
- [Help](#)

Tools

- [What links here](#)
- [Related changes](#)
- [Upload file](#)
- [Special pages](#)
- [Permanent link](#)
- [Page information](#)

Print/export

- [Create a book](#)
- [Download as PDF](#)
- [Download as Plain text](#)

Page

Discussion

View

Edit

History

Unchecked

The Standard Template Library

(Author: [D. Menasce](#))

The Standard Template Library, or STL, is a C++ library that has been developed to provide users with useful tools to manipulate data regardless of a specific data type.

The whole concept of the STL revolves around the concept of data-type abstraction, templating, a concept related to the category of generic programming.

This introduction has been conceived to introduce you to the intricacies of STL starting from trivial examples to more and more complex variations in order to illustrate the benefits gained by adoption of STL components instead of ad-hoc, private code implementations, a way to avoid the reinvent-the-wheel syndrome. As we will see and discuss, the STL provides an fundamental added benefit, namely a highly efficient implementation.

- [Why STL](#)
- [Overview of STL](#)
- [Containers](#)
- [Iterators](#)
- [Algorithms](#)
- [Functors](#)

As an interesting exercise, attempted to provide an introductory lesson on STD (a C++ example), giving students the possibility of trying simple exercises in real time (compilation and run)

(Author: D. Menasce)

Navigation

- Main page
- Recent changes
- Random page
- Help

Tools

- What links here
- Related changes
- Upload file
- Special pages
- Permanent link
- Page information

Print/export

- Create a book
- Download as PDF
- Download as Plain text
- Printable version

Personal tools

-  English
- Menasce
-  7
- Talk

In this chapter I would like to illustrate what was the original idea that led [Alexander Stepanov](#) to the development of the STL in the early '80.

Consider the following example, a C++ array of integers (`int`) for which we would like to compute the sum (to show line numbers press the "Run this code" button, which also displays the result in real time: note that the "Share" button allows you to interactively modify the code and check your alternative implementations):

Run this code

```
// g++ -o ex1 ex1.cc && ./ex1
#include <iostream>

using namespace std;

int myArray[] = {16, 2, 77, 40, 12071};

int n, result=0;

//=====
int main ()
{
    for ( n=0 ; n<5 ; ++n )
    {
        result += myArray[n];
    }
    cout << "Sum:" << result << endl ;
    return 0;
}
```

Explanation text

Interactive
compile & run
button!

Code snippets with
practical examples

- What links here
- Related changes
- Upload file
- Special pages
- Permanent link
- Page information

Print/export

- Create a book
- Download as PDF
- Download as Plain text
- Printable version

Personal tools

-  English
- Menasce
-  7
- Talk
- Preferences

Run

Share

Exit

GCC 4.9 (C++14)

GCC 4.6

GCC 4.7

GCC 4.7 (C++11)

GCC 4.8

GCC 4.8 (C++11)

GCC 4.9

GCC 4.9 (C++11)

GCC 4.9 (C++14)

clang 3.5

clang 3.5 (C++11)

clang 3.5 (C++14)

```

1 // g++ -o ex1 ex1.
2
3 #include <iostream>
4
5 using namespace std;
6
7 int myArray[] = {1
8
9 int n, result=0;
10
11 //=====
12 int main ()
13 {
14 for ( n=0; n<5; ++n)

```

Output:

Building and running...




- Related changes
- Upload file
- Special pages
- Permanent link
- Page information

Print/export

- Create a book
- Download as PDF
- Download as Plain text
- Printable version

Personal tools

-  English
- Create account
- Log in

Run **Share** Exit GCC 4.9 (C++14)

```
1 // g++ -o ex1 ex1.cc && ./ex1
2
3 #include <iostream>
4
5 using namespace std;
6
7 int myArray[] = {16, 2, 77, 40, 12071};
8
9 int n, result=0;
10
11 //=====
12 int main ()
13 {
14     for ( n=0 ; n<5 ; ++n )
```

Output:

Sum: 12206



```
1 // g++ -o ex1 ex1.cc && ./ex1
2
3 #include <iostream>
4
5 using namespace std;
6
7 int myArray[] = {16, 2, 77, 40, 12071};
8
9 int n, result=0;
10 doble a ;
11
12 //=====
13 int main ()
14 {
15     for ( n=0 ; n<5 ; ++n )
16     {
17         result += myArray[n];
18     }
```

```
main.cpp:10:1: error: 'doble' does not name a type
  doble a ;
  ^
```

```
g++-4.9 -std=c++14 -O2 -Wall -Wextra -pedantic -pthread
-pedantic-errors main.cpp -lm && ./a.out
```

Compile, link and run...

Share!



```
1 // g++ -o ex1 ex1.cc && ./ex1
2
3 #include <iostream>
4
5 using namespace std;
6
7 int myArray[] = {16, 2, 77, 40, 12071};
8
9 int n, result=0;
10 double a ;
11
12 //=====
13 int main ()
14 {
15     for ( n=0 ; n<5 ; ++n )
16     {
17         result += myArray[n];
18     }
```

Sum: 12206

```
g++-4.9 -std=c++14 -O2 -Wall -Wextra -pedantic -pthread
-pedantic-errors main.cpp -lm && ./a.out
```

Compile, link and run...

Share!



This approach can have far fetching implications:

- The obvious one is a boost to student's learning speed (shallow learning curve)
- The second is a door open to things like online documentation
 - ✓ At Fermilab there is interest in examining the possibilities offered by this tools as a way to ease the learning curve of projects like Art and ArtDag (<https://cdcv.s.fnal.gov/redmine/projects/artdaq-demo/wiki>) thus targeting for training not only students but researchers as well.
 - ✓ This is a very wide and ambitious project, we will not place a high priority for this (unless adequate volunteer man-power shows up) but still this example shows the immense possibilities for on-line training and tutoring
- A third opportunity comes from a possible collaboration with the ROOT team for the development of remote (web-based) interactive ROOT session. Again, something of a somewhat lower priority for the T&TWG for now (many analogous possibilities loom on the horizon).

Conclusions

- The T&TWG has begun both periodic meetings and concrete activities
- We identified a possible platform to deploy our solutions (WikiFM)
 - ✓ Both platform and solutions are still experimental, need to asses and define a clear final target
- Begun populating WikiFM with tentative material
- Explored the possibility of interactively compile & run simple snippets of code (this still requires significant investment)
- We seek help from anyone interested
 - ✓ First to define the portfolio of what we want to publish (both external links and our own home-brewed material). Build the portfolio tree.
 - ✓ Second to contribute with real training material (editing and adding stuff to the site is trivial being based on wikimedia).