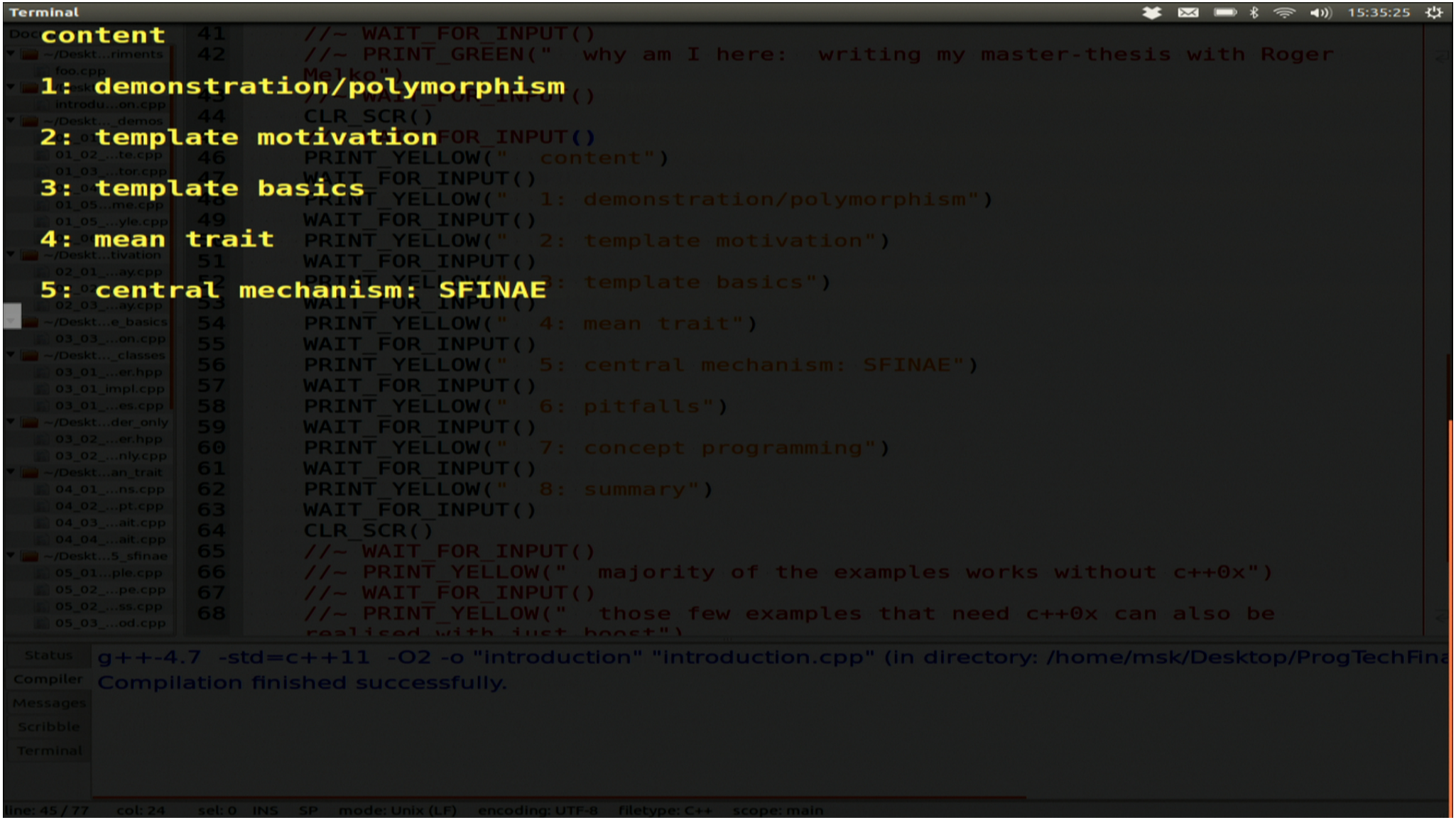
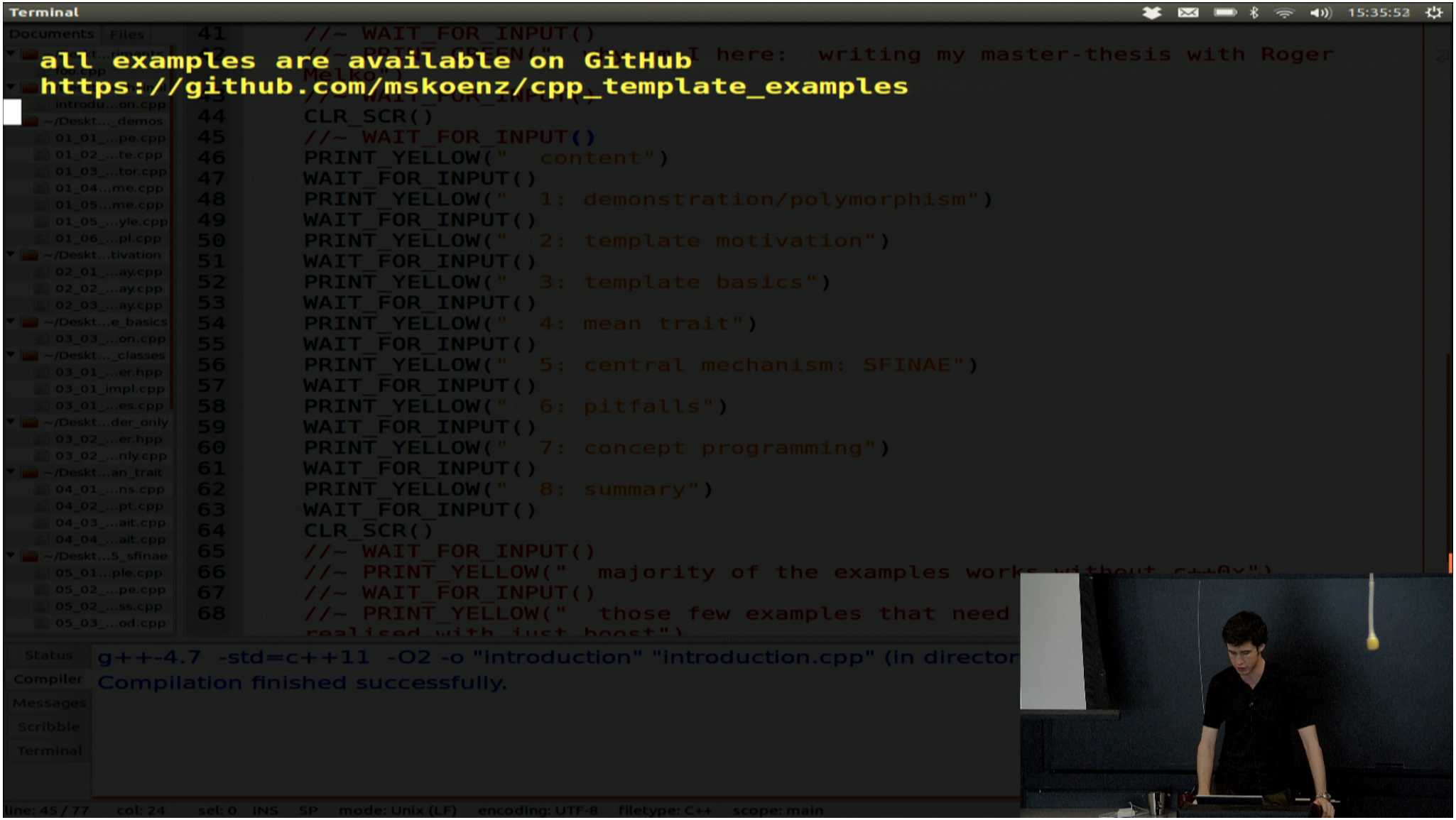
Title: C++ template magic - Part 3

Date: Jul 05, 2013  03:30 PM

URL: http://pirsa.org/13070028

Abstract: A general introduction to c++ templates. As motivation the possibilities will be demonstrated in short examples in order to show the might of these programming techniques. After covering the fundamentals of templates the following topics will be discussed: polymorphism, traits, template specialisation, SFINAE and in case there is time left template meta programming.


content
1: demonstration/polymorphism
2: template motivation
3: template basics
4: mean trait
5: central mechanism: SFINAE
Terminal
15:35:25
//~ WAIT_FOR_INPUT()
//~ PRINT_GREEN("  why am I here:  writing my master-thesis with Roger Melko")
CLR_SCR()
PRINT_YELLOW("  content")
PRINT_YELLOW("  1: demonstration/polymorphism")
PRINT_YELLOW("  2: template motivation")
PRINT_YELLOW("  3: template basics")
PRINT_YELLOW("  4: mean trait")
PRINT_YELLOW("  5: central mechanism: SFINAE")
PRINT_YELLOW("  6: pitfalls")
PRINT_YELLOW("  7: concept programming")
PRINT_YELLOW("  8: summary")
//~ PRINT_YELLOW("  majority of the examples works without c++0x")
//~ PRINT_YELLOW("  those few examples that need c++0x can also be
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "introduction" "introduction.cpp" (in directory: /home/msk/Desktop/ProgTechFina
Compilation finished successfully.


Terminal
15:35:53
all examples are available on GitHub
https://github.com/mskoenz/cpp_template_examples
41 //~ WAIT_FOR_INPUT()
44 CLR_SCR()
45 //~ WAIT_FOR_INPUT()
46 PRINT_YELLOW(" content")
47 WAIT_FOR_INPUT()
48 PRINT_YELLOW(" 1: demonstration/polymorphism")
49 WAIT_FOR_INPUT()
50 PRINT_YELLOW(" 2: template motivation")
51 WAIT_FOR_INPUT()
52 PRINT_YELLOW(" 3: template basics")
53 WAIT_FOR_INPUT()
54 PRINT_YELLOW(" 4: mean trait")
55 WAIT_FOR_INPUT()
56 PRINT_YELLOW(" 5: central mechanism: SFINAE")
57 WAIT_FOR_INPUT()
58 PRINT_YELLOW(" 6: pitfalls")
59 WAIT_FOR_INPUT()
60 PRINT_YELLOW(" 7: concept programming")
61 WAIT_FOR_INPUT()
62 PRINT_YELLOW(" 8: summary")
63 WAIT_FOR_INPUT()
64 CLR_SCR()
65 //~ WAIT_FOR_INPUT()
66 //~ PRINT_YELLOW(" majority of the examples works
67 //~ WAIT_FOR_INPUT()
68 //~ PRINT_YELLOW(" those few examples that need
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "introduction" "introduction.cpp" (in director
Compilation finished successfully.

09_01_handwritten.cpp - /home/msk/Desktop/ProgTechFinal/09_proof - Geany

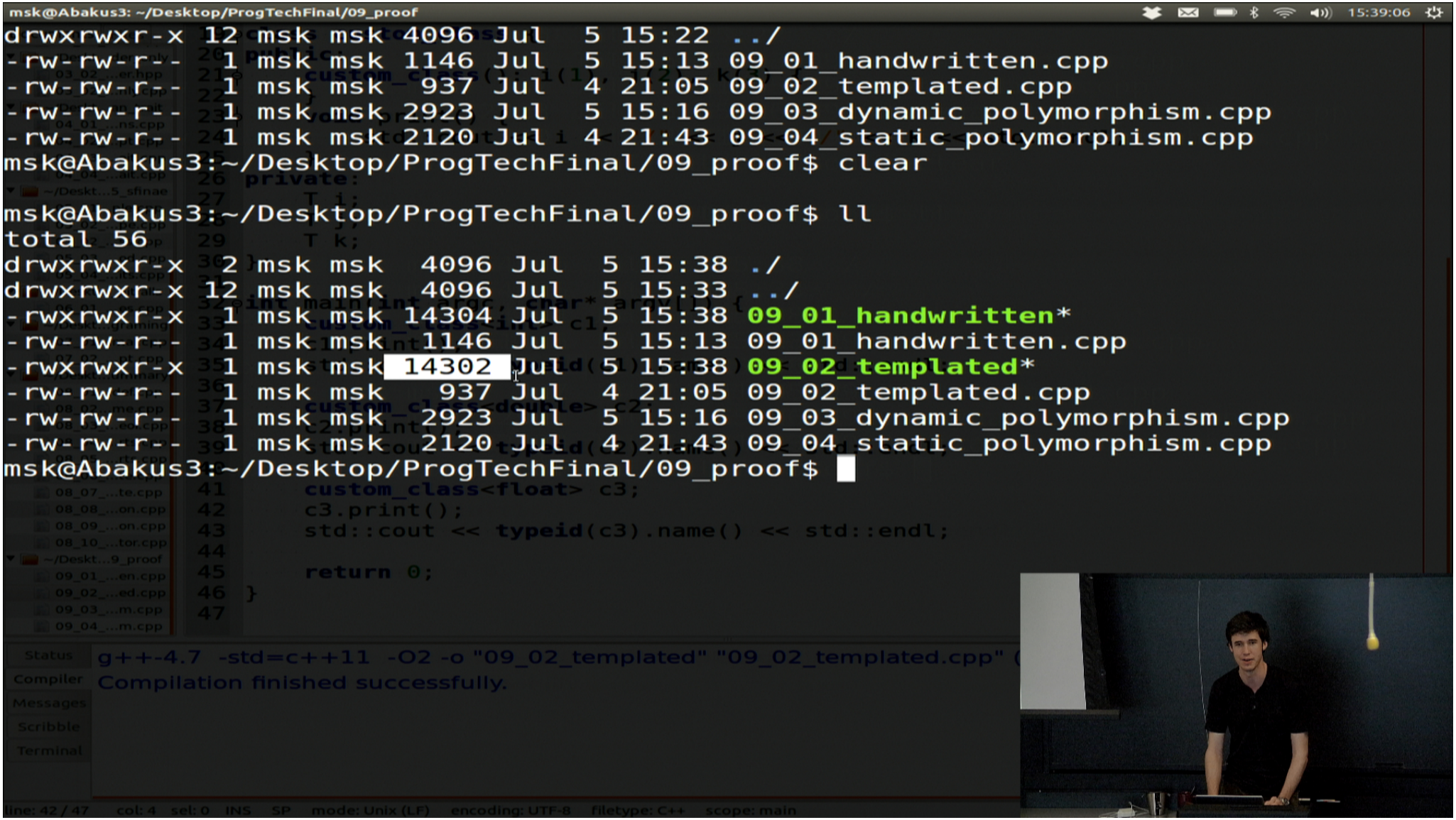
```
 4
 5  #include <iostream>
 6  #include <typeinfo>
 7
 8  class custom_classIiE {
 9  public:
10      custom_classIiE(): i(1), j(2), k(3) {
11      }
12      void print() {
13          std::cout << i << "/" << j << "/" << k << std::endl;
14      }
15  private:
16      int i;
17      int j;
18      int k;
19  };
20
21  class custom_classIdE {
22  public:
23      custom_classIdE(): i(1), j(2), k(3) {
24      }
25      void print() {
26          std::cout << i << "/" << j << "/" << k << std::endl;
27      }
28  private:
29      double i;
30      double j;
31      double k;
32  };
```

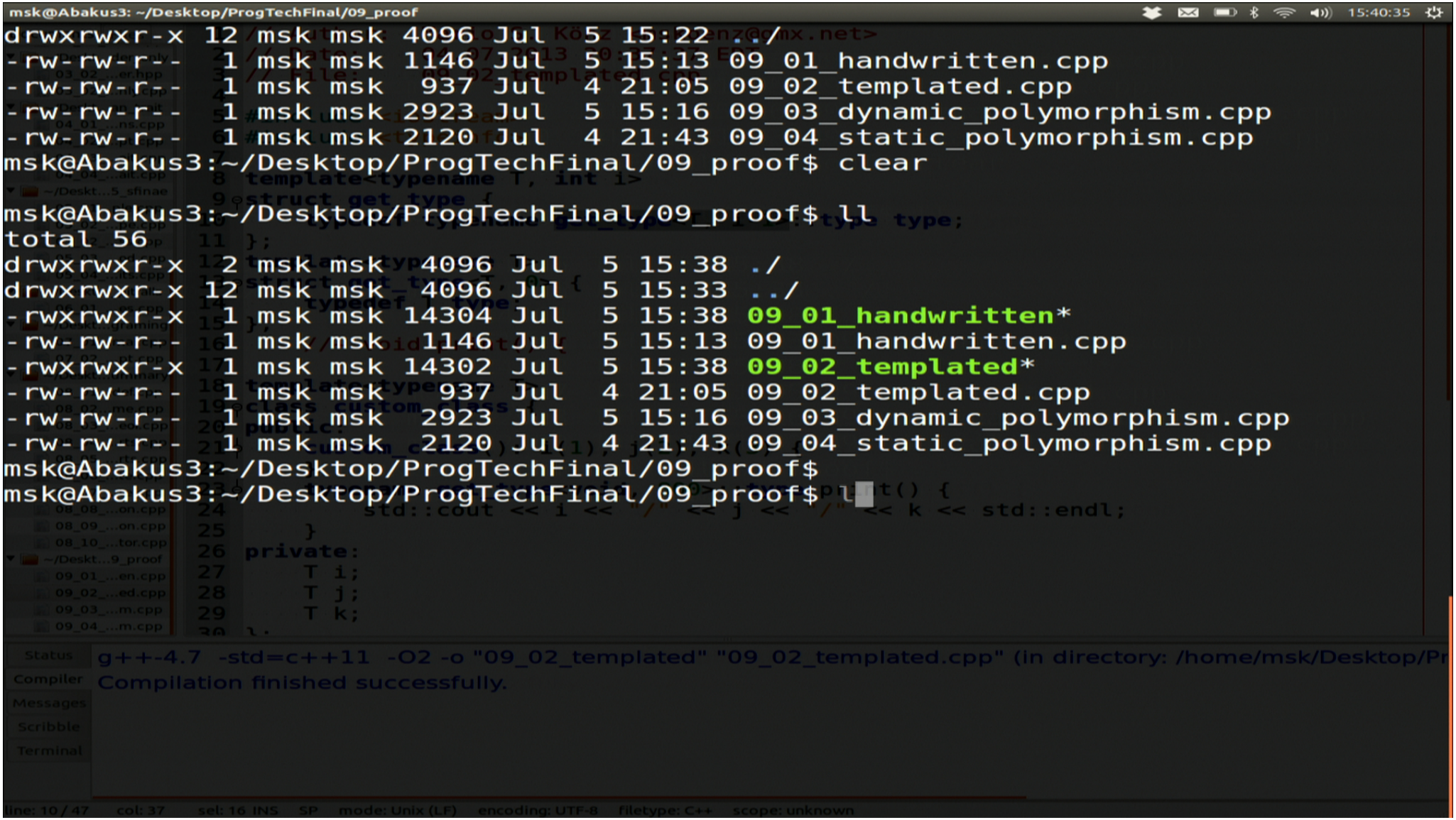
Status
Compiler
Messages
Scribble
Terminal

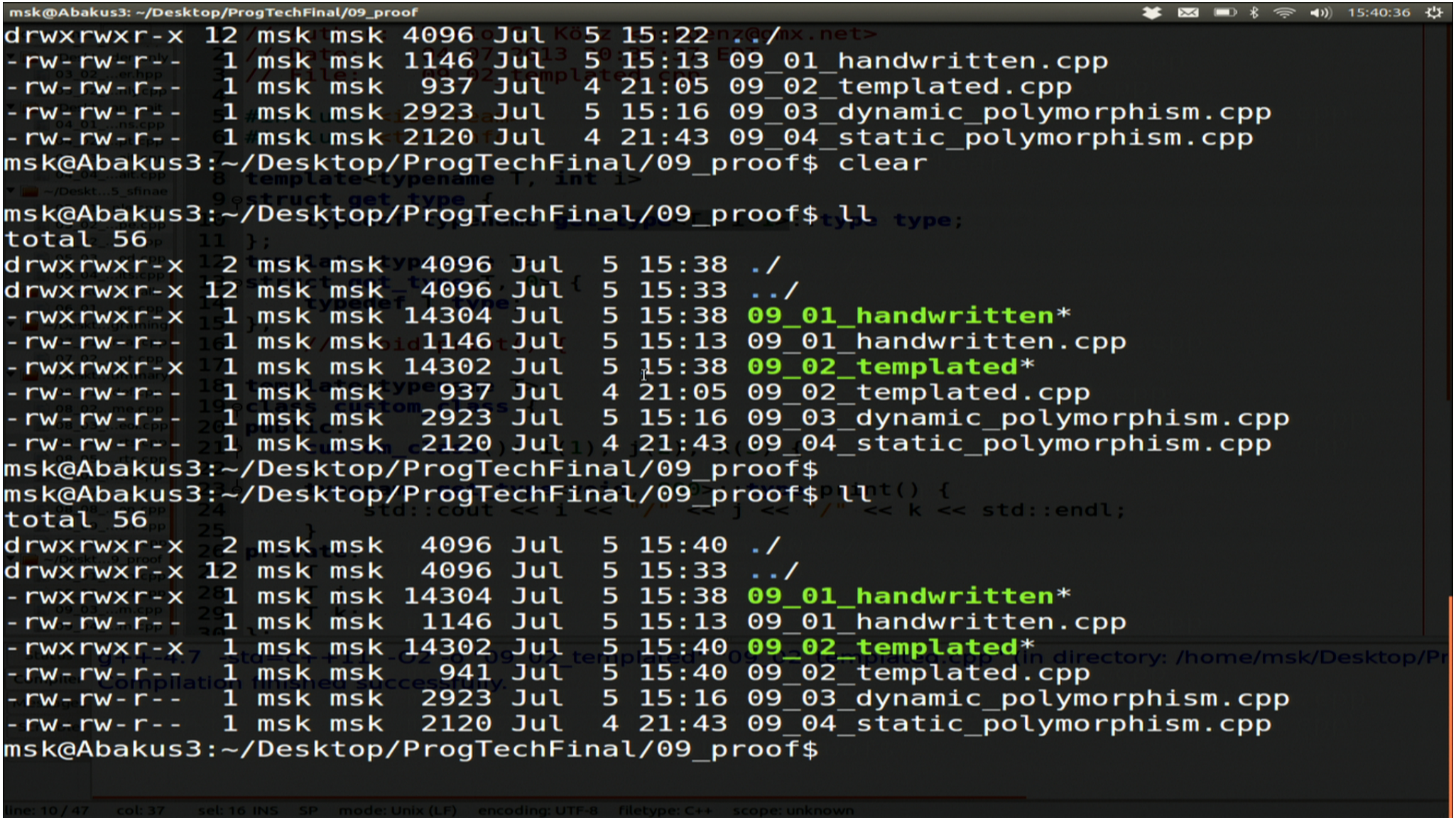
g++-4.7 -std=c++11 -O2 -o "foo" "foo.cpp" (in directory: /home/msk/Desk

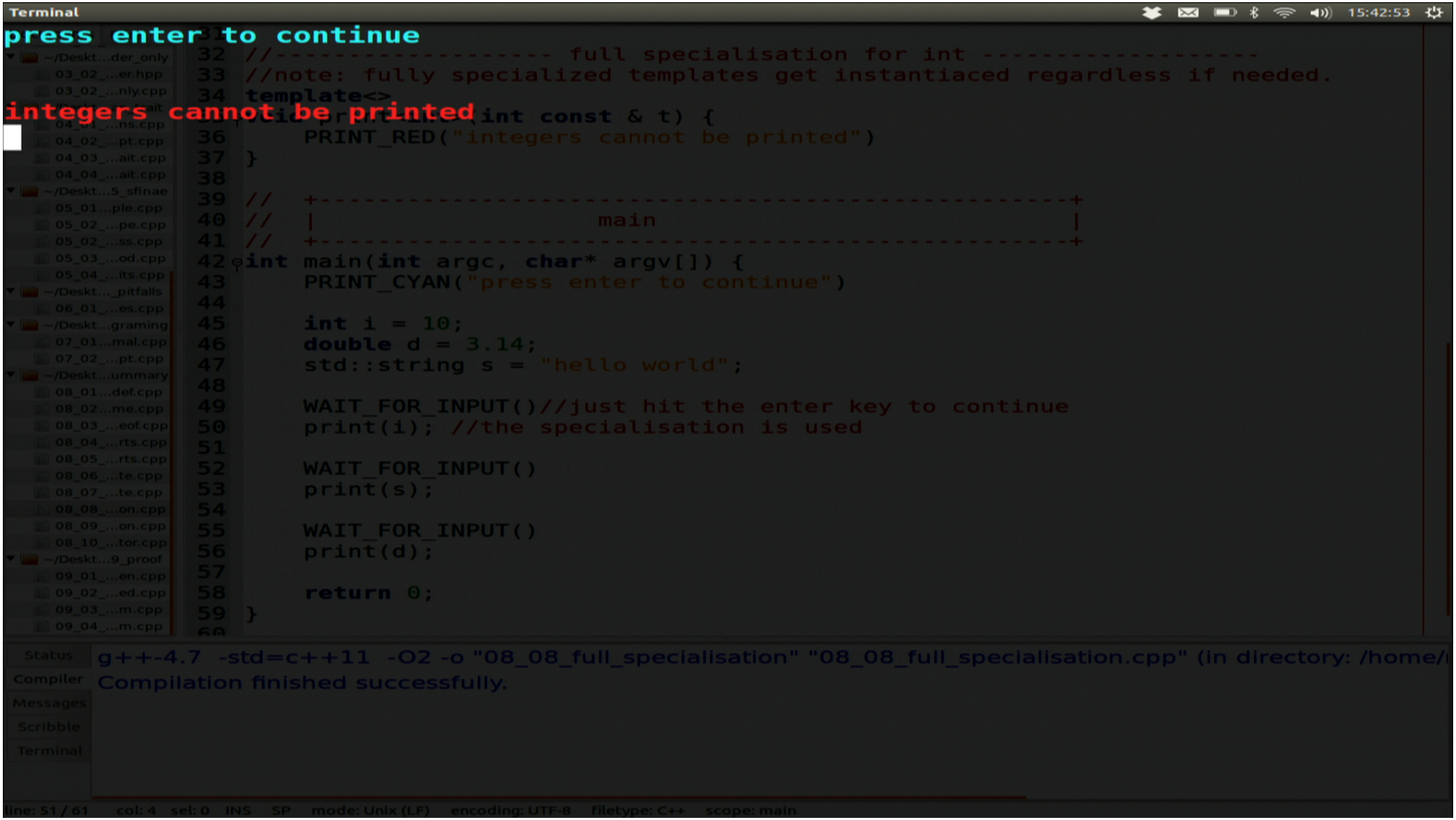
Compilation finished successfully.

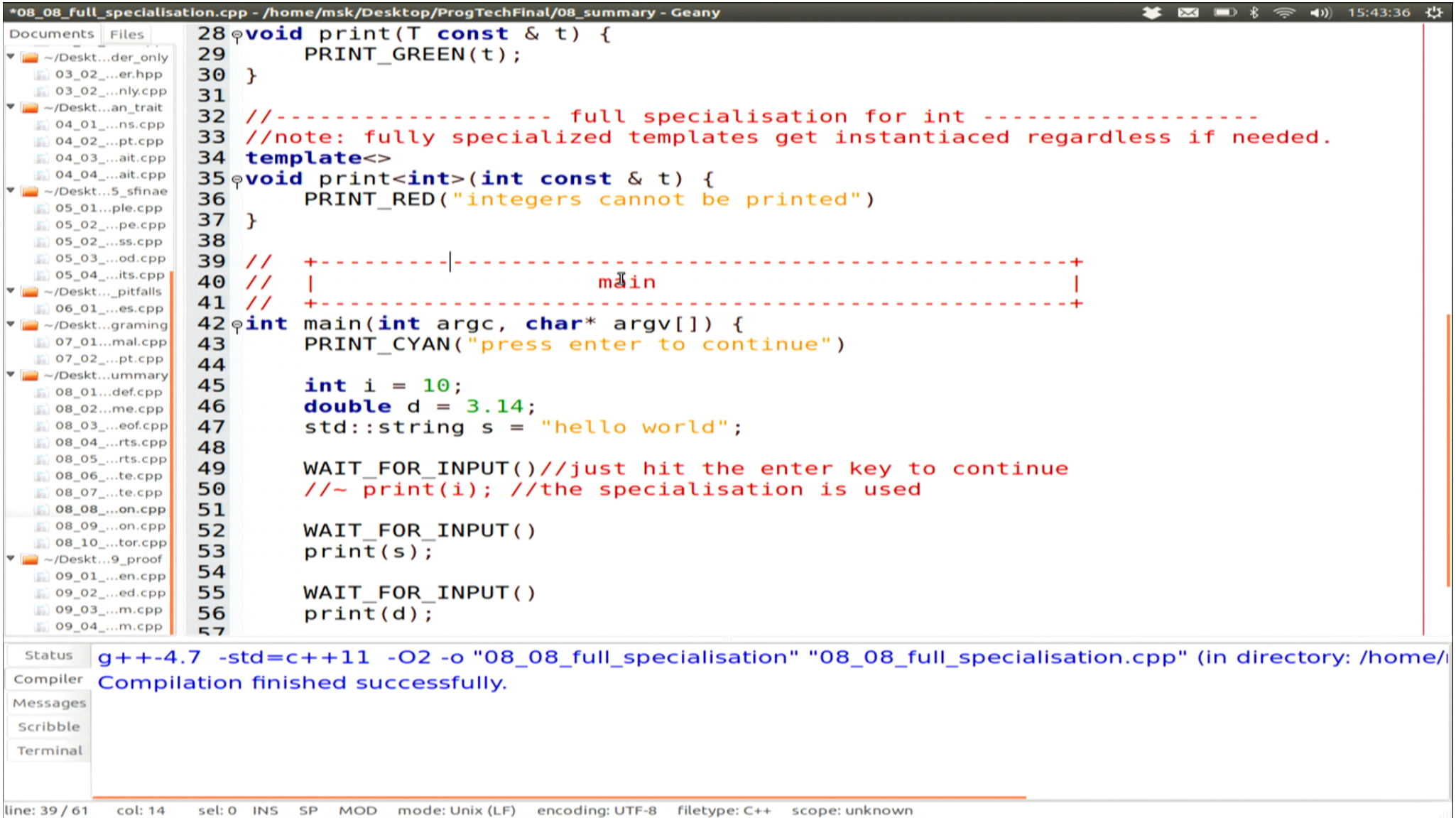
line: 18 / 63 col: 2 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown


msk@Abakus3: ~/Desktop/ProgTechFinal/09_proof
15:39:06
drwxrwxr-x 12 msk msk 4096 Jul 5 15:22 ../
-rw-rw-r-- 1 msk msk 1146 Jul 5 15:13 09_01_handwritten.cpp
-rw-rw-r-- 1 msk msk 937 Jul 4 21:05 09_02_templated.cpp
-rw-rw-r-- 1 msk msk 2923 Jul 5 15:16 09_03_dynamic_polymorphism.cpp
-rw-rw-r-- 1 msk msk 2120 Jul 4 21:43 09_04_static_polymorphism.cpp
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$ clear
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$ ll
total 56
drwxrwxr-x 2 msk msk 4096 Jul 5 15:38 ./
drwxrwxr-x 12 msk msk 4096 Jul 5 15:33 ../
-rwxrwxr-x 1 msk msk 14304 Jul 5 15:38 09_01_handwritten*
-rw-rw-r-- 1 msk msk 1146 Jul 5 15:13 09_01_handwritten.cpp
-rwxrwxr-x 1 msk msk 14302 Jul 5 15:38 09_02_templated*
-rw-rw-r-- 1 msk msk 937 Jul 4 21:05 09_02_templated.cpp
-rw-rw-r-- 1 msk msk 2923 Jul 5 15:16 09_03_dynamic_polymorphism.cpp
-rw-rw-r-- 1 msk msk 2120 Jul 4 21:43 09_04_static_polymorphism.cpp
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$
custom_class<float> c3;
c3.print();
std::cout << typeid(c3).name() << std::endl;
return 0;
}
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "09_02_templated" "09_02_templated.cpp"
Compilation finished successfully.


msk@Abakus3: ~/Desktop/ProgTechFinal/09_proof
15:40:35
drwxrwxr-x 12 msk msk 4096 Jul 5 15:22 ./
-rw-rw-r-- 1 msk msk 1146 Jul 5 15:13 09_01_handwritten.cpp
-rw-rw-r-- 1 msk msk 937 Jul 4 21:05 09_02_templated.cpp
-rw-rw-r-- 1 msk msk 2923 Jul 5 15:16 09_03_dynamic_polymorphism.cpp
-rw-rw-r-- 1 msk msk 2120 Jul 4 21:43 09_04_static_polymorphism.cpp
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$ clear
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$ ll
total 56
drwxrwxr-x 2 msk msk 4096 Jul 5 15:38 ./
drwxrwxr-x 12 msk msk 4096 Jul 5 15:33 ../
-rwxrwxr-x 1 msk msk 14304 Jul 5 15:38 09_01_handwritten*
-rw-rw-r-- 1 msk msk 1146 Jul 5 15:13 09_01_handwritten.cpp
-rwxrwxr-x 1 msk msk 14302 Jul 5 15:38 09_02_templated*
-rw-rw-r-- 1 msk msk 937 Jul 4 21:05 09_02_templated.cpp
-rw-rw-r-- 1 msk msk 2923 Jul 5 15:16 09_03_dynamic_polymorphism.cpp
-rw-rw-r-- 1 msk msk 2120 Jul 4 21:43 09_04_static_polymorphism.cpp
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "09_02_templated" "09_02_templated.cpp" (in directory: /home/msk/Desktop/Pr
Compilation finished successfully.


msk@Abakus3: ~/Desktop/ProgTechFinal/09_proof
15:40:36
drwxrwxr-x 12 msk msk 4096 Jul 5 15:22 ./
-rw-rw-r-- 1 msk msk 1146 Jul 5 15:13 09_01_handwritten.cpp
-rw-rw-r-- 1 msk msk 937 Jul 4 21:05 09_02_templated.cpp
-rw-rw-r-- 1 msk msk 2923 Jul 5 15:16 09_03_dynamic_polymorphism.cpp
-rw-rw-r-- 1 msk msk 2120 Jul 4 21:43 09_04_static_polymorphism.cpp
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$ clear
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$ ll
total 56
drwxrwxr-x 2 msk msk 4096 Jul 5 15:38 ./
drwxrwxr-x 12 msk msk 4096 Jul 5 15:33 ../
-rwxrwxr-x 1 msk msk 14304 Jul 5 15:38 09_01_handwritten*
-rw-rw-r-- 1 msk msk 1146 Jul 5 15:13 09_01_handwritten.cpp
-rwxrwxr-x 1 msk msk 14302 Jul 5 15:38 09_02_templated*
-rw-rw-r-- 1 msk msk 937 Jul 4 21:05 09_02_templated.cpp
-rw-rw-r-- 1 msk msk 2923 Jul 5 15:16 09_03_dynamic_polymorphism.cpp
-rw-rw-r-- 1 msk msk 2120 Jul 4 21:43 09_04_static_polymorphism.cpp
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$ ll
total 56
drwxrwxr-x 2 msk msk 4096 Jul 5 15:40 ./
drwxrwxr-x 12 msk msk 4096 Jul 5 15:33 ../
-rwxrwxr-x 1 msk msk 14304 Jul 5 15:38 09_01_handwritten*
-rw-rw-r-- 1 msk msk 1146 Jul 5 15:13 09_01_handwritten.cpp
-rwxrwxr-x 1 msk msk 14302 Jul 5 15:40 09_02_templated*
-rw-rw-r-- 1 msk msk 941 Jul 5 15:40 09_02_templated.cpp
-rw-rw-r-- 1 msk msk 2923 Jul 5 15:16 09_03_dynamic_polymorphism.cpp
-rw-rw-r-- 1 msk msk 2120 Jul 4 21:43 09_04_static_polymorphism.cpp
msk@Abakus3:~/Desktop/ProgTechFinal/09_proof$


Terminal
15:42:53
press enter to continue
integers cannot be printed
32 //------------------- full specialisation for int -------------------
33 //note: fully specialized templates get instantiaced regardless if needed.
34 template<>
35 void print(int const & t) {
36 PRINT_RED("integers cannot be printed")
37 }
38
39 // +-----------------------------------------------------+
40 // | main |
41 // +-----------------------------------------------------+
42 int main(int argc, char* argv[]) {
43 PRINT_CYAN("press enter to continue")
44
45 int i = 10;
46 double d = 3.14;
47 std::string s = "hello world";
48
49 WAIT_FOR_INPUT()//just hit the enter key to continue
50 print(i); //the specialisation is used
51
52 WAIT_FOR_INPUT()
53 print(s);
54
55 WAIT_FOR_INPUT()
56 print(d);
57
58 return 0;
59 }
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "08_08_full_specialisation" "08_08_full_specialisation.cpp" (in directory: /home/
Compilation finished successfully.


*08_08_full_specialisation.cpp - /home/msk/Desktop/ProgTechFinal/08_summary - Geany
15:43:36
Documents
Files
28 void print(T const & t) {
29 PRINT_GREEN(t);
30 }
31
32 //------------------- full specialisation for int -------------------
33 //note: fully specialized templates get instantiaced regardless if needed.
34 template<>
35 void print<int>(int const & t) {
36 PRINT_RED("integers cannot be printed")
37 }
38
39 // +---------+-----------------------------------------------+
40 // | main |
41 // +-----------------------------------------------------------+
42 int main(int argc, char* argv[]) {
43 PRINT_CYAN("press enter to continue")
44
45 int i = 10;
46 double d = 3.14;
47 std::string s = "hello world";
48
49 WAIT_FOR_INPUT()//just hit the enter key to continue
50 //~ print(i); //the specialisation is used
51
52 WAIT_FOR_INPUT()
53 print(s);
54
55 WAIT_FOR_INPUT()
56 print(d);
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "08_08_full_specialisation" "08_08_full_specialisation.cpp" (in directory: /home/
Compilation finished successfully.
line: 39 / 61 col: 14 sel: 0 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown

08_09_partial_...cialisation.cpp - /home/msk/Desktop/ProgTechFinal/08_summary - Geany

Documents Files

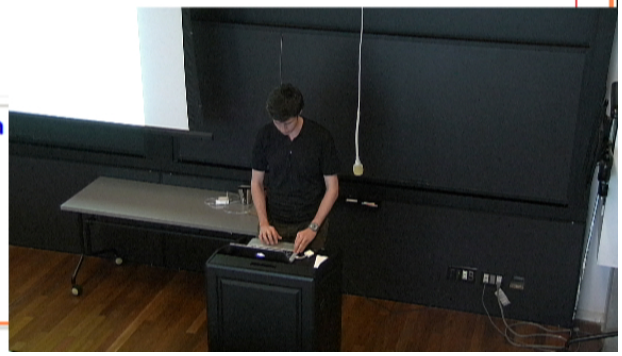
~/Deskt...der_only
03_02_...er.hpp
03_02_...nly.cpp
~/Deskt...an_trait
04_01_...ns.cpp
04_02_...pt.cpp
04_03_...ait.cpp
04_04_...ait.cpp
~/Deskt...5_sfinae
05_01...ple.cpp
05_02_...pe.cpp
05_02_...ss.cpp
05_03_...od.cpp
05_04_...its.cpp
~/Deskt..._pitfalls
06_01_...es.cpp
~/Deskt...graming
07_01...mal.cpp
07_02_...pt.cpp
~/Deskt...ummary
08_01...def.cpp
08_02...me.cpp
08_03_...eof.cpp
08_04_...rts.cpp
08_05_...rts.cpp
08_06_...te.cpp
08_07_...te.cpp
08_08_...on.cpp
08_09_...on.cpp
08_10_...tor.cpp
~/Deskt...9_proof
09_01_...en.cpp
09_02_...ed.cpp
09_03_...m.cpp
09_04_...m.cpp

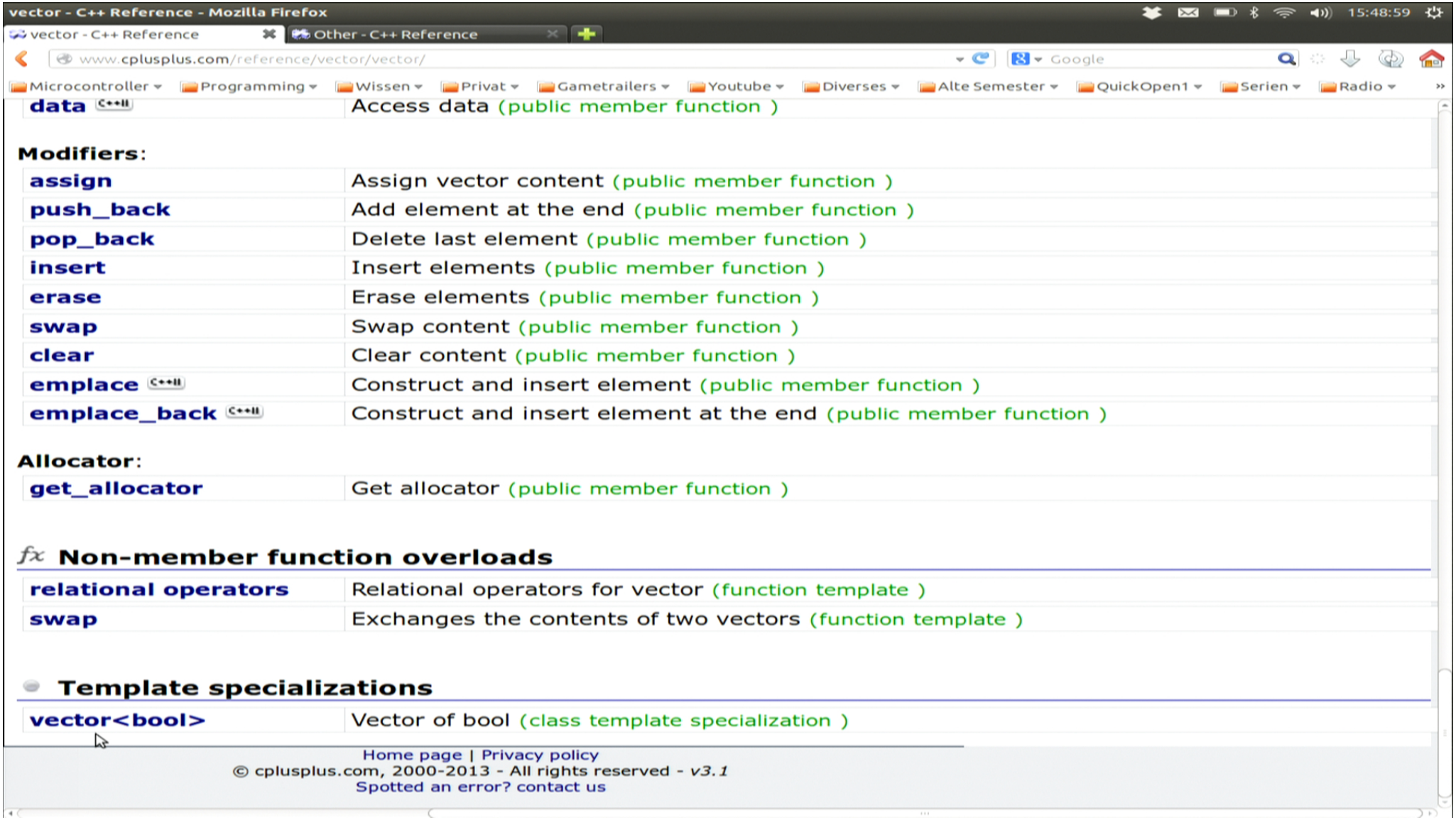
```
35
36 //   +-----------------------------------------------------+
37 //   |                      main                           |
38 //   +-----------------------------------------------------+
39 int main(int argc, char* argv[]) {
40     PRINT_CYAN("press enter to continue")
41
42
43     WAIT_FOR_INPUT()//just hit the enter key to continue
44     //~ std::pair<int, int> p1(1, 2);
45     auto p1 = std::make_pair(1, 2);
46     print_pair(p1); //the specialisation is used
47
48     WAIT_FOR_INPUT()
49     auto p2 = std::make_pair(3.1, 4);
50     print_pair(p2);
51
52     WAIT_FOR_INPUT()
53     auto p3 = std::make_pair(5, 6.1);
54     print_pair(p3); //the specialisation is used
55
56     WAIT_FOR_INPUT()
57     auto p4 = std::make_pair(7.1, 8.1);
58     print_pair(p4);
59
60     return 0;
61 }
62
63
```

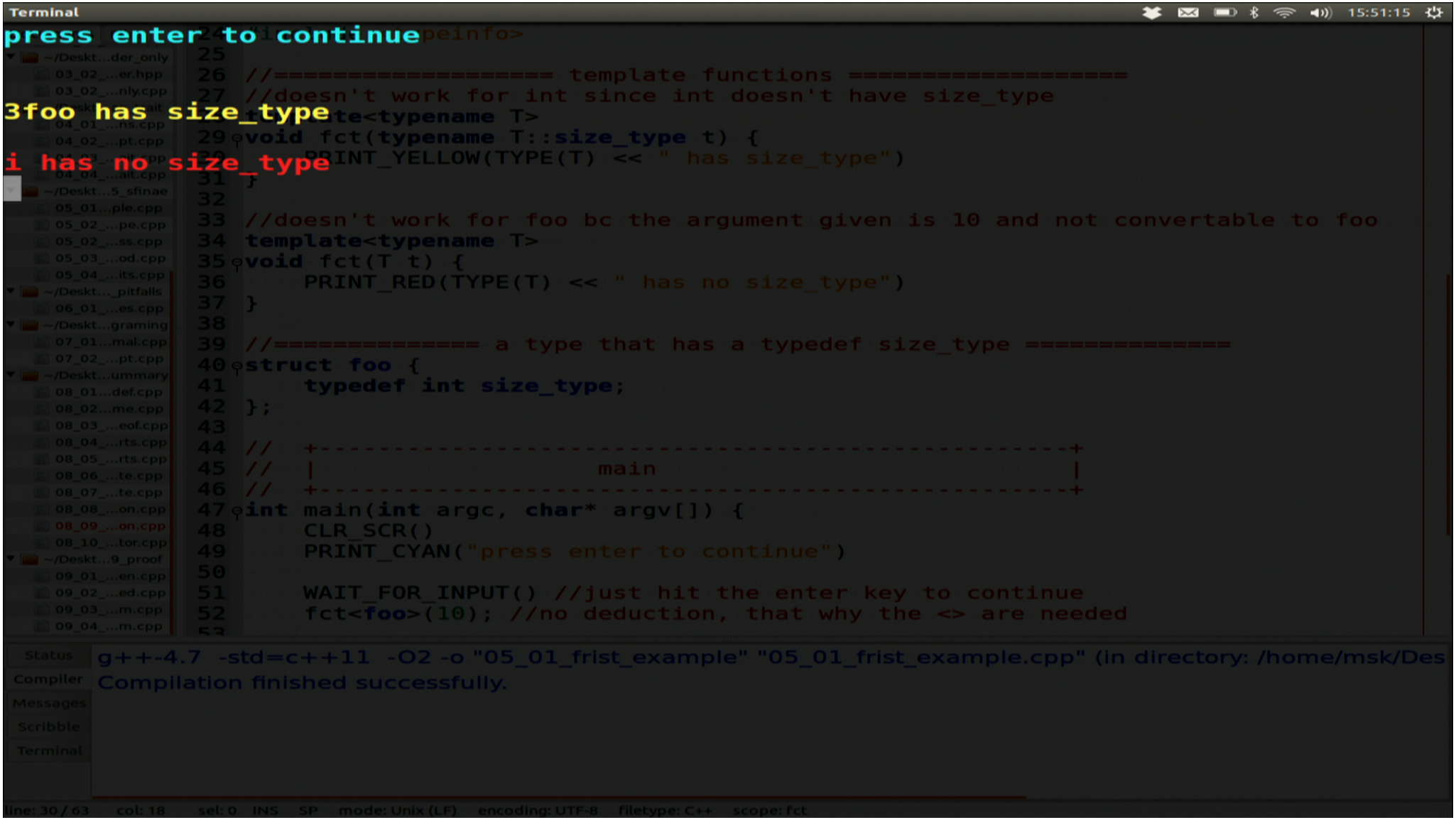
Status
Compiler
Messages
Scribble
Terminal

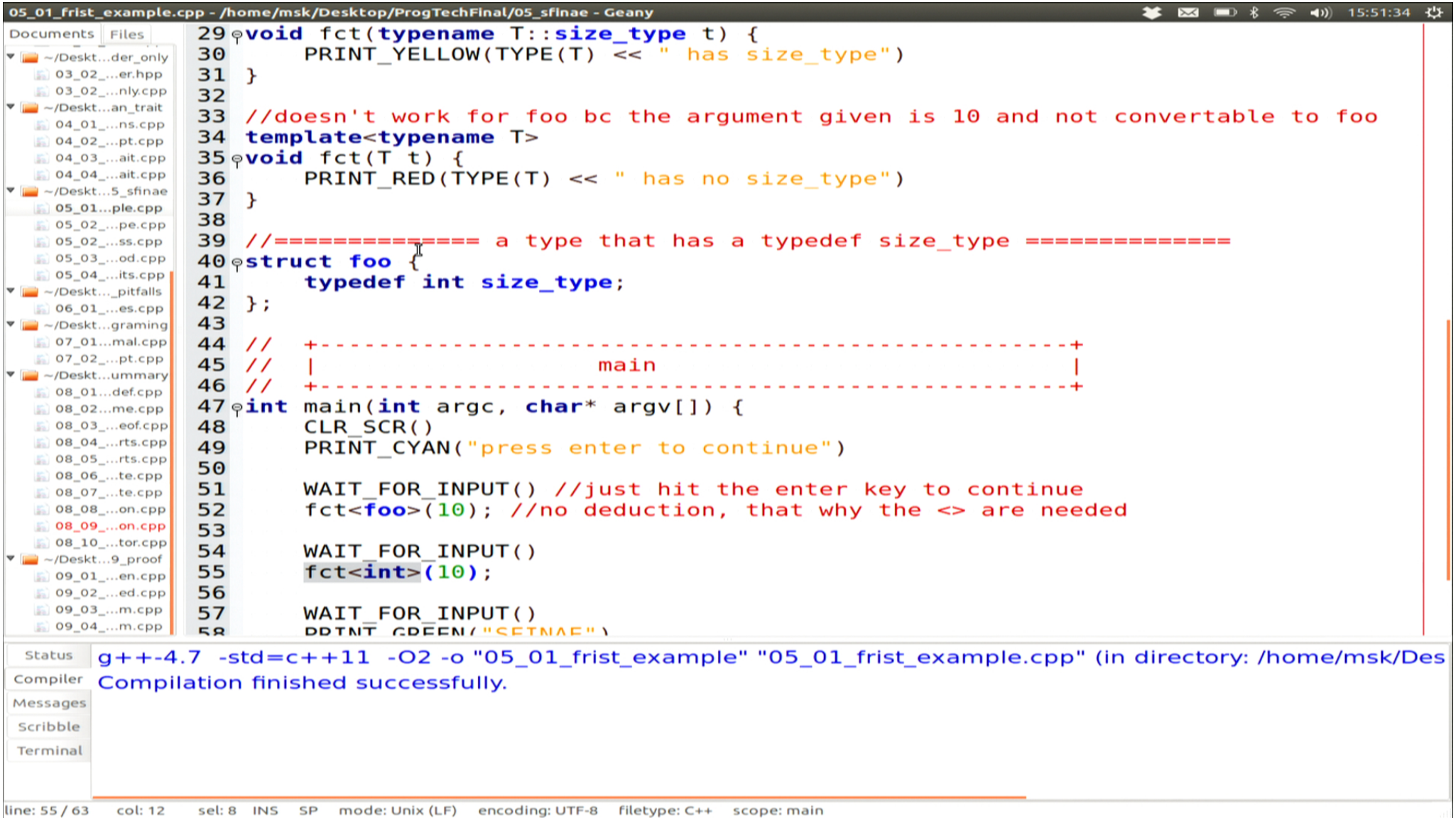
```
g++-4.7 -std=c++11 -O2 -o "08_09_partial_specialisation" "08_09_partia
Compilation finished successfully.
```

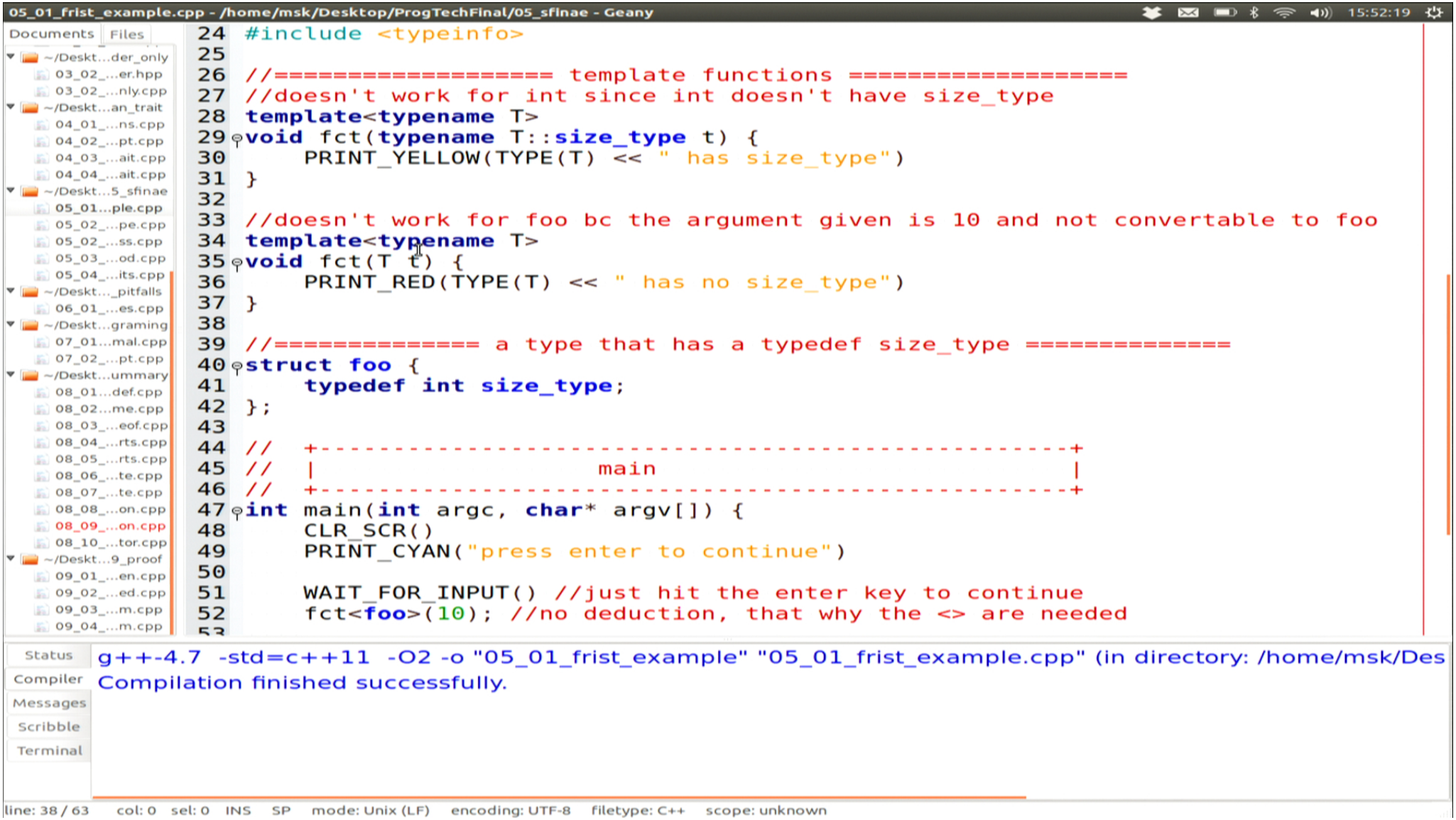
line: 33 / 63  col: 4  sel: 0  INS  SP  mode: Unix (LF)  encoding: UTF-8  filetype: C++  scope: print_pair

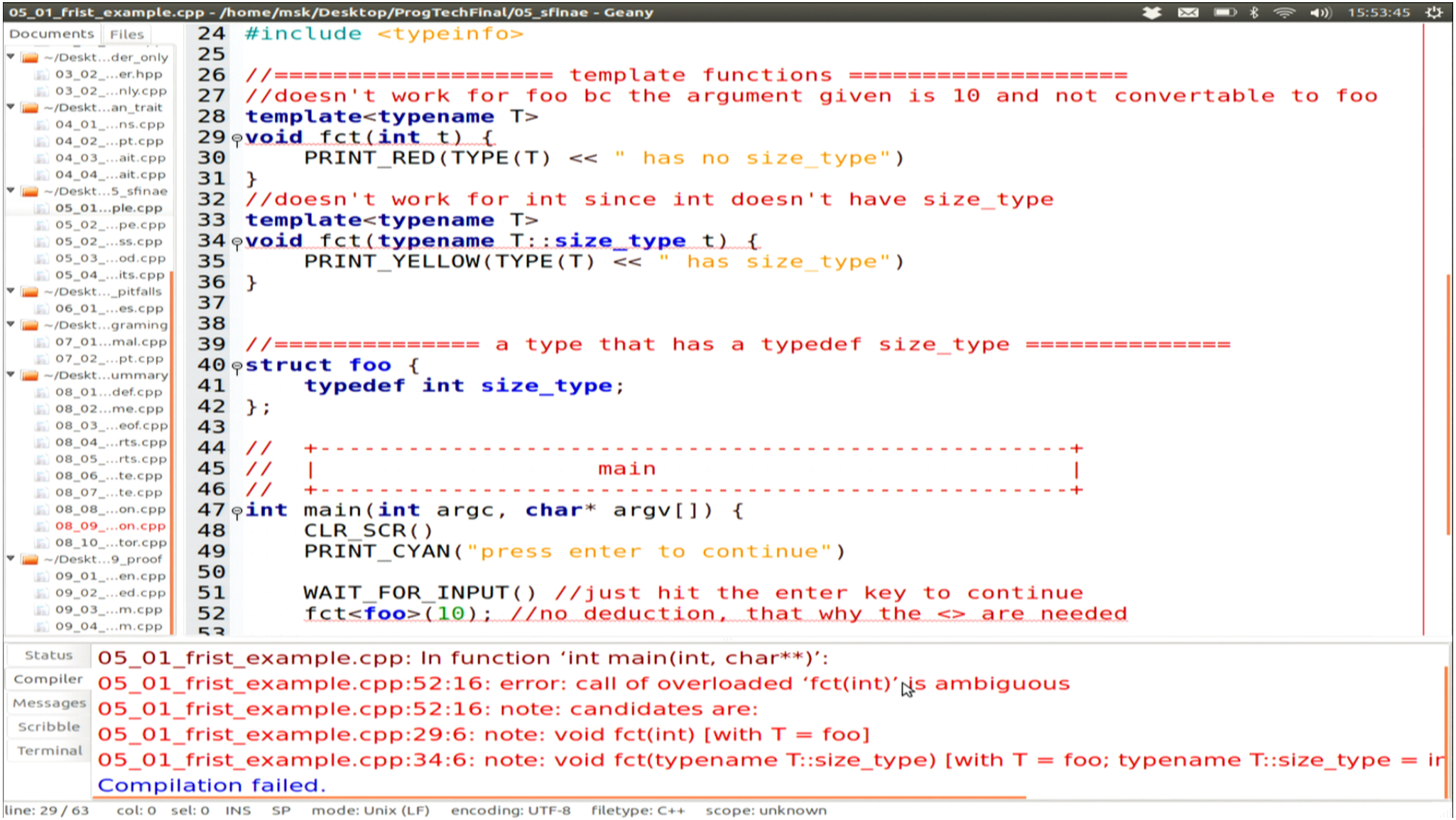


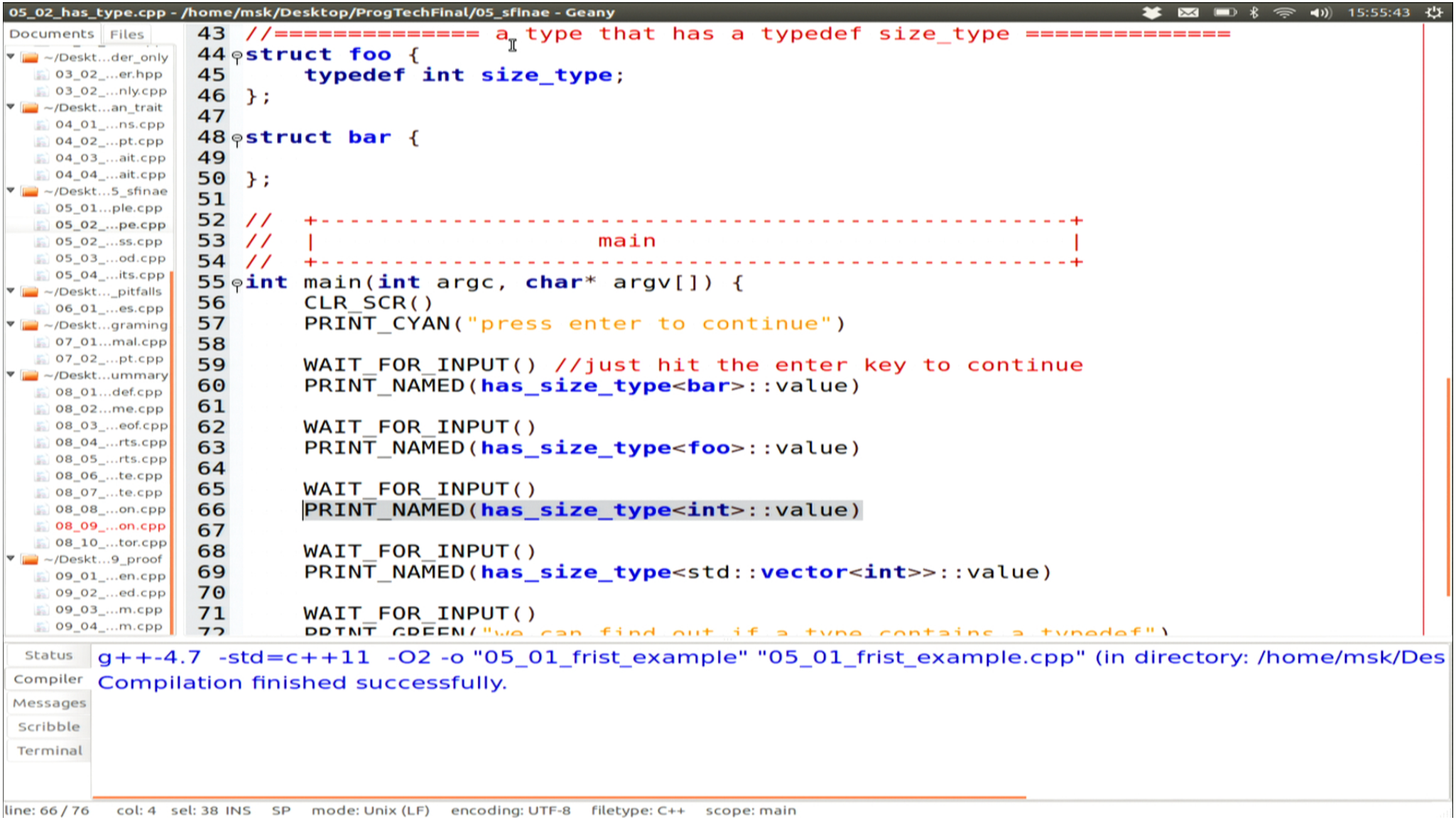

vector - C++ Reference - Mozilla Firefox
15:48:59
vector - C++ Reference
Other - C++ Reference
www.cplusplus.com/reference/vector/vector/
Google
Microcontroller
Programming
Wissen
Privat
Gametrailers
Youtube
Diverses
Alte Semester
QuickOpen1
Serien
Radio
data C++11 Access data (public member function )
Modifiers:
assign Assign vector content (public member function )
push_back Add element at the end (public member function )
pop_back Delete last element (public member function )
insert Insert elements (public member function )
erase Erase elements (public member function )
swap Swap content (public member function )
clear Clear content (public member function )
emplace C++11 Construct and insert element (public member function )
emplace_back C++11 Construct and insert element at the end (public member function )
Allocator:
get_allocator Get allocator (public member function )
Non-member function overloads
relational operators Relational operators for vector (function template )
swap Exchanges the contents of two vectors (function template )
Template specializations
vector<bool> Vector of bool (class template specialization )
Home page | Privacy policy
© cplusplus.com, 2000-2013 - All rights reserved - v3.1
Spotted an error? contact us

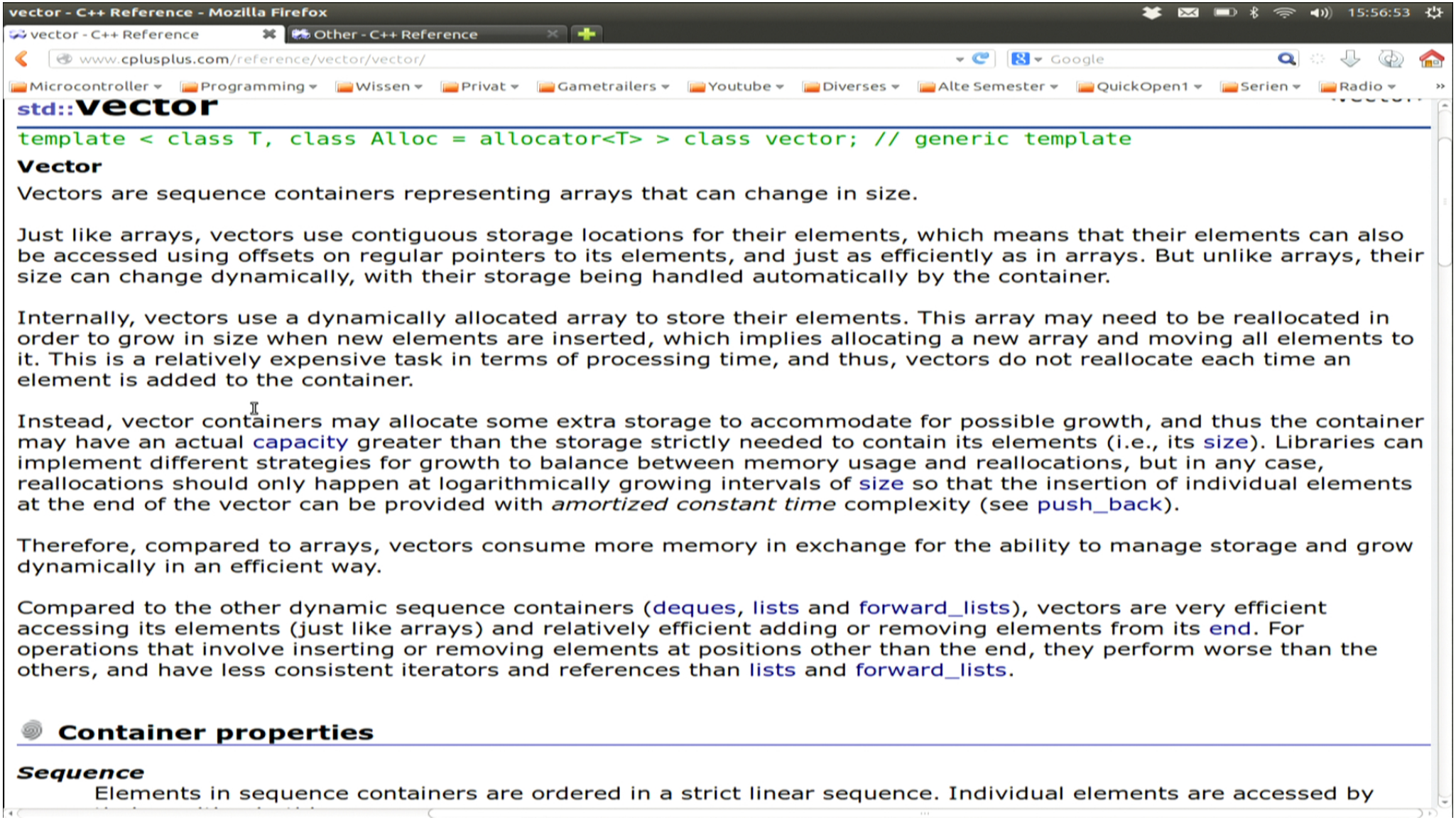

Terminal
15:51:15
press enter to continue
3foo has size_type
i has no size_type
//================== template functions ==================
//doesn't work for int since int doesn't have size_type
template<typename T>
void fct(typename T::size_type t) {
PRINT_YELLOW(TYPE(T) << " has size_type")
}
//doesn't work for foo bc the argument given is 10 and not convertable to foo
template<typename T>
void fct(T t) {
PRINT_RED(TYPE(T) << " has no size_type")
}
//============== a type that has a typedef size_type ==============
struct foo {
typedef int size_type;
};
//+----------------------------------------------------+
//| main |
//+----------------------------------------------------+
int main(int argc, char* argv[]) {
CLR_SCR()
PRINT_CYAN("press enter to continue")
WAIT_FOR_INPUT() //just hit the enter key to continue
fct<foo>(10); //no deduction, that why the <> are needed
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_01_frist_example" "05_01_frist_example.cpp" (in directory: /home/msk/Des
Compilation finished successfully.

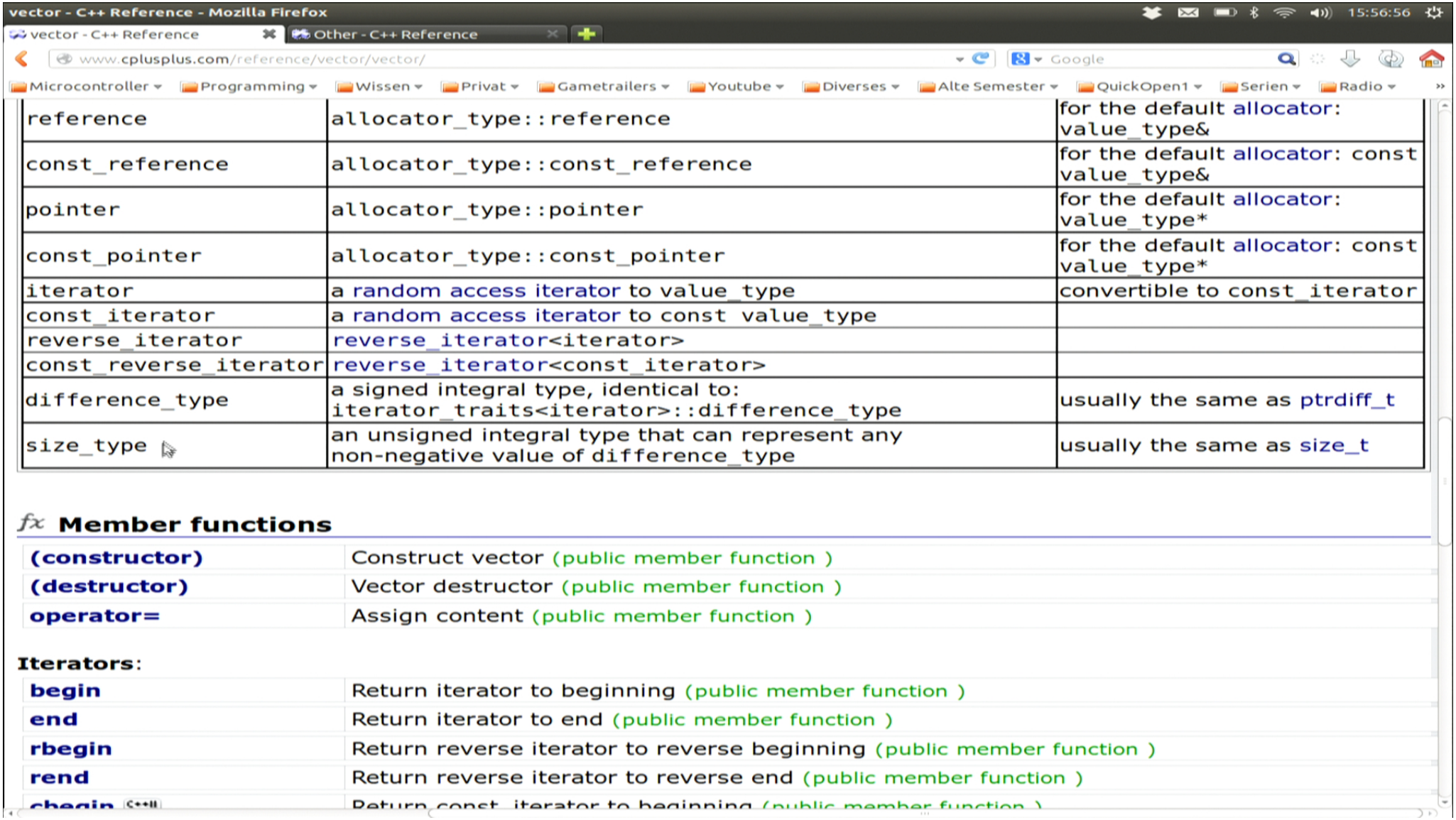

05_01_frist_example.cpp - /home/msk/Desktop/ProgTechFinal/05_sfinae - Geany
15:51:34
Documents
Files
~/Deskt...der_only
03_02_...er.hpp
03_02_...nly.cpp
~/Deskt...an_trait
04_01_...ns.cpp
04_02_...pt.cpp
04_03_...ait.cpp
04_04_...ait.cpp
~/Deskt...5_sfinae
05_01...ple.cpp
05_02_...pe.cpp
05_02_...ss.cpp
05_03_...od.cpp
05_04_...its.cpp
~/Deskt..._pitfalls
06_01_...es.cpp
~/Deskt...graming
07_01...mal.cpp
07_02_...pt.cpp
~/Deskt...ummary
08_01...def.cpp
08_02...me.cpp
08_03_...eof.cpp
08_04_...rts.cpp
08_05_...rts.cpp
08_06_...te.cpp
08_07_...te.cpp
08_08_...on.cpp
08_09_...on.cpp
08_10_...tor.cpp
~/Deskt...9_proof
09_01_...en.cpp
09_02_...ed.cpp
09_03_...m.cpp
09_04_...m.cpp
29 void fct(typename T::size_type t) {
30 PRINT_YELLOW(TYPE(T) << " has size_type")
31 }
32
33 //doesn't work for foo bc the argument given is 10 and not convertable to foo
34 template<typename T>
35 void fct(T t) {
36 PRINT_RED(TYPE(T) << " has no size_type")
37 }
38
39 //============== a type that has a typedef size_type ==============
40 struct foo {
41 typedef int size_type;
42 };
43
44 // +----------------------------------------------------+
45 // | main |
46 // +----------------------------------------------------+
47 int main(int argc, char* argv[]) {
48 CLR_SCR()
49 PRINT_CYAN("press enter to continue")
50
51 WAIT_FOR_INPUT() //just hit the enter key to continue
52 fct<foo>(10); //no deduction, that why the <> are needed
53
54 WAIT_FOR_INPUT()
55 fct<int>(10);
56
57 WAIT_FOR_INPUT()
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_01_frist_example" "05_01_frist_example.cpp" (in directory: /home/msk/Des
Compilation finished successfully.
line: 55 / 63 col: 12 sel: 8 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main


05_01_frist_example.cpp - /home/msk/Desktop/ProgTechFinal/05_sfinae - Geany
15:52:19
Documents
Files
~/Deskt...der_only
03_02_...er.hpp
03_02_...nly.cpp
~/Deskt...an_trait
04_01_...ns.cpp
04_02_...pt.cpp
04_03_...ait.cpp
04_04_...ait.cpp
~/Deskt...5_sfinae
05_01...ple.cpp
05_02_...pe.cpp
05_02_...ss.cpp
05_03_...od.cpp
05_04_...its.cpp
~/Deskt..._pitfalls
06_01_...es.cpp
~/Deskt...graming
07_01...mal.cpp
07_02_...pt.cpp
~/Deskt...ummary
08_01...def.cpp
08_02...me.cpp
08_03_...eof.cpp
08_04_...rts.cpp
08_05_...rts.cpp
08_06_...te.cpp
08_07_...te.cpp
08_08_...on.cpp
08_09_...on.cpp
08_10_...tor.cpp
~/Deskt...9_proof
09_01_...en.cpp
09_02_...ed.cpp
09_03_...m.cpp
09_04_...m.cpp
24 #include <typeinfo>
25
26 //==================== template functions ====================
27 //doesn't work for int since int doesn't have size_type
28 template<typename T>
29 void fct(typename T::size_type t) {
30 PRINT_YELLOW(TYPE(T) << " has size_type")
31 }
32
33 //doesn't work for foo bc the argument given is 10 and not convertable to foo
34 template<typename T>
35 void fct(T t) {
36 PRINT_RED(TYPE(T) << " has no size_type")
37 }
38
39 //============== a type that has a typedef size_type ==============
40 struct foo {
41 typedef int size_type;
42 };
43
44 // +-----------------------------------------------------+
45 // | main |
46 // +-----------------------------------------------------+
47 int main(int argc, char* argv[]) {
48 CLR_SCR()
49 PRINT_CYAN("press enter to continue")
50
51 WAIT_FOR_INPUT() //just hit the enter key to continue
52 fct<foo>(10); //no deduction, that why the <> are needed
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_01_frist_example" "05_01_frist_example.cpp" (in directory: /home/msk/Des
Compilation finished successfully.
line: 38 / 63 col: 0 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown


05_01_frist_example.cpp - /home/msk/Desktop/ProgTechFinal/05_sfinae - Geany
15:53:45
Documents
Files
~/Deskt...der_only
03_02_...er.hpp
03_02_...nly.cpp
~/Deskt...an_trait
04_01_...ns.cpp
04_02_...pt.cpp
04_03_...ait.cpp
04_04_...ait.cpp
~/Deskt...5_sfinae
05_01...ple.cpp
05_02_...pe.cpp
05_02_...ss.cpp
05_03_...od.cpp
05_04_...its.cpp
~/Deskt..._pitfalls
06_01_...es.cpp
~/Deskt...graming
07_01...mal.cpp
07_02_...pt.cpp
~/Deskt...ummary
08_01...def.cpp
08_02...me.cpp
08_03_...eof.cpp
08_04_...rts.cpp
08_05_...rts.cpp
08_06_...te.cpp
08_07_...te.cpp
08_08_...on.cpp
08_09_...on.cpp
08_10_...tor.cpp
~/Deskt...9_proof
09_01_...en.cpp
09_02_...ed.cpp
09_03_...m.cpp
09_04_...m.cpp
24 #include <typeinfo>
25
26 //==================== template functions ====================
27 //doesn't work for foo bc the argument given is 10 and not convertable to foo
28 template<typename T>
29 void fct(int t) {
30 PRINT_RED(TYPE(T) << " has no size_type")
31 }
32 //doesn't work for int since int doesn't have size_type
33 template<typename T>
34 void fct(typename T::size_type t) {
35 PRINT_YELLOW(TYPE(T) << " has size_type")
36 }
37
38
39 //============== a type that has a typedef size_type ==============
40 struct foo {
41 typedef int size_type;
42 };
43
44 // +----------------------------------------------------+
45 // | main |
46 // +----------------------------------------------------+
47 int main(int argc, char* argv[]) {
48 CLR_SCR()
49 PRINT_CYAN("press enter to continue")
50
51 WAIT_FOR_INPUT() //just hit the enter key to continue
52 fct<foo>(10); //no deduction, that why the <> are needed
Status
Compiler
Messages
Scribble
Terminal
05_01_frist_example.cpp: In function 'int main(int, char**)':
05_01_frist_example.cpp:52:16: error: call of overloaded 'fct(int)' is ambiguous
05_01_frist_example.cpp:52:16: note: candidates are:
05_01_frist_example.cpp:29:6: note: void fct(int) [with T = foo]
05_01_frist_example.cpp:34:6: note: void fct(typename T::size_type) [with T = foo; typename T::size_type = in
Compilation failed.
line: 29 / 63 col: 0 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown


05_02_has_type.cpp - /home/msk/Desktop/ProgTechFinal/05_sfinae - Geany
15:55:43
Documents
Files
~/Deskt...der_only
03_02_...er.hpp
03_02_...nly.cpp
~/Deskt...an_trait
04_01_...ns.cpp
04_02_...pt.cpp
04_03_...ait.cpp
04_04_...ait.cpp
~/Deskt...5_sfinae
05_01...ple.cpp
05_02_...pe.cpp
05_02_...ss.cpp
05_03_...od.cpp
05_04_...its.cpp
~/Deskt..._pitfalls
06_01_...es.cpp
~/Deskt...graming
07_01...mal.cpp
07_02_...pt.cpp
~/Deskt...ummary
08_01...def.cpp
08_02...me.cpp
08_03_...eof.cpp
08_04_...rts.cpp
08_05_...rts.cpp
08_06_...te.cpp
08_07_...te.cpp
08_08_...on.cpp
08_09_...on.cpp
08_10_...tor.cpp
~/Deskt...9_proof
09_01_...en.cpp
09_02_...ed.cpp
09_03_...m.cpp
09_04_...m.cpp
43 //=============== a type that has a typedef size_type ===============
44 struct foo {
45 typedef int size_type;
46 };
47
48 struct bar {
49
50 };
51
52 // +-----------------------------------------------------+
53 // | main |
54 // +-----------------------------------------------------+
55 int main(int argc, char* argv[]) {
56 CLR_SCR()
57 PRINT_CYAN("press enter to continue")
58
59 WAIT_FOR_INPUT() //just hit the enter key to continue
60 PRINT_NAMED(has_size_type<bar>::value)
61
62 WAIT_FOR_INPUT()
63 PRINT_NAMED(has_size_type<foo>::value)
64
65 WAIT_FOR_INPUT()
66 PRINT_NAMED(has_size_type<int>::value)
67
68 WAIT_FOR_INPUT()
69 PRINT_NAMED(has_size_type<std::vector<int>>::value)
70
71 WAIT_FOR_INPUT()
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_01_frist_example" "05_01_frist_example.cpp" (in directory: /home/msk/Des
Compilation finished successfully.
line: 66 / 76 col: 4 sel: 38 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main


vector - C++ Reference - Mozilla Firefox
15:56:53
vector - C++ Reference
Other - C++ Reference
www.cplusplus.com/reference/vector/vector/
Google
Microcontroller
Programming
Wissen
Privat
Gametrailers
Youtube
Diverses
Alte Semester
QuickOpen1
Serien
Radio
std::vector
template < class T, class Alloc = allocator<T> > class vector; // generic template
Vector
Vectors are sequence containers representing arrays that can change in size.
Just like arrays, vectors use contiguous storage locations for their elements, which means that their elements can also be accessed using offsets on regular pointers to its elements, and just as efficiently as in arrays. But unlike arrays, their size can change dynamically, with their storage being handled automatically by the container.
Internally, vectors use a dynamically allocated array to store their elements. This array may need to be reallocated in order to grow in size when new elements are inserted, which implies allocating a new array and moving all elements to it. This is a relatively expensive task in terms of processing time, and thus, vectors do not reallocate each time an element is added to the container.
Instead, vector containers may allocate some extra storage to accommodate for possible growth, and thus the container may have an actual capacity greater than the storage strictly needed to contain its elements (i.e., its size). Libraries can implement different strategies for growth to balance between memory usage and reallocations, but in any case, reallocations should only happen at logarithmically growing intervals of size so that the insertion of individual elements at the end of the vector can be provided with amortized constant time complexity (see push_back).
Therefore, compared to arrays, vectors consume more memory in exchange for the ability to manage storage and grow dynamically in an efficient way.
Compared to the other dynamic sequence containers (deques, lists and forward_lists), vectors are very efficient accessing its elements (just like arrays) and relatively efficient adding or removing elements from its end. For operations that involve inserting or removing elements at positions other than the end, they perform worse than the others, and have less consistent iterators and references than lists and forward_lists.
Container properties
Sequence
Elements in sequence containers are ordered in a strict linear sequence. Individual elements are accessed by




| | | |
|---|---|---|
| reference | allocator_type::reference | for the default allocator: value_type& |
| const_reference | allocator_type::const_reference | for the default allocator: const value_type& |
| pointer | allocator_type::pointer | for the default allocator: value_type* |
| const_pointer | allocator_type::const_pointer | for the default allocator: const value_type* |
| iterator | a random access iterator to value_type | convertible to const_iterator |
| const_iterator | a random access iterator to const value_type | |
| reverse_iterator | reverse_iterator<iterator> | |
| const_reverse_iterator | reverse_iterator<const_iterator> | |
| difference_type | a signed integral type, identical to: iterator_traits<iterator>::difference_type | usually the same as ptrdiff_t |
| size_type | an unsigned integral type that can represent any non-negative value of difference_type | usually the same as size_t |

## Member functions

| | |
|---|---|
| **(constructor)** | Construct vector (public member function ) |
| **(destructor)** | Vector destructor (public member function ) |
| **operator=** | Assign content (public member function ) |

**Iterators**:

| | |
|---|---|
| **begin** | Return iterator to beginning (public member function ) |
| **end** | Return iterator to end (public member function ) |
| **rbegin** | Return reverse iterator to reverse beginning (public member function ) |
| **rend** | Return reverse iterator to reverse end (public member function ) |
| **cbegin** | Return const_iterator to beginning (public member function ) |

05_02_has_type.cpp - /home/msk/Desktop/ProgTechFinal/05_sfinae - Geany

```
31 //==================== has_size_type ====================
32 template<typename T>
33 struct has_size_type {
34
35     template<typename U>
36     static char check(typename U::size_type * i);
37
38     template<typename U>
39     static double check(...);
40
41     enum {value = (sizeof(check<T>(NULL)) == sizeof(char))};
42 };
43 //============== a type that has a typedef size_type ==============
44 struct foo {
45     typedef int size_type;
46 };
47
48 struct bar {
49
50 };
51
52 //   +-----------------------------------------------------+
53 //   |                      main                           |
54 //   +-----------------------------------------------------+
55 int main(int argc, char* argv[]) {
56     CLR_SCR()
57     PRINT_CYAN("press enter to continue")
58
59     WAIT_FOR_INPUT() //just hit the enter key to cont
60     PRINT_NAMED(has_size_type<bar>::value)
```

Status
Compiler
Messages
Scribble
Terminal

```
g++-4.7 -std=c++11 -O2 -o "05_02_has_type" "05_02_has_type.cpp" (in
Compilation finished successfully.
```

line: 36 / 76  col: 15  sel: 4  INS  SP  mode: Unix (LF)  encoding: UTF-8  filetype: C++  scope: PRINT_MAGENTA

*05_02_has_type.cpp - /home/msk/Desktop/ProgTechFinal/05_sfinae - Geany

```
48 struct bar {
49
50 };
51
52 //   +------------------------------------------------------+
53 //   |                        main                          |
54 //   +------------------------------------------------------+
55 int main(int argc, char* argv[]) {
56     CLR_SCR()
57     PRINT_CYAN("press enter to continue")
58
59     my_template<foo> a;
60
61     WAIT_FOR_INPUT() //just hit the enter key to continue
62     PRINT_NAMED(has_size_type<bar>::value)
63
64     WAIT_FOR_INPUT()
65     PRINT_NAMED(has_size_type<foo>::value)
66
67     WAIT_FOR_INPUT()
68     PRINT_NAMED(has_size_type<int>::value)
69
70     WAIT_FOR_INPUT()
71     PRINT_NAMED(has_size_type<std::vector<int>>::value)
72
73     WAIT_FOR_INPUT()
74     PRINT_GREEN("we can find out if a type contains a typedef")
75
76     return 0;
```

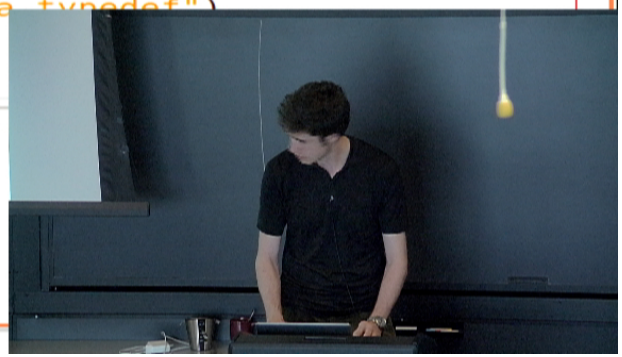
Documents | Files

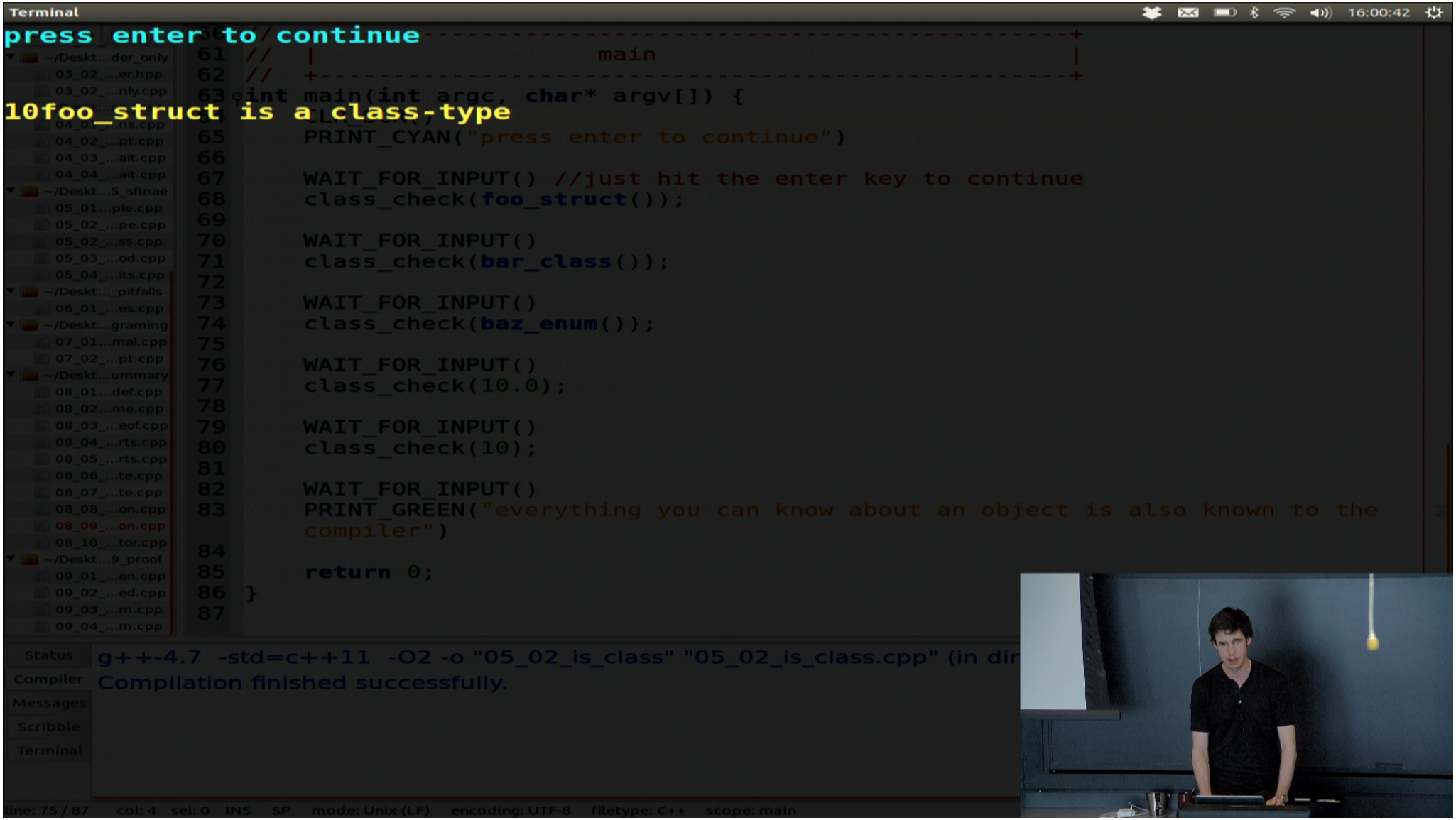
~/Deskt...der_only
03_02_...er.hpp
03_02_...nly.cpp
~/Deskt...an_trait
04_01_...ns.cpp
04_02_...pt.cpp
04_03_...ait.cpp
04_04_...ait.cpp
~/Deskt...5_sfinae
05_01...ple.cpp
05_02_...pe.cpp
05_02_...ss.cpp
05_03_...od.cpp
05_04_...its.cpp
~/Deskt..._pitfalls
06_01_...es.cpp
~/Deskt...graming
07_01...mal.cpp
07_02_...pt.cpp
~/Deskt...ummary
08_01...def.cpp
08_02...me.cpp
08_03_...eof.cpp
08_04_...rts.cpp
08_05_...rts.cpp
08_06_...te.cpp
08_07_...te.cpp
08_08_...on.cpp
08_09_...on.cpp
08_10_...tor.cpp
~/Deskt...9_proof
09_01_...en.cpp
09_02_...ed.cpp
09_03_...m.cpp
09_04_...m.cpp

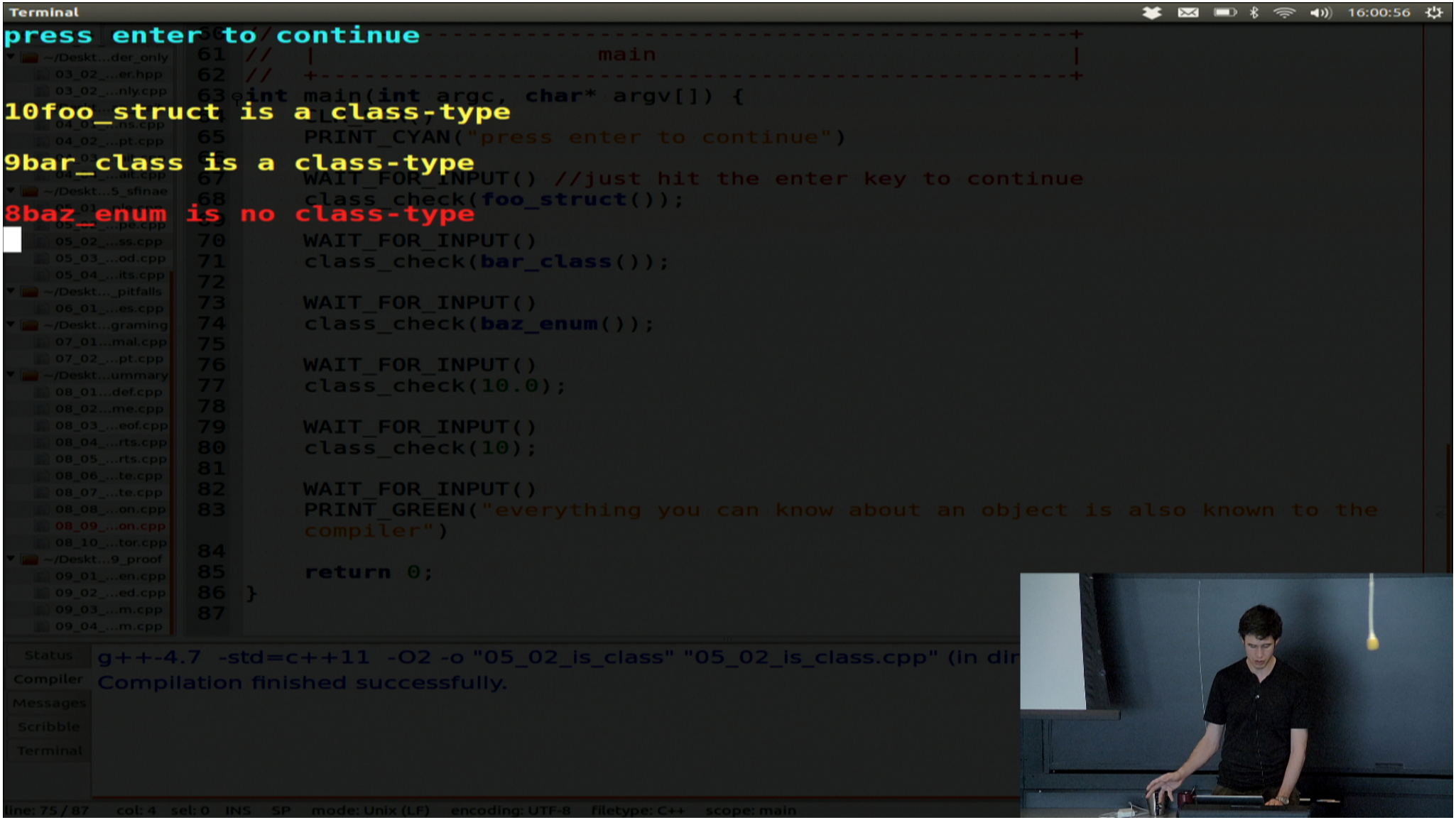
Status
Compiler
Messages
Scribble
Terminal

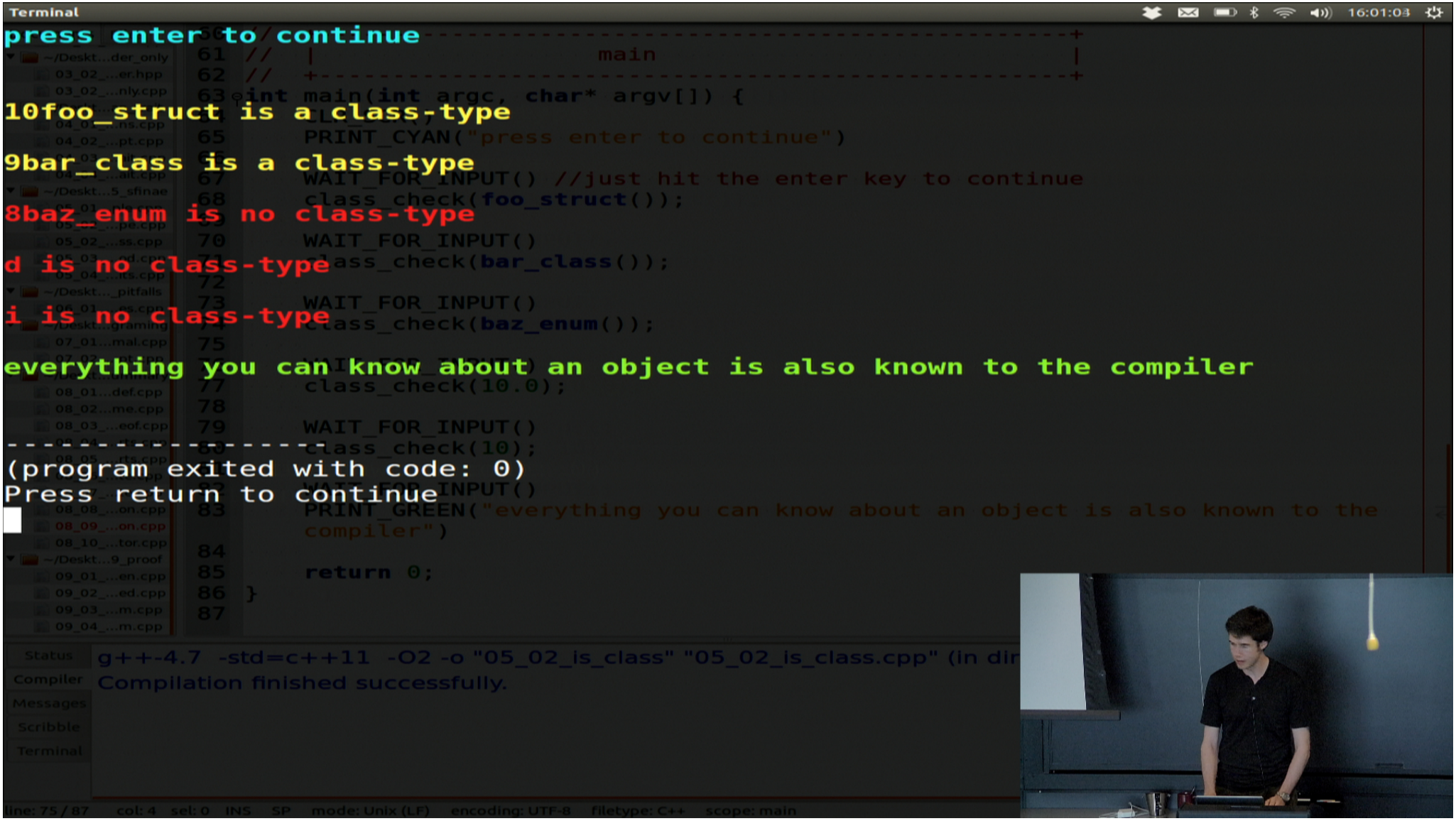
```
g++-4.7 -std=c++11 -O2 -o "05_02_has_type" "05_02_has_type.cpp" (in
Compilation finished successfully.
```

line: 59 / 78  col: 23  sel: 0  INS  SP  MOD  mode: Unix (LF)  encoding: UTF-8  filetype: C++  scope: main




Terminal
16:00:42
press enter to continue
10foo_struct is a class-type
61 // | main |
62 // +----------+
63 int main(int argc, char* argv[]) {
65 PRINT_CYAN("press enter to continue")
67 WAIT_FOR_INPUT() //just hit the enter key to continue
68 class_check(foo_struct());
70 WAIT_FOR_INPUT()
71 class_check(bar_class());
73 WAIT_FOR_INPUT()
74 class_check(baz_enum());
76 WAIT_FOR_INPUT()
77 class_check(10.0);
79 WAIT_FOR_INPUT()
80 class_check(10);
82 WAIT_FOR_INPUT()
83 PRINT_GREEN("everything you can know about an object is also known to the compiler")
85 return 0;
86 }
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_02_is_class" "05_02_is_class.cpp" (in dir
Compilation finished successfully.


Terminal
16:00:56
press enter to continue
10foo_struct is a class-type
9bar_class is a class-type
8baz_enum is no class-type
61 // |  main  |
62 // +----------+
63 int main(int argc, char* argv[]) {
65 PRINT_CYAN("press enter to continue")
67 WAIT_FOR_INPUT() //just hit the enter key to continue
68 class_check(foo_struct());
70 WAIT_FOR_INPUT()
71 class_check(bar_class());
73 WAIT_FOR_INPUT()
74 class_check(baz_enum());
76 WAIT_FOR_INPUT()
77 class_check(10.0);
79 WAIT_FOR_INPUT()
80 class_check(10);
82 WAIT_FOR_INPUT()
83 PRINT_GREEN("everything you can know about an object is also known to the compiler")
85 return 0;
86 }
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_02_is_class" "05_02_is_class.cpp" (in dir
Compilation finished successfully.


Terminal
16:01:03
press enter to continue
10foo_struct is a class-type
9bar_class is a class-type
8baz_enum is no class-type
d is no class-type
i is no class-type
everything you can know about an object is also known to the compiler
(program exited with code: 0)
Press return to continue
main
int main(int argc, char* argv[]) {
PRINT_CYAN("press enter to continue")
WAIT_FOR_INPUT() //just hit the enter key to continue
class_check(foo_struct());
WAIT_FOR_INPUT()
class_check(bar_class());
WAIT_FOR_INPUT()
class_check(baz_enum());
class_check(10.0);
WAIT_FOR_INPUT()
class_check(10);
PRINT_GREEN("everything you can know about an object is also known to the compiler")
return 0;
}
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_02_is_class" "05_02_is_class.cpp" (in dir
Compilation finished successfully.

05_02_is_class.cpp - /home/msk/Desktop/ProgTechFinal/05_sfinae - Geany

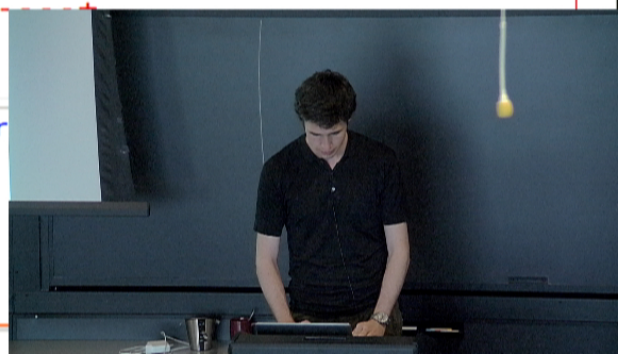
```
36
37      template<typename U>
38      static double check(...); //just catch anything else (variadic function)
39
40      enum {value = (sizeof(char) == sizeof(check<T>(NULL))) };
41  };
42  //=================== some classes ====================
43  struct foo_struct {
44  };
45
46  class bar_class {
47  };
48
49  enum baz_enum {
50  };
51
52  template<typename T>
53  void class_check(T const & t) {
54      if(is_class<T>::value)
55          PRINT_YELLOW(TYPE(T) << " is a class-type")
56      else
57          PRINT_RED(TYPE(T) << " is no class-type")
58  }
59
60  //  +-----------------------------------------------------+
61  //  |                      main                           |
62  //  +-----------------------------------------------------+
63  int main(int argc, char* argv[]) {
64      CLR_SCR()
65      PRINT_CYAN("press enter to continue")
```

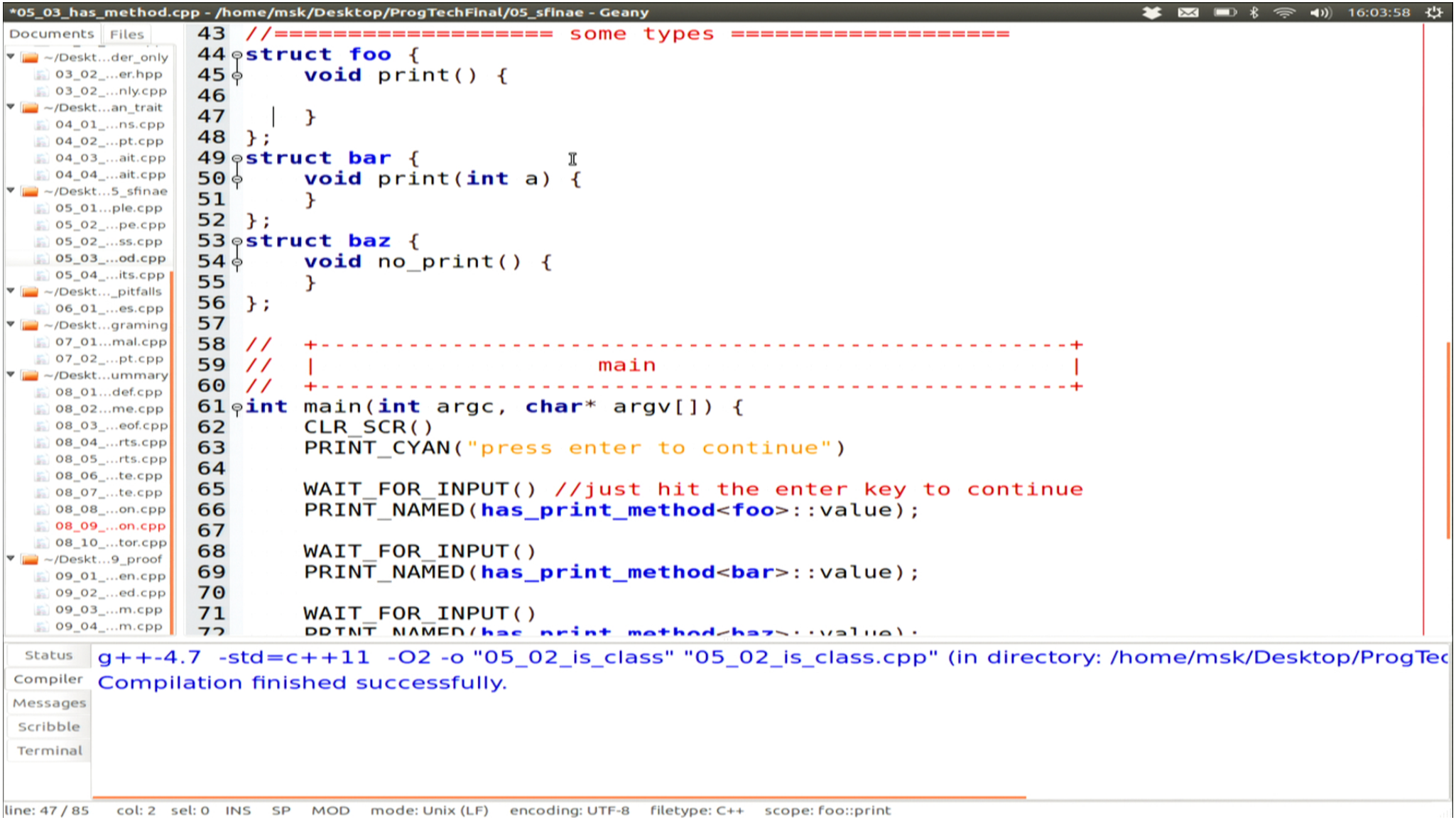
Status
Compiler
Messages
Scribble
Terminal

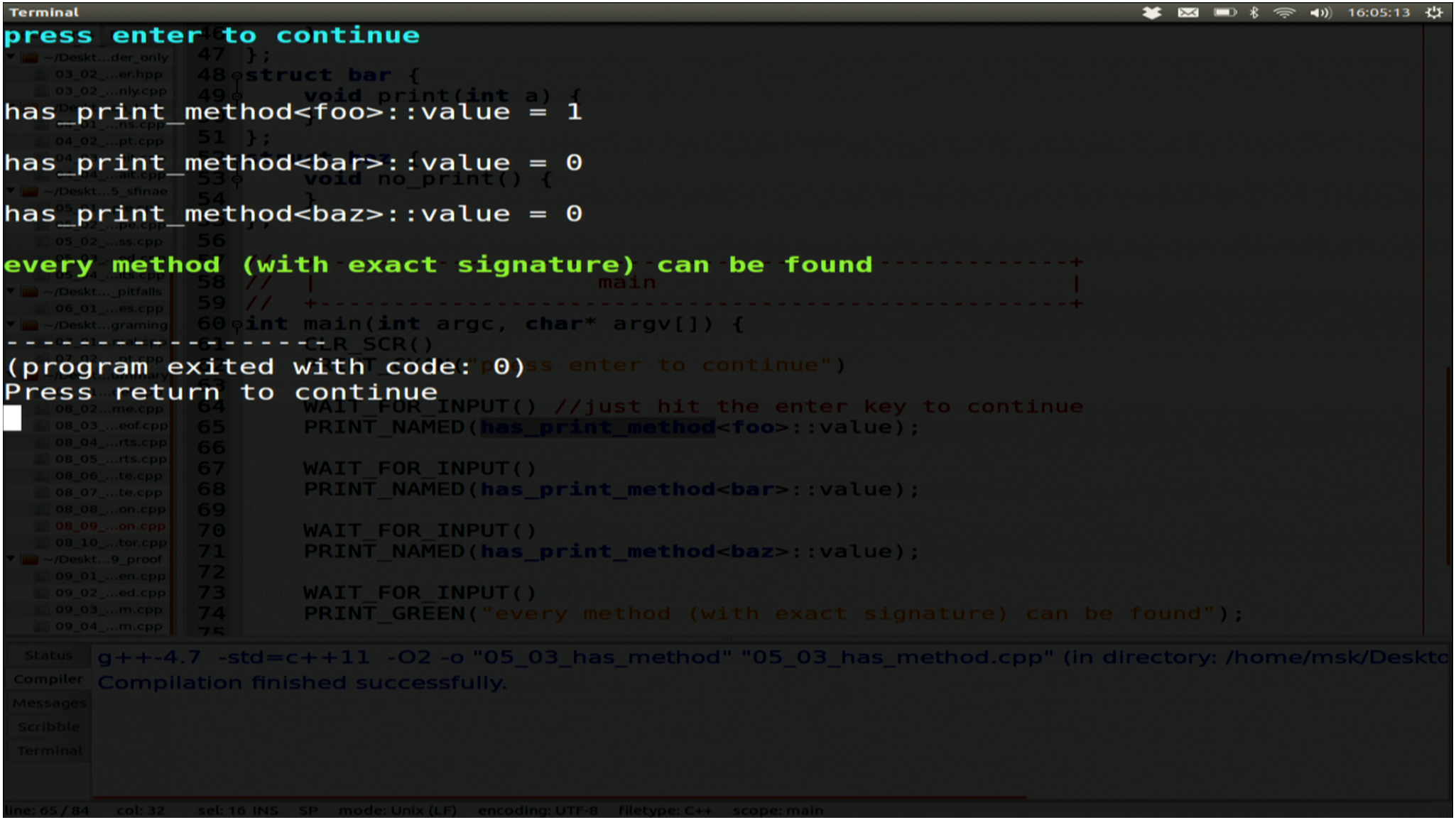
g++-4.7 -std=c++11 -O2 -o "05_02_is_class" "05_02_is_class.cpp" (in dir

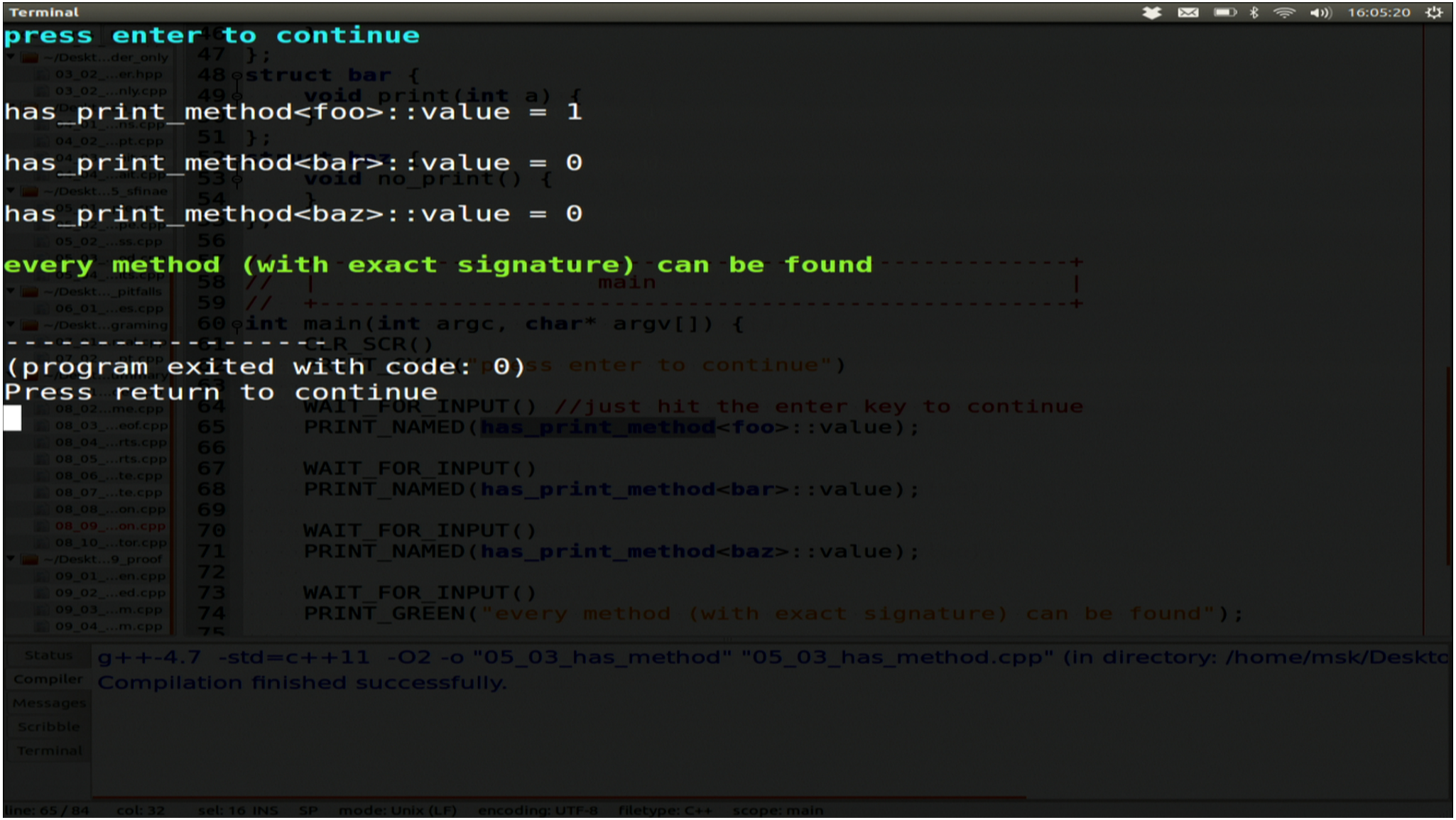
Compilation finished successfully.

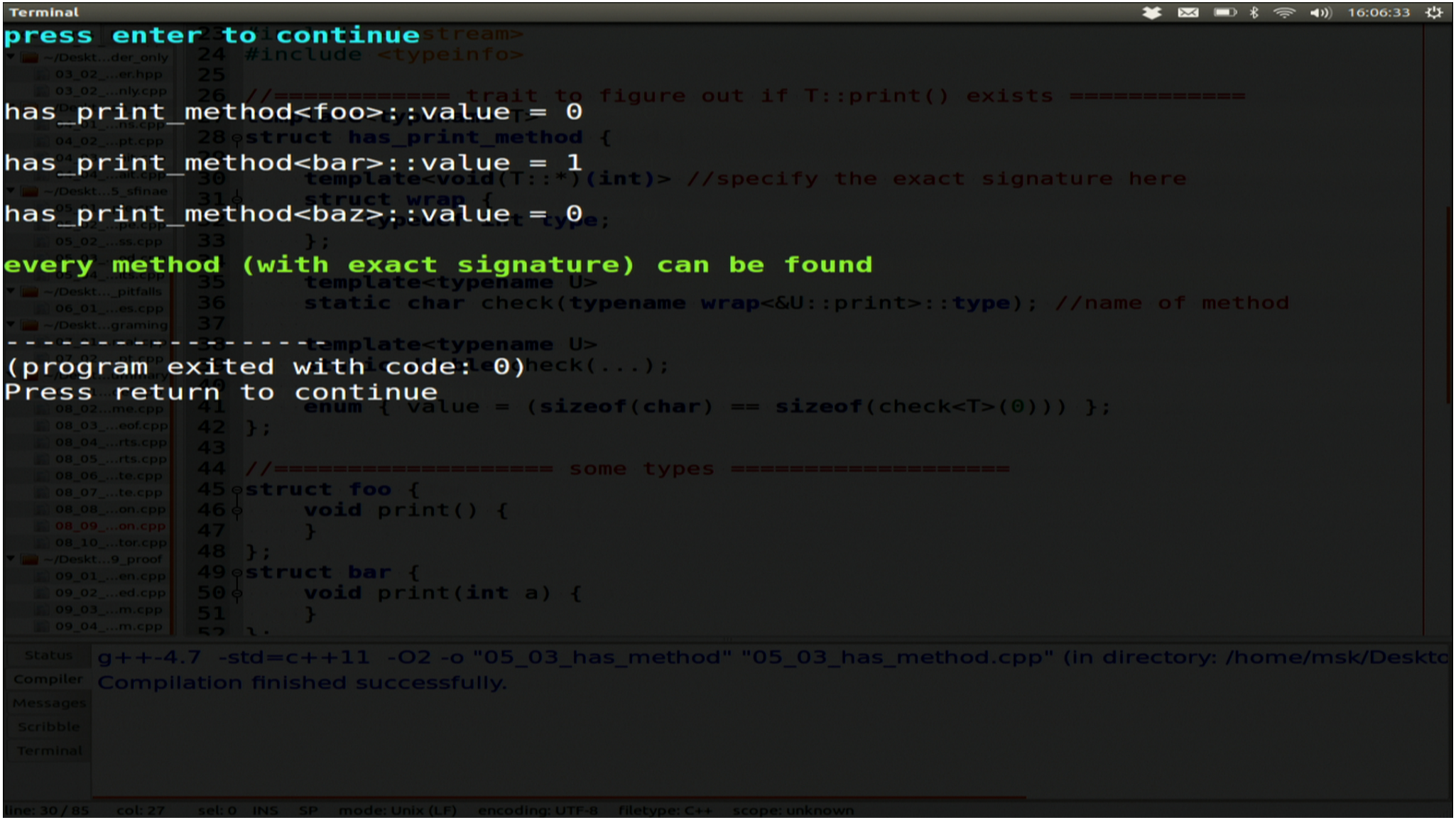
line: 75 / 87  col: 4  sel: 0  INS  SP  mode: Unix (LF)  encoding: UTF-8  filetype: C++  scope: main




*05_03_has_method.cpp - /home/msk/Desktop/ProgTechFinal/05_sfinae - Geany
16:03:58
Documents
Files
~/Deskt...der_only
03_02_...er.hpp
03_02_...nly.cpp
~/Deskt...an_trait
04_01_...ns.cpp
04_02_...pt.cpp
04_03_...ait.cpp
04_04_...ait.cpp
~/Deskt...5_sfinae
05_01...ple.cpp
05_02_...pe.cpp
05_02_...ss.cpp
05_03_...od.cpp
05_04_...its.cpp
~/Deskt..._pitfalls
06_01_...es.cpp
~/Deskt...graming
07_01...mal.cpp
07_02_...pt.cpp
~/Deskt...ummary
08_01...def.cpp
08_02...me.cpp
08_03_...eof.cpp
08_04_...rts.cpp
08_05_...rts.cpp
08_06_...te.cpp
08_07_...te.cpp
08_08_...on.cpp
08_09_...on.cpp
08_10_...tor.cpp
~/Deskt...9_proof
09_01_...en.cpp
09_02_...ed.cpp
09_03_...m.cpp
09_04_...m.cpp
43 //==================== some types ====================
44 struct foo {
45 void print() {
46
47 }
48 };
49 struct bar {
50 void print(int a) {
51 }
52 };
53 struct baz {
54 void no_print() {
55 }
56 };
57
58 // +----------------------------------------------------+
59 // | main |
60 // +----------------------------------------------------+
61 int main(int argc, char* argv[]) {
62 CLR_SCR()
63 PRINT_CYAN("press enter to continue")
64
65 WAIT_FOR_INPUT() //just hit the enter key to continue
66 PRINT_NAMED(has_print_method<foo>::value);
67
68 WAIT_FOR_INPUT()
69 PRINT_NAMED(has_print_method<bar>::value);
70
71 WAIT_FOR_INPUT()
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_02_is_class" "05_02_is_class.cpp" (in directory: /home/msk/Desktop/ProgTec
Compilation finished successfully.
line: 47 / 85 col: 2 sel: 0 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: foo::print


Terminal
16:05:13
press enter to continue
has_print_method<foo>::value = 1
has_print_method<bar>::value = 0
has_print_method<baz>::value = 0
every method (with exact signature) can be found
(program exited with code: 0)
Press return to continue
47 };
48 struct bar {
49 void print(int a) {
51 };
53 void no_print() {
54 }
58 // main
59 //
60 int main(int argc, char* argv[]) {
CLR_SCR()
64 WAIT_FOR_INPUT() //just hit the enter key to continue
65 PRINT_NAMED(has_print_method<foo>::value);
66
67 WAIT_FOR_INPUT()
68 PRINT_NAMED(has_print_method<bar>::value);
69
70 WAIT_FOR_INPUT()
71 PRINT_NAMED(has_print_method<baz>::value);
72
73 WAIT_FOR_INPUT()
74 PRINT_GREEN("every method (with exact signature) can be found");
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_03_has_method" "05_03_has_method.cpp" (in directory: /home/msk/Deskto
Compilation finished successfully.
line: 65 / 84 col: 32 sel: 16 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main


Terminal
16:05:20
press enter to continue
has_print_method<foo>::value = 1
has_print_method<bar>::value = 0
has_print_method<baz>::value = 0
every method (with exact signature) can be found
(program exited with code: 0)
Press return to continue
main
WAIT_FOR_INPUT() //just hit the enter key to continue
PRINT_NAMED(has_print_method<foo>::value);
WAIT_FOR_INPUT()
PRINT_NAMED(has_print_method<bar>::value);
WAIT_FOR_INPUT()
PRINT_NAMED(has_print_method<baz>::value);
WAIT_FOR_INPUT()
PRINT_GREEN("every method (with exact signature) can be found");
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_03_has_method" "05_03_has_method.cpp" (in directory: /home/msk/Deskto
Compilation finished successfully.


Terminal
16:06:33
press enter to continue
has_print_method<foo>::value = 0
has_print_method<bar>::value = 1
has_print_method<baz>::value = 0
every method (with exact signature) can be found
(program exited with code: 0)
Press return to continue
#include <typeinfo>
//============ trait to figure out if T::print() exists ============
struct has_print_method {
template<void(T::*)(int)> //specify the exact signature here
struct wrap {
};
template<typename U>
static char check(typename wrap<&U::print>::type); //name of method
template<typename U>
enum { value = (sizeof(char) == sizeof(check<T>(0))) };
};
//==================== some types ====================
struct foo {
void print() {
}
};
struct bar {
void print(int a) {
}
};
g++-4.7 -std=c++11 -O2 -o "05_03_has_method" "05_03_has_method.cpp" (in directory: /home/msk/Deskto
Compilation finished successfully.
Status
Compiler
Messages
Scribble
Terminal
line: 30 / 85 col: 27 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown

05_04_std_traits.cpp - /home/msk/Desktop/ProgTechFinal/05_sfinae - Geany

```
47      PRINT_RED("fct for " << TYPE(T) << " without +")
48  }
49  //if there is only one parameter, put enable_if on the return type
50  //you cannot wrap the first parameter with enable_if bc of deduction
51  //==================== some types ====================
52  struct foo {};
53
54  foo operator+(foo const & a, foo const & b){
55      return foo();
56  }
57
58  struct bar {
59      void operator+(bar const & a) const {
60      }
61  };
62
63  struct no_plus {
64  };
65
66  //   +-----------------------------------------------------+
67  //   |                      main                           |
68  //   +-----------------------------------------------------+
69  int main(int argc, char* argv[]) {
70      CLR_SCR()
71      PRINT_CYAN("press enter to continue")
72
73      WAIT_FOR_INPUT() //just hit the enter key to continue
74      fct(foo());
75
76      WAIT_FOR_INPUT()
```

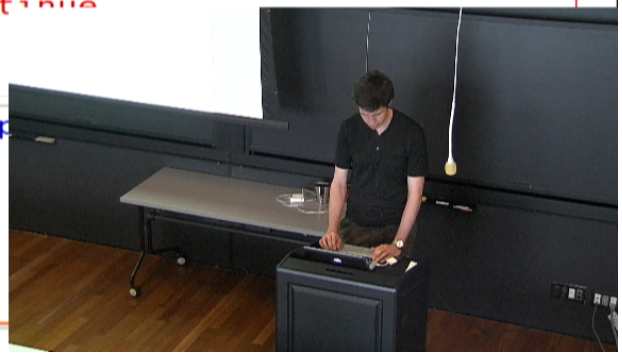
Documents | Files

~/Deskt...der_only: 03_02_...er.hpp, 03_02_...nly.cpp
~/Deskt...an_trait: 04_01_...ns.cpp, 04_02_...pt.cpp, 04_03_...ait.cpp, 04_04_...ait.cpp
~/Deskt...5_sfinae: 05_01...ple.cpp, 05_02_...pe.cpp, 05_02_...ss.cpp, 05_03_...od.cpp, 05_04_...its.cpp
~/Deskt..._pitfalls: 06_01_...es.cpp
~/Deskt...graming: 07_01...mal.cpp, 07_02_...pt.cpp
~/Deskt...ummary: 08_01...def.cpp, 08_02...me.cpp, 08_03_...eof.cpp, 08_04_...rts.cpp, 08_05_...rts.cpp, 08_06_...te.cpp, 08_07_...te.cpp, 08_08_...on.cpp, 08_09_...on.cpp, 08_10_...tor.cpp
~/Deskt...9_proof: 09_01_...en.cpp, 09_02_...ed.cpp, 09_03_...m.cpp, 09_04_...m.cpp

Status
Compiler
Messages
Scribble
Terminal

```
g++-4.7 -std=c++11 -O2 -o "05_03_has_method" "05_03_has_method.cp
Compilation finished successfully.
```

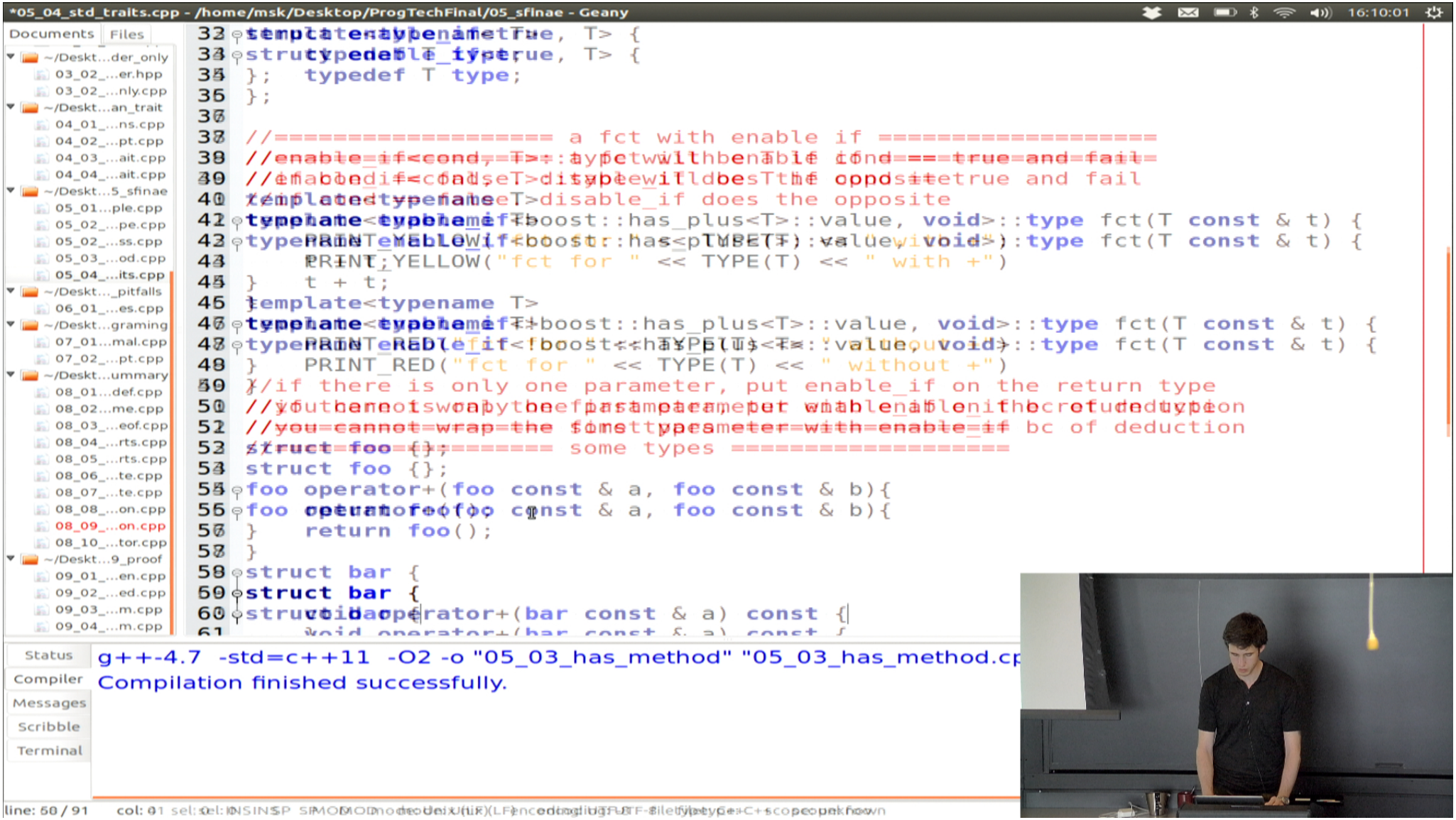
line: 63 / 90  col: 4  sel: 0  INS  SP  mode: Unix (LF)  encoding: UTF-8  filetype: C++  scope: unknown

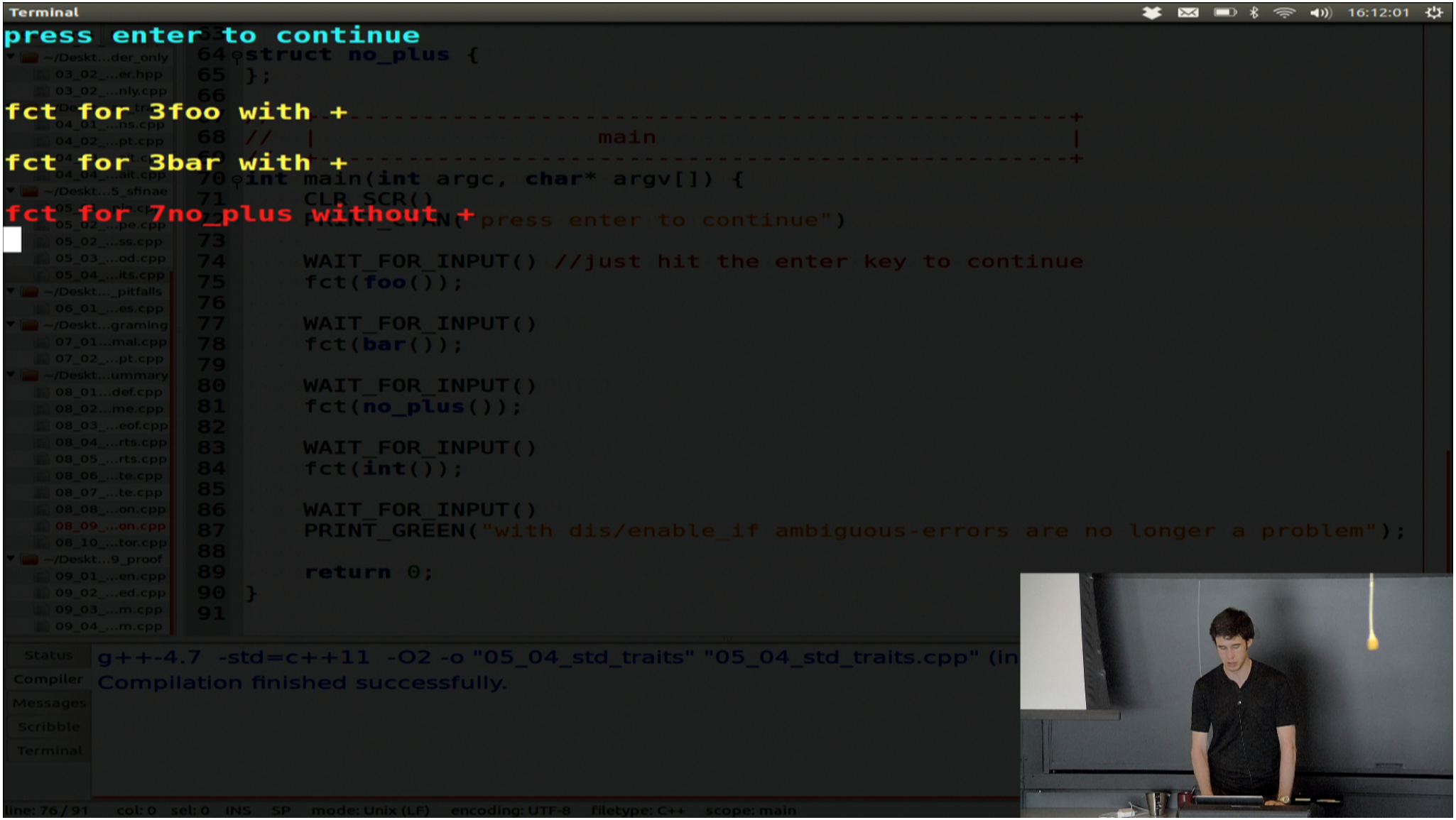


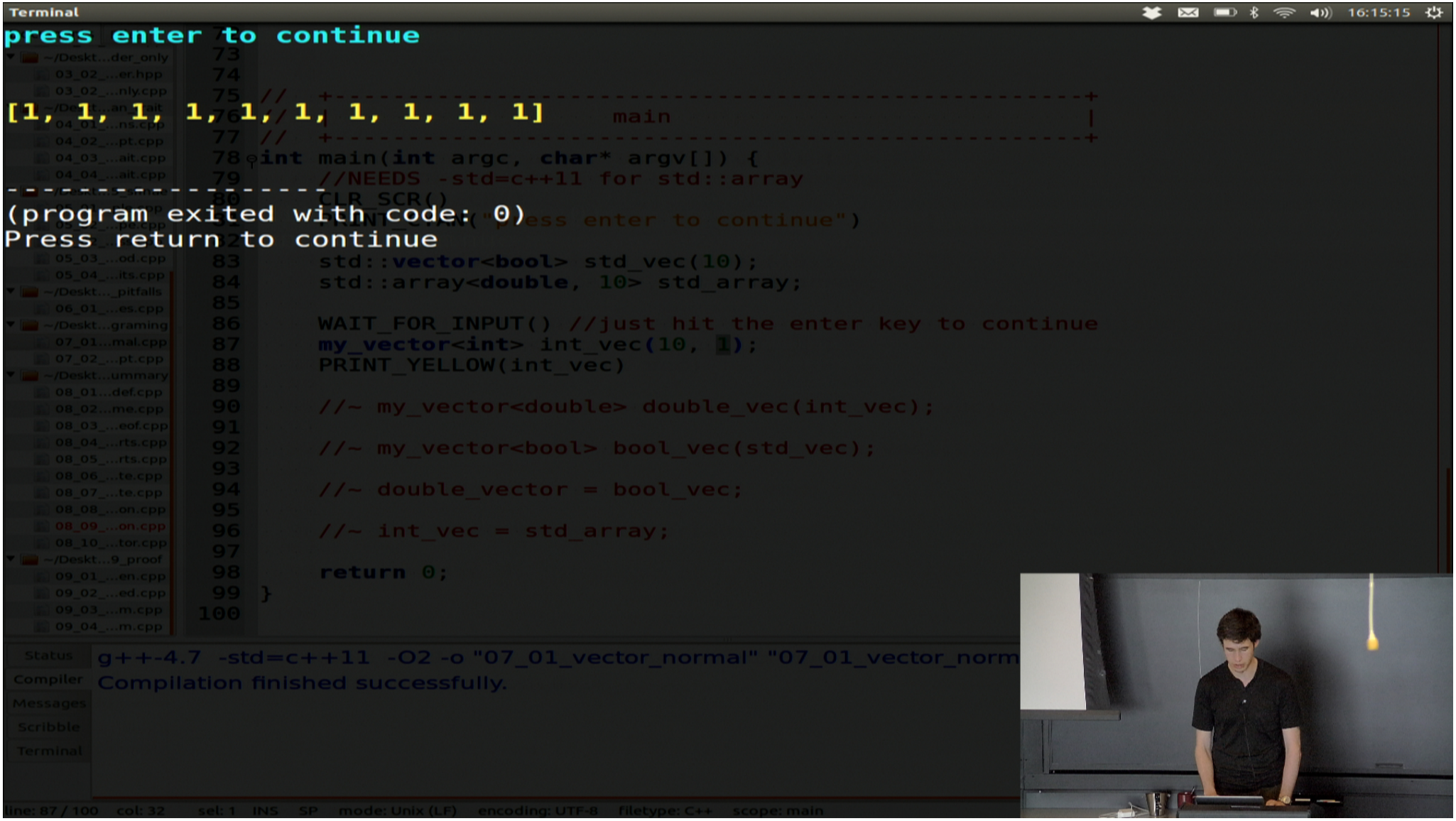
*05_04_std_traits.cpp - /home/msk/Desktop/ProgTechFinal/05_sfinae - Geany

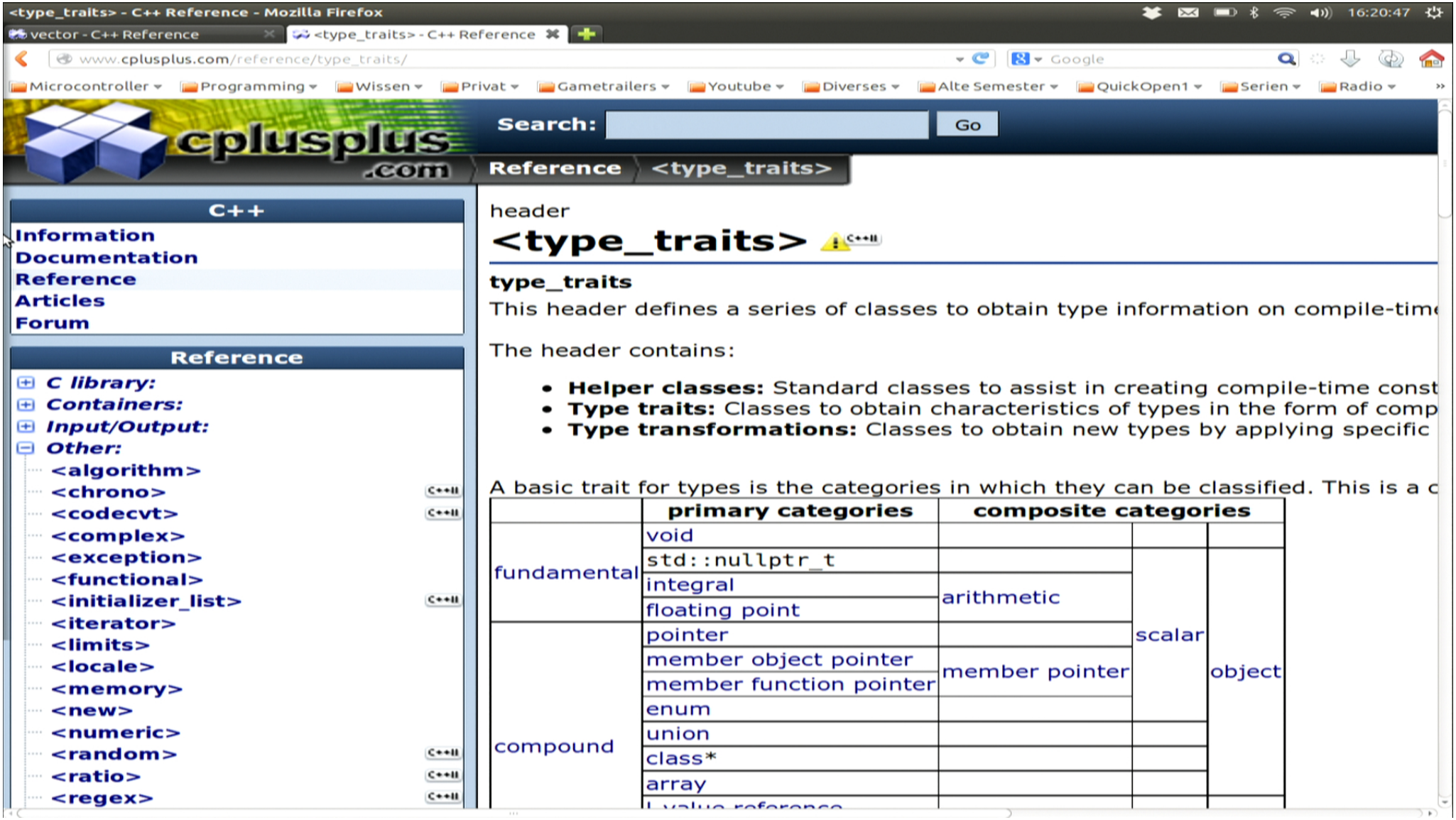
```
24 #include <typeinfo>
25 #include <boost/type_traits.hpp>
26 #include <boost/utility/enable_if.hpp> //enable_if is now part c++11
27
28 template<bool cond, typename T>
29 struct enable_if {
30 };
31
32 template<typename T>
33 struct enable_if<true, T> {
34     typedef T type;
35 };
36
37
38 //==================== a fct with enable if ====================
39 //enable_if<cond, T>::type will be T if cond == true and fail
40 //if cond == false. disable_if does the opposite
41 template<typename T>
42 typename enable_if<boost::has_plus<T>::value, void>::type fct(T const & t) {
43     PRINT_YELLOW("fct for " << TYPE(T) << " with +")
44     t + t;
45 }
46 template<typename T>
47 typename enable_if<!boost::has_plus<T>::value, void>::type fct(T const & t) {
48     PRINT_RED("fct for " << TYPE(T) << " without +")
49 }
50 //if there is only one parameter, put enable_if on the return type
51 //you cannot wrap the first parameter with enable_if
52 //==================== some types ====================
```

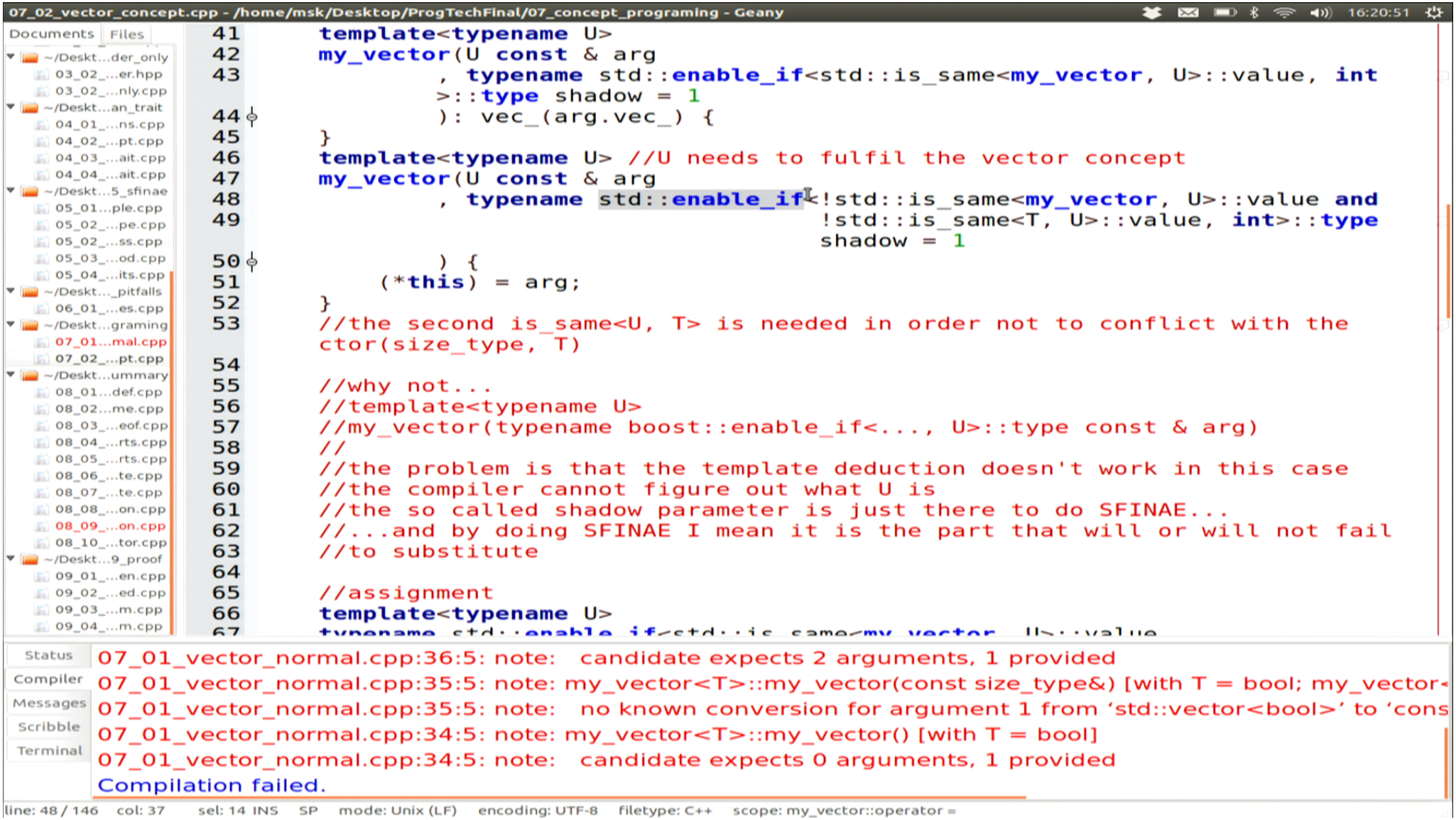
```
g++-4.7 -std=c++11 -O2 -o "05_03_has_method" "05_03_has_method.cp
Compilation finished successfully.
```

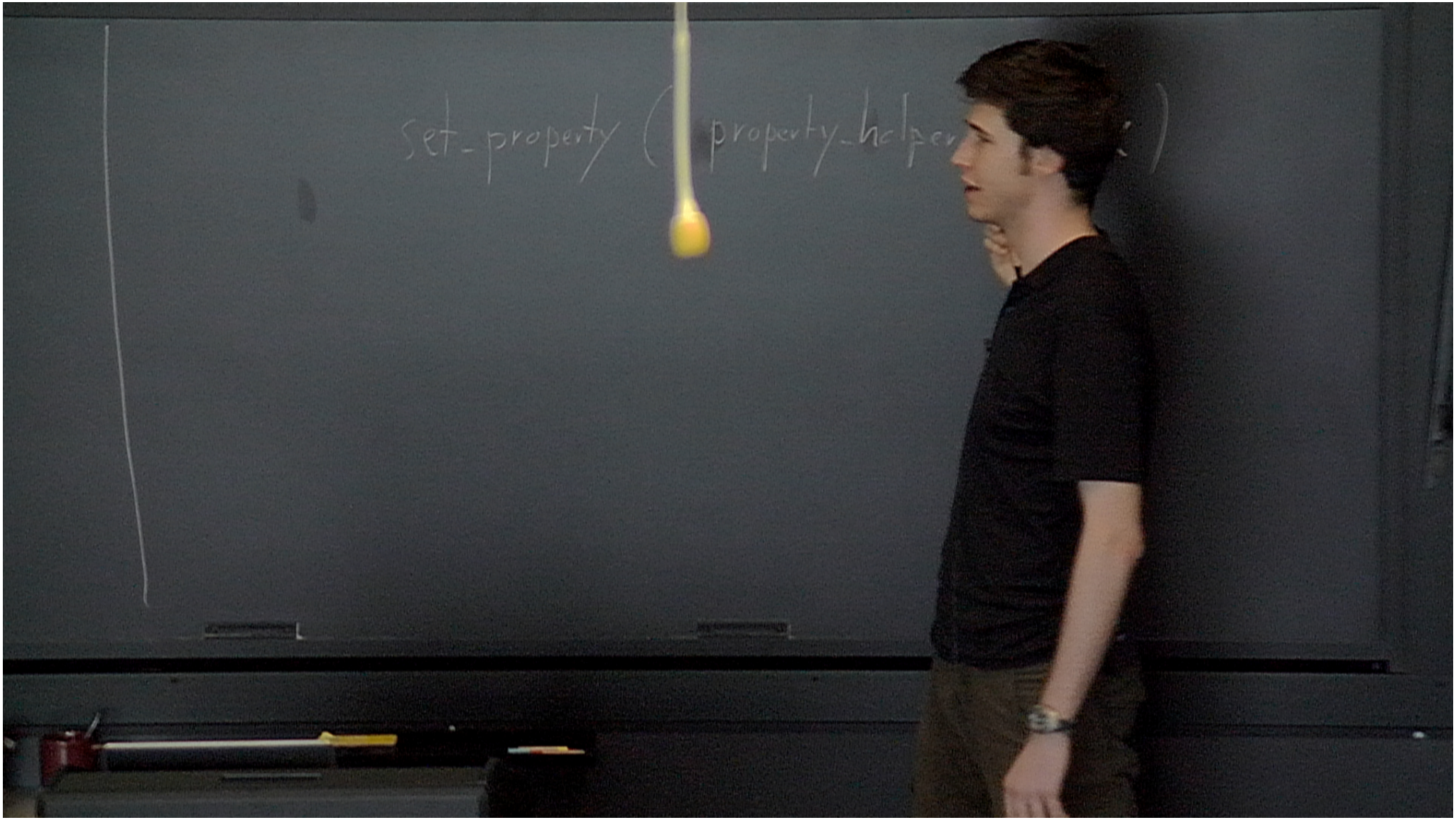

*05_04_std_traits.cpp - /home/msk/Desktop/ProgTechFinal/05_sfinae - Geany
16:10:01
Documents
Files
~/Deskt...der_only
03_02_...er.hpp
03_02_...nly.cpp
~/Deskt...an_trait
04_01_...ns.cpp
04_02_...pt.cpp
04_03_...ait.cpp
04_04_...ait.cpp
~/Deskt...5_sfinae
05_01...ple.cpp
05_02_...pe.cpp
05_02_...ss.cpp
05_03_...od.cpp
05_04_...its.cpp
~/Deskt..._pitfalls
06_01_...es.cpp
~/Deskt...graming
07_01...mal.cpp
07_02_...pt.cpp
~/Deskt...ummary
08_01...def.cpp
08_02...me.cpp
08_03_...eof.cpp
08_04_...rts.cpp
08_05_...rts.cpp
08_06_...te.cpp
08_07_...te.cpp
08_08_...on.cpp
08_09_...on.cpp
08_10_...tor.cpp
~/Deskt...9_proof
09_01_...en.cpp
09_02_...ed.cpp
09_03_...m.cpp
09_04_...m.cpp
template<typename T, bool, T> {
struct foo {};
foo operator+(foo const & a, foo const & b){
struct bar {
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_03_has_method" "05_03_has_method.cp
Compilation finished successfully.
line: 60 / 91

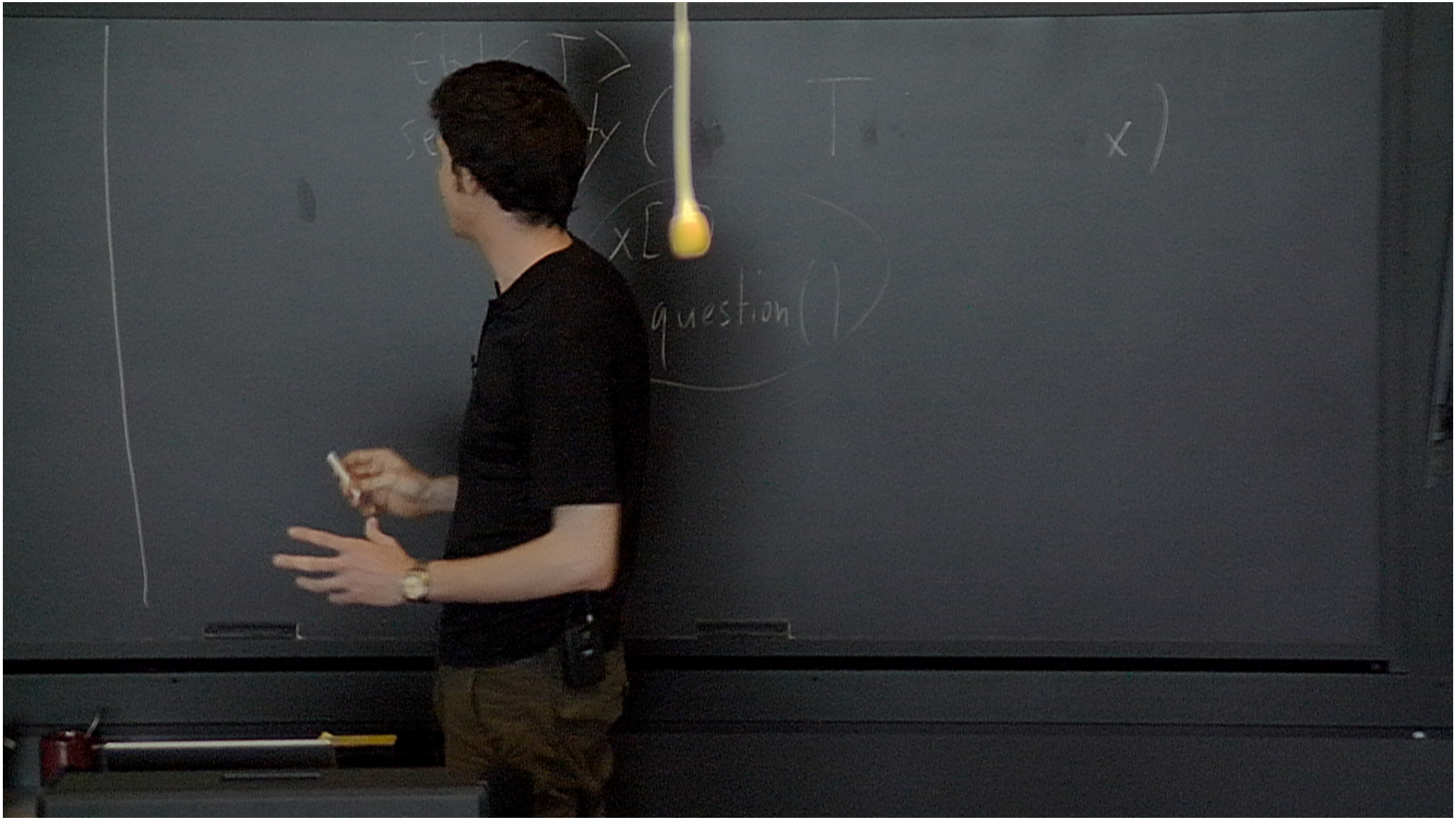

Terminal
16:12:01
press enter to continue
fct for 3foo with +
fct for 3bar with +
fct for 7no_plus without +
struct no_plus {
};
// | main |
int main(int argc, char* argv[]) {
CLR_SCR()
PRINT_CYAN("press enter to continue")
WAIT_FOR_INPUT() //just hit the enter key to continue
fct(foo());
WAIT_FOR_INPUT()
fct(bar());
WAIT_FOR_INPUT()
fct(no_plus());
WAIT_FOR_INPUT()
fct(int());
WAIT_FOR_INPUT()
PRINT_GREEN("with dis/enable_if ambiguous-errors are no longer a problem");
return 0;
}
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "05_04_std_traits" "05_04_std_traits.cpp" (in
Compilation finished successfully.

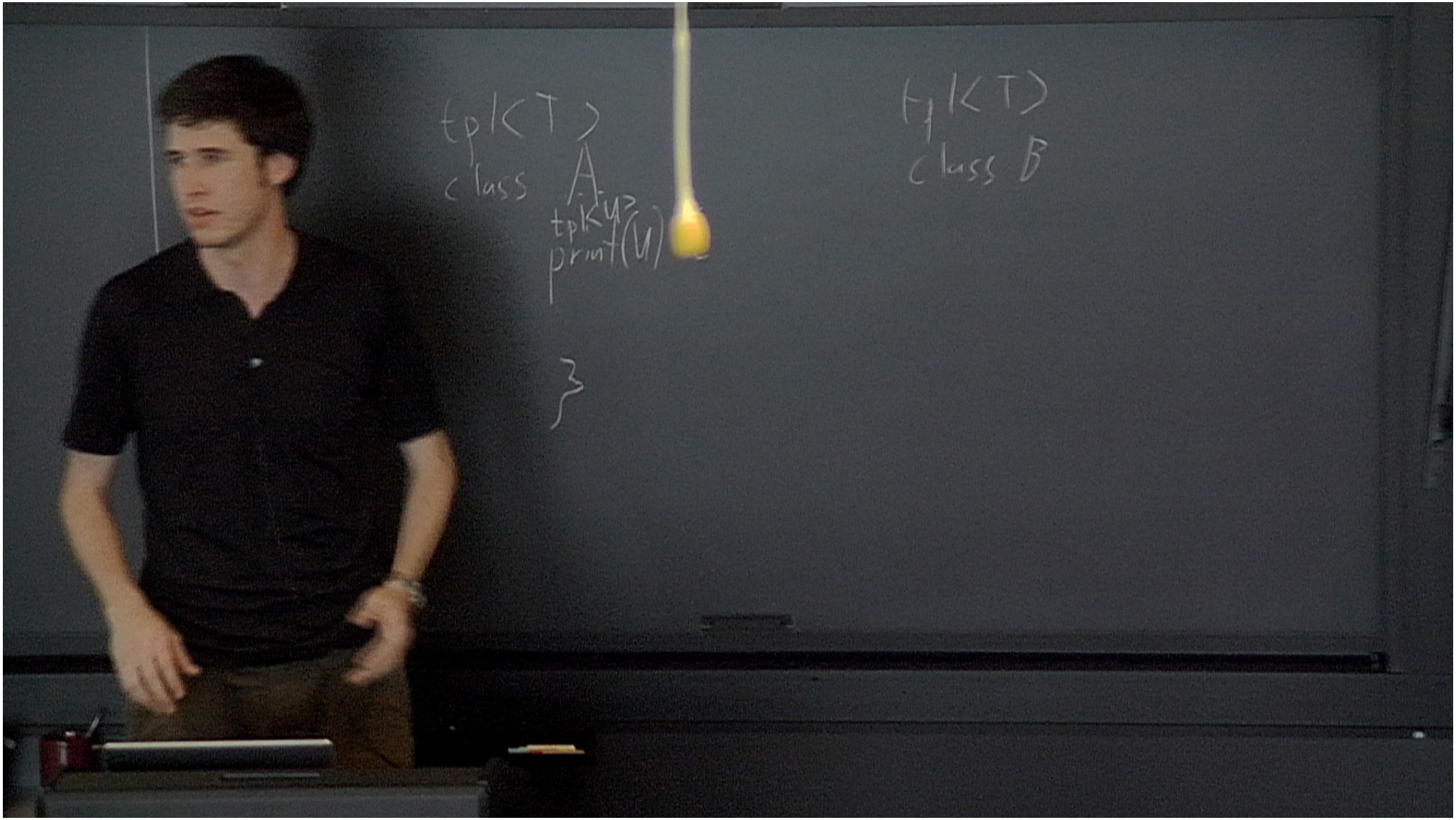

Terminal
16:15:15
press enter to continue
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
main
(program exited with code: 0)
Press return to continue
int main(int argc, char* argv[]) {
//NEEDS -std=c++11 for std::array
CLR_SCR()
PRINT_CYAN("press enter to continue")
std::vector<bool> std_vec(10);
std::array<double, 10> std_array;
WAIT_FOR_INPUT() //just hit the enter key to continue
my_vector<int> int_vec(10, 1);
PRINT_YELLOW(int_vec)
//~ my_vector<double> double_vec(int_vec);
//~ my_vector<bool> bool_vec(std_vec);
//~ double_vector = bool_vec;
//~ int_vec = std_array;
return 0;
}
Status
Compiler
Messages
Scribble
Terminal
g++-4.7 -std=c++11 -O2 -o "07_01_vector_normal" "07_01_vector_norm
Compilation finished successfully.


<type_traits> - C++ Reference - Mozilla Firefox
vector - C++ Reference
<type_traits> - C++ Reference
www.cplusplus.com/reference/type_traits/
Google
Microcontroller
Programming
Wissen
Privat
Gametrailers
Youtube
Diverses
Alte Semester
QuickOpen1
Serien
Radio
cplusplus.com
Search:
Go
Reference
<type_traits>
C++
Information
Documentation
Reference
Articles
Forum
Reference
C library:
Containers:
Input/Output:
Other:
<algorithm>
<chrono>
<codecvt>
<complex>
<exception>
<functional>
<initializer_list>
<iterator>
<limits>
<locale>
<memory>
<new>
<numeric>
<random>
<ratio>
<regex>
header
<type_traits>
type_traits
This header defines a series of classes to obtain type information on compile-tim
The header contains:
Helper classes: Standard classes to assist in creating compile-time const
Type traits: Classes to obtain characteristics of types in the form of comp
Type transformations: Classes to obtain new types by applying specific
A basic trait for types is the categories in which they can be classified. This is a c
primary categories
composite categories
fundamental
void
std::nullptr_t
integral
floating point
arithmetic
scalar
object
compound
pointer
member object pointer
member function pointer
member pointer
enum
union
class*
array


07_02_vector_concept.cpp - /home/msk/Desktop/ProgTechFinal/07_concept_programing - Geany
16:20:51
Documents
Files
template<typename U>
my_vector(U const & arg
, typename std::enable_if<std::is_same<my_vector, U>::value, int
>::type shadow = 1
): vec_(arg.vec_) {
}
template<typename U> //U needs to fulfil the vector concept
my_vector(U const & arg
, typename std::enable_if<!std::is_same<my_vector, U>::value and
!std::is_same<T, U>::value, int>::type
shadow = 1
) {
(*this) = arg;
}
//the second is_same<U, T> is needed in order not to conflict with the
ctor(size_type, T)
//why not...
//template<typename U>
//my_vector(typename boost::enable_if<..., U>::type const & arg)
//
//the problem is that the template deduction doesn't work in this case
//the compiler cannot figure out what U is
//the so called shadow parameter is just there to do SFINAE...
//...and by doing SFINAE I mean it is the part that will or will not fail
//to substitute
//assignment
template<typename U>
Status
Compiler
Messages
Scribble
Terminal
07_01_vector_normal.cpp:36:5: note: candidate expects 2 arguments, 1 provided
07_01_vector_normal.cpp:35:5: note: my_vector<T>::my_vector(const size_type&) [with T = bool; my_vector<
07_01_vector_normal.cpp:35:5: note: no known conversion for argument 1 from 'std::vector<bool>' to 'cons
07_01_vector_normal.cpp:34:5: note: my_vector<T>::my_vector() [with T = bool]
07_01_vector_normal.cpp:34:5: note: candidate expects 0 arguments, 1 provided
Compilation failed.
line: 48 / 146 col: 37 sel: 14 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: my_vector::operator =


set_property


question()


tpl<T>
class A
tpKyp
print(U)
}
tyl<T>
class B