

Title: C++ template magic - Part 2

Date: Jul 03, 2013 03:30 PM

URL: <http://pirsa.org/13070026>

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.

```
Terminal 15:33:08
Documents Files
content
1: demonstration/polymorphism
24 #include <typeinfo>
25
26 // +-----+
27 // |                               |                               |
28 // +-----+
29 int main(int argc, char* argv[]) {
30     CLR_SCR()
31     WAIT_FOR_INPUT() //just hit the enter button to continue
32
33     //-- PRINT_GREEN(" name:          Mario Könz")
34     //-- WAIT_FOR_INPUT()
35     //-- PRINT_GREEN(" study:        interdisciplinary science ETH Zuerich")
36     //-- WAIT_FOR_INPUT()
37     //-- PRINT_GREEN(" main subject:  physics and computational science")
38     //-- WAIT_FOR_INPUT()
39     //-- PRINT_GREEN(" c++ experience: 3 semesters programming techniques with
Matthias Troyer")
40     //-- PRINT_GREEN("                               worked 2 years on the ALPS library")
41     //-- WAIT_FOR_INPUT()
42     //-- PRINT_GREEN(" why am I here:  writing my master-thesis with Roger
Melko")
43     //-- 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")
}

Status g++-4.7 -std=c++11 -O2 -o "Introduction" "introduction.cpp" (in directory: /home/msk/Desktop/ProgTechFin
Compiler Compilation finished successfully.
Messages
Scribble
Terminal
line: 43 / 77 col: 24 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main
```



```
Terminal 15:33:17
Documents Files
content
1: demonstration/polymorphism
2: template motivation
3: template basics
24 #include <typeinfo>
25
26 // +-----+
27 // |                               |                               |
28 // +-----+
29 int main(int argc, char* argv[]) {
30     CLR_SCR();
31     WAIT_FOR_INPUT() //just hit the enter button to continue
32
33     PRINT_GREEN("  name:                Mario K\"onz")
34     //~ WAIT_FOR_INPUT()
35     //~ PRINT_GREEN("  study:            interdisciplinary science ETH Zuerich")
36     //~ WAIT_FOR_INPUT()
37     //~ PRINT_GREEN("  main subject:  physics and computational science")
38     //~ WAIT_FOR_INPUT()
39     //~ PRINT_GREEN("  c++ experience: 3 semesters programming techniques with
Matthias Troyer")
40     //~ PRINT_GREEN("  work experience: worked 2 years on the ALPS library")
41     //~ WAIT_FOR_INPUT()
42     //~ PRINT_GREEN("  why am I here:  writing my master-thesis with Roger
Melko")
43     //~ 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")

Status g++-4.7 -std=c++11 -O2 -o "Introduction" "introduction.cpp" (in directory: /home/msk/Desktop/ProgTechFin
Compiler Compilation finished successfully.
Messages
Scribble
Terminal
line: 43 / 77 col: 24 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main
```



```
Terminal 15:34:00
Documents Files
content
1: demonstration/polymorphism
2: template motivation
3: template basics
4: mean trait
24 #include <typeinfo>
25
26 // +-----+
27 // |                               | main |
28 // +-----+
29 int main(int argc, char* argv[]) {
30     CLR_SCR();
31     WAIT_FOR_INPUT() //just hit the enter button to continue
32
33     PRINT_GREEN(" name:           Mario K\"onz")
34     //~ WAIT_FOR_INPUT()
35     //~ PRINT_GREEN(" study:       interdisciplinary science ETH Zuerich")
36     //~ WAIT_FOR_INPUT()
37     //~ PRINT_GREEN(" main subject: physics and computational science")
38     //~ WAIT_FOR_INPUT()
39     //~ PRINT_GREEN(" c++ experience: 3 semesters programming techniques with
Matthias Troyer")
40     //~ PRINT_GREEN("                               worked 2 years on the ALPS library")
41     //~ WAIT_FOR_INPUT()
42     //~ PRINT_GREEN(" why am I here: writing my master-thesis with Roger
Melko")
43     //~ 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")

Status g++-4.7 -std=c++11 -O2 -o "Introduction" "introduction.cpp" (in directory: /home/msk/Desktop/ProgTechFin
Compiler Compilation finished successfully.
Messages
Scribble
Terminal
line: 43 / 77 col: 24 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main
```




Terminal

```
24 #include <typeinfo>
25
26 // +-----+
27 // |                               main                               |
28 // +-----+
29 int main(int argc, char* argv[]) {
30     CLR_SCR()
31     WAIT_FOR_INPUT() //just hit the enter button to continue
32
33     //-- PRINT_GREEN(" name:           Mario Könz")
34     //-- WAIT_FOR_INPUT()
35     //-- PRINT_GREEN(" study:         interdisciplinary science ETH Zuerich")
36     //-- WAIT_FOR_INPUT()
37     //-- PRINT_GREEN(" main subject:  physics and computational science")
38     //-- WAIT_FOR_INPUT()
39     //-- PRINT_GREEN(" c++ experience: 3 semesters programming techniques with
Matthias Troyer")
40     //-- PRINT_GREEN("                               worked 2 years on the ALPS library")
41     //-- WAIT_FOR_INPUT()
42     //-- PRINT_GREEN(" why am I here:  writing my master-thesis with Roger
Melko")
43     //-- 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")
```

Status g++-4.7 -std=c++11 -O2 -o "Introduction" "introduction.cpp" (in director  
Compiler Compilation finished successfully.  
Messages  
Scribble  
Terminal

line: 43 / 77 col: 24 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main





Terminal


Documents Files

majority of the examples works without c++0x  
those few examples that need c++0x can also be realised with just boost  
all examples are available on GitHub  
[https://github.com/mskoenz/cpp\\_template\\_examples](https://github.com/mskoenz/cpp_template_examples)

```
24 #include <typeinfo>
25
26 // -----+-----
27 // |                               |                               |
28 // |                               |                               |
29 int main(int argc, char* argv[]) {
30     CLR_SCR();
31     WAIT_FOR_INPUT(); //just hit the enter button to continue
32
33     //-- PRINT_GREEN("  name:           Mario Könz")
34     //-- WAIT_FOR_INPUT()
35     //-- PRINT_GREEN("  study:         interdisciplinary science ETH Zuerich")
36     //-- WAIT_FOR_INPUT()
37     //-- PRINT_GREEN("  main subject:  physics and computational science")
38     //-- WAIT_FOR_INPUT()
39     //-- PRINT_GREEN("  c++ experience: 3 semesters programming techniques with
Matthias Troyer")
40     //-- PRINT_GREEN("                               worked 2 years on the ALPS library")
41     //-- WAIT_FOR_INPUT()
42     //-- PRINT_GREEN("  why am I here:  writing my master-thesis with Roger
Melko")
43     //-- 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")
```

Status g++-4.7 -std=c++11 -O2 -o "Introduction" "introduction.cpp" (in director  
Compiler Compilation finished successfully.  
Messages  
Scribble  
Terminal

line: 43 / 77 col: 24 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main



```
08_01_typedef.cpp - /home/msk/Desktop/ProgTechFinal/08_summary - Geany
1 // Author: Mario S. Könz <mskoenz@gmx.net>
2 // Date: 28.06.2013 22:11:46 EDT
3 // File: 08_01_typedef.cpp
4
5 #include <iostream>
6 #include <vector>
7 //===== typedefs =====
8
9 //use typedef to call types by another name like a reference
10
11 //int is way to long, we need an abbreviation
12 typedef int I;
13
14 //shorter name
15 typedef std::vector<std::vector<int>> int_array_type;
16 //use boost::multi_array instead of vector<vector<T>> for critical code
17 //vector<vector<T>> can have bad locality
18
19 //c++11
20 template<typename T>
21 using array_type = std::vector<std::vector<T>>; //new style "typedef"
22
23 // +-----+
24 // | main |
25 // +-----+
26 int main(int argc, char* argv[]) {
27     //now you can use the new name like an the old, long compicated one
28
29     I nr1 = 1;
```

Status g++-4.7 -std=c++11 -O2 -o "introduction" "introduction.cpp" (in directory: /home/msk/Desktop/ProgTechFinal)
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 12 / 58 col: 14 sel: 14 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown



```
*08_01_typedef.cpp - /home/msk/Desktop/ProgTechFinal/08_summary - Geany
9 //use typedef to call types by another name like a reference
10
11 //int is way to long, we need an abreviation
12 typedef int I;
13
14 //shorter name
15 typedef std::vector<std::vector<int>> int_array_type;
16 //use boost::multi_array instead of vector<vector<T>> for critical code
17 //vector<vector<T>> can have bad locality
18
19 //c++11
20 template<typename T>
21 using array_type = std::vector<std::vector<T>>; //new style "typedef"
22
23 // +-----+
24 // |                    main                    |
25 // +-----+
26 int main(int argc, char* argv[]) {
27     //now you can use the new name like an the old, long compicated one
28
29     I nr1 = 1;
30     I nr2 = 1;
31     int_array_type m1;
32     array_type<I> m2;
33
34     //everything you can do with int, you can now also do with I
35
36     std::cout << nr1 << std::endl;
37     std::cout << nr2 << std::endl;

```

Status g++-4.7 -std=c++11 -O2 -o "introduction" "introduction.cpp" (in directory: /home/msk/Desktop/ProgTechFinal)
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 36 / 58 col: 20 sel: 0 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main



```
*08_01_typedef.cpp - /home/msk/Desktop/ProgTechFinal/08_summary - Geany
Documents Files
~/Deskt...e_basics
  03_03_...on.cpp
~/Deskt...classes
  03_01_...er.hpp
  03_01_...impl.cpp
  03_01_...es.cpp
~/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_...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
29  I nr1 = 1;
30  I nr2 = 1;
31  int_array_type m1;
32  array_type<I> m2;
33
34  //everything you can do with int, you can now also do with I
35
36  std::cout << nr1 << std::endl;
37  std::cout << nr2 << std::endl;
38
39  //typedefs are normaly done inside a class/struct not in a function
40  //some compilers will warn you about typedefs outside a class/struct
41
42  //make code more flexible and readable
43  typedef int spin_type;
44
45  //could be anything
46  int a1;
47
48  //long names are cumbersome
49  int spin_a1;
50
51  //now the name doesn't need to explain what the var does
52  spin_type a2;
53
54
55  a2+a1
56  //better having a longer type than name since you need the type normally
    less often

Status g++-4.7 -std=c++11 -O2 -o "introduction" "introduction.cpp" (in directory: /home/msk/Desktop/ProgTechFinal/08_summary)
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 55 / 60 col: 4 sel: 5 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main
```

\*08\_02\_typename.cpp - /home/msk/Desktop/ProgTechFinal/08\_summary - Geany

```

22
23 //typename is used for two purposes:
24
25 //first: it's used when declaring templates and just says that whatever
26 //follows (T) is a type.
27 template<typename T>
28 struct my_template {
29     typedef T type;
30 };
31
32 //second: to extract types that are inside a template struct/class and
33 //depend somehow on the template parameter (dependent scope)
34 //basically the following pattern ....<T>...::some_type
35 template<typename T>
36 struct my_second_template {
37     typedef typename my_template<T>::type second_type;
38     //i need typename here because int_type is inside a struct that depends
39     //on the template parameter T. Just remove typename to see the error
40 };
41
42 // +-----+
43 // |                               main                               |
44 // +-----+
45 int main(int argc, char* argv[]) {
46
47     //no need for typename here, since there is no unknown T-dependency
48     //T==int an all is good :-)
49     my_template<int>::type nr1 = 0;
50

```

Status **g++-4.7 -std=c++11 -O2 -o "introduction" "introduction.cpp" (in director**

Compiler **Compilation finished successfully.**

Messages

Scribble

Terminal

line: 29 / 60 col: 19 sel: 8 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown



\*08\_02\_typeofname.cpp - /home/msk/Desktop/ProgTechFinal/08\_summary - Geany

```

28 struct my_template {
29     typedef T type;
30 };
31
32 //second: to extract types that are inside a template struct/class and
33 //depend somehow on the template parameter (dependent scope)
34 //basically the following pattern ....<T>...::some_type
35 template<typename T>
36 struct my_second_template {
37     typedef typename my_template<T>::type second_type;
38     //i need typename here because int_type is inside a struct that depends
39     //on the template parameter T. Just remove typename to see the error
40 };
41
42 // +-----+
43 // |           I           main           |
44 // +-----+
45 int main(int argc, char* argv[]) {
46
47     //no need for typename here, since there is no unknown T-dependency
48     //T==int an all is good :- )
49     my_template<int>::type nr1 = 0;
50
51     //same here
52     my_second_template<int>::second_type nr2 = 0;
53
54     PRINT_GREEN(nr1)
55     PRINT_GREEN(nr2)
56

```

Status `g++-4.7 -std=c++11 -O2 -o "introduction" "introduction.cpp" (in director`

Compiler `Compilation finished successfully.`

Messages

Scribble

Terminal

line: 49 / 60 col: 26 sel: 22 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main





\*08\_03\_typeInfo\_sizeof.cpp - /home/msk/Desktop/ProgTechFinal/08\_summary - Geany

```

32 // +-----+
33 // |                               |
34 // +-----+
35 int main(int argc, char* argv[]) {
36
37     //sizeof() tells you how many bytes a type has. It's a one of a kind
38     //function since it transforms types into integers.
39     //But you can also insert actual objects like 1
40     //works during compiletime
41     //sizeof: types/objects --> int
42     PRINT_GREEN( sizeof(int) )
43     PRINT_GREEN( sizeof(1) )
44
45     //typeid() from the <typeinfo> header works similarly. You can apply
46     //it to a type or object and it will return a type_info-object with
47     //information
48     //about the type
49     PRINT_YELLOW( typeid(int).name() )
50     PRINT_YELLOW( typeid(1).name() )
51
52     //the names are mostly short, but for custom types it will use
53     //scope::class_name as name (+ some numbers and markers)
54     PRINT_RED( typeid(example::my_struct<int>).name() )
55
56     //you can "store" the type during runtime using typeid()
57
58     return 0;
59 }

```

Status `g++-4.7 -std=c++11 -O2 -o "08_02_typename" "08_02_typename.cpp" (`

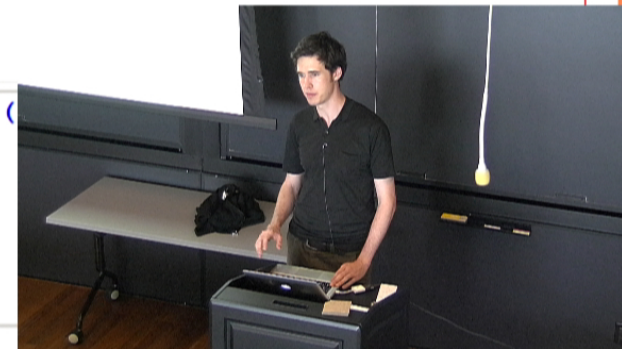
Compiler `Compilation finished successfully.`

Messages

Scribble

Terminal

line: 42 / 61 col: 17 sel: 0 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main



```
08_03_typeinfo_sizeof.cpp - /home/msk/Desktop/ProgTechFinal/08_summary - Geany
8 #define TYPE(T) typeid(T).name()
9 #define CLR_SCR() std::cout << "\033[2J\033[100A";
10 #define NEW_LINE() std::cout << std::endl;
11 #define WAIT_FOR_INPUT() while(std::cin.getcount() == 0) std::cin.get(); std::cin
    .get();
12 #define ASSERT_MSG(cond, msg) if(cond) {PRINT_RED(msg); throw std::
    runtime_error("error");}
13 #define PRINT_NAMED(x) std::cout << #x << " = " << x << std::endl; // #x
    changes the variable name into a string "x"
14 #define PRINT_RED(x) std::cout << "\033[1;31m" << x << "\033[0m" << std::endl;
15 #define PRINT_BLUE(x) std::cout << "\033[1;34m" << x << "\033[0m" << std::endl;
16 #define PRINT_CYAN(x) std::cout << "\033[1;36m" << x << "\033[0m" << std::endl;
17 #define PRINT_GREEN(x) std::cout << "\033[1;32m" << x << "\033[0m" << std::endl;
18 #define PRINT_YELLOW(x) std::cout << "\033[1;33m" << x << "\033[0m" << std::endl;
19 #define PRINT_MAGENTA(x) std::cout << "\033[1;35m" << x << "\033[0m" << std:::
    endl;
20 //===== includes =====
21 #include <iostream>
22 #include <typeinfo>
23
24 namespace example {
25     template<typename T>
26     struct my_struct {
27
28     };
29
30 } //end namespace example
31
32 // +-----
```

Status **g++-4.7 -std=c++11 -O2 -o "08\_03\_typeinfo\_sizeof" "08\_03\_typeinfo\_sizeof.cpp" -c++**

Compiler **Compilation finished successfully.**

Messages

Scribble

Terminal

line: 22 / 61 col: 19 sel: 19 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown





08\_03\_typeinfo\_sizeof.cpp - /home/msk/Desktop/ProgTechFinal/08\_summary - Geany

```
32 // +-----+
33 // |                               |
34 // +-----+
35 int main(int argc, char* argv[]) {
36
37     //sizeof() tells you how many bytes a type has. It's a one of a kind
38     //function since it transforms types into integers.
39     //But you can also insert actual objects like 1
40     //works during compiletime
41     //sizeof: types/objects --> int
42     PRINT_GREEN( sizeof(int) )
43     PRINT_GREEN( sizeof(1) )
44
45     //typeid() from the <typeinfo> header works similarly. You can apply
46     //it to a type or object and it will return a type_info-object with
47     //information
48     //about the type
49     PRINT_YELLOW( typeid(int).name() )
50     PRINT_YELLOW( typeid(1).name() )
51
52     //the names are mostly short, but for custom types it will use
53     //scope::class_name as name (+ some numbers and markers)
54     PRINT_RED( typeid(example::my_struct<int>).name() |)
55
56     //you can "store" the type during runtime using typeid()
57
58     return 0;
59 }
```

Status g++-4.7 -std=c++11 -O2 -o "08\_03\_typeinfo\_sizeof" "08\_03\_typeinfo\_sizeof.cpp" -lstdc++


Compiler Compilation finished successfully.

Messages

Scribble

Terminal

line: 54 / 61 col: 54 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main





Terminal File Edit View Search Terminal Help 15:44:01

```
32 // +-----+
33 // |                               |
34 // +-----+
35 int main(int argc, char* argv[]) {
36
37     //sizeof: types/objects --> int
38     //function since it transforms types into integers.
39     //But you can also insert actual objects like 1
40     //works during compiletime
41     //sizeof: types/objects --> int
42     printf("sizeof(int) = %d\n", sizeof(int) );
43     printf("sizeof(1) = %d\n", sizeof(1) );
44
45     //typeid() from the <typeinfo> header works similarly. You can apply
46     //it to a type or object and it will return a type_info-object with
47     //information
48     //about the type
49     PRINT_YELLOW( typeid(int).name() )
50     PRINT_YELLOW( typeid(1).name() )
51
52     //the names are mostly short, but for custom types it will use
53     //scope::class_name as name (+ some numbers and markers)
54     PRINT_RED( typeid(example::my_struct<int>).name() )
55
56     //you can "store" the type during runtime using typeid()
57
58     return 0;
59 }
```

(program exited with code: 0)  
Press return to continue

Status g++-4.7 -std=c++11 -O2 -o "08\_03\_typeinfo\_sizeof" "08\_03\_typeinfo\_sizeof.cpp" -std=c++11  
Compiler Compilation finished successfully.  
Messages  
Scribble  
Terminal

line: 52 / 61 col: 0 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main



08\_04\_asserts.cpp - /home/msk/Desktop/ProgTechFinal/08\_summary - Geany

```

26  double res;
27  PRINT_GREEN("please enter a number: ")
28  std::cin >> res;
29  return res;
30 }
31
32 // +-----+
33 // |                               main                               |
34 // +-----+
35 int main(int argc, char* argv[]) {
36     I
37     //asserts from the <assert.h> header are very useful to make sure
38     //the state of your program is actually the way you think it is
39
40     //p could be a probability
41     double p = 0;
42
43     //do whatever
44     p = read_fct();
45
46     //since I know p is a probability, it should be between 0 and 1.
47     //an assert can check that and you don't lose performance because
48     //all asserts vanish if compiled with -DNDEBUG
49     assert(p >= 0 and p <= 1);
50
51     //change the return in fct to see the fail
52     //leave the "fail" and compile with -DNDEBUG to disable the assert
53
54     //IMPORTANT: if you turn off the assertion, you h

```

Status `g++-4.7 -std=c++11 -O2 -o "08_04_asserts" "08_04_asserts.cpp" (in dire`

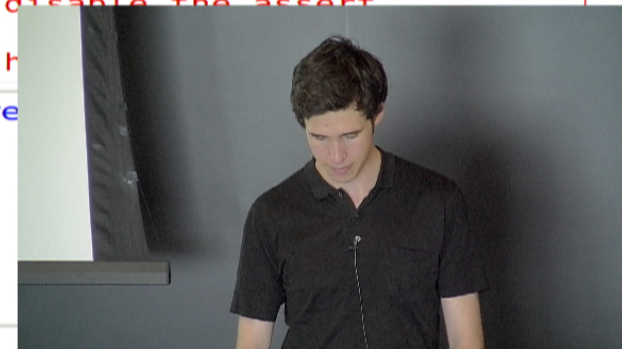
Compiler

Messages

Scribble

Terminal

line: 46 / 62 col: 19 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main






Terminal

```
please enter a number:
12
08_04_asserts: 08_04_asserts.cpp:49: int main(int, char**): Assertion `p >= 0 and p <= 1' failed.
Aborted (core dumped)
-----+-----
32 // |                                     |
33 // |                                     |
34 // +-----+-----+
-----+-----
35 int main(int argc, char* argv[]) {
36     //the state of your program is actually the way you think it is
37
38     //p could be a probability
39     double p = 0;
40
41     //do whatever
42     p = read_fct();
43
44     //since I know p is a probability, it should be between 0 and 1.
45     //an assert can check that and you don't lose performance because
46     //all asserts vanish if compiled with -DNDEBUG
47     assert(p >= 0 and p <= 1);
48
49     //change the return in fct to see the fail
50     //leave the "fail" and compile with -DNDEBUG to disable the assert
51
52     //IMPORTANT: if you turn off the assertion, you h
53
54
(program exited with code: 134)
Press return to continue from the <assert.h> header are very useful to make sure
```

Status g++-4.7 -std=c++11 -O2 -o "08\_04\_asserts" "08\_04\_asserts.cpp" (in dire  
Compiler Compilation finished successfully.  
Messages  
Scribble  
Terminal

line: 46 / 62 col: 19 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main



08\_04\_asserts.cpp - /home/msk/Desktop/ProgTechFinal/08\_summary - Geany

```
34 // +-----+
35 int main(int argc, char* argv[]) {
36
37     //asserts from the <assert.h> header are very useful to make sure
38     //the state of your program is actually the way you think it is
39
40     //p could be a probability
41     double p = 0;
42
43     //do whatever
44     p = read_fct();
45
46     //since I know p is a probability, it should be between 0 and 1.
47     //an assert can check that and you don't lose performance because
48     //all asserts vanish if compiled with -DNDEBUG
49     assert(p >= 0 and p <= 1);
50
51     //change the return in fct to see the fail
52     //leave the "fail" and compile with -DNDEBUG to disable the assert
53
54     //IMPORTANT: if you turn off the assertion, you have no security anymore
55     //for critical checks use if and throw an error like this:
56
57     //~ if(p < 0 or p > 1)
58         //~ throw std::runtime_error("p must be between 0 and 1");
59
60     return 0;
61 }
62
```

Status g++-4.7 -DNDEBUG -std=c++11 -O2 -o "08\_04\_asserts" "08\_04\_asserts.  
Compiler Compilation finished successfully.  
Messages  
Scribble  
Terminal

line: 50 / 62 col: 4 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main





```
*08_04_asserts.cpp - /home/msk/Desktop/ProgTechFinal/08_summary - Geany
Documents Files
~/Deskt...e_basics
  03_03_...on.cpp
~/Deskt...classes
  03_01_...er.hpp
  03_01_...impl.cpp
  03_01_...es.cpp
~/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_...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
34 // +-----+
35 int main(int argc, char* argv[]) {
36
37     //asserts from the <assert.h> header are very useful to make sure
38     //the state of your program is actually the way you think it is
39
40     //p could be a probability
41     double p = 0;
42
43     //do whatever
44     p = read_fct();
45
46     //since I know p is a probability, it should be between 0 and 1.
47     //an assert can check that and you don't loose performance because
48     //all asserts vanish if compiled with -DNDEBUG
49     assert(p >= 0 and p <= 1);
50
51     //change the return in fct to see the fail
52     //leave the "fail" and compile with -DNDEBUG to disable the assert
53
54     //IMPORTANT: if you turn off the assertion, you have no security anymore
55     //for critical checks use if and throw an error like this:
56
57     if(p < 0 or p > 1)
58         throw std::runtime_error("p must be between 0 and 1");
59
60     return 0;
61 }
62

Status g++-4.7 -DNDEBUG -std=c++11 -O2 -o "08_04_asserts" "08_04_asserts.cpp" (in directory: /home/msk/Desk
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 57 / 62 col: 22 sel: 18 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main
```

08\_06\_class\_template.cpp - /home/msk/Desktop/ProgTechFinal/08\_summary - Geany

1 // Author: Mario S. Könz <mskoenz@gmx.net>  
2 // Date: 28.06.2013 22:59:19 EDT  
3 // File: 08\_06\_class\_template.cpp  
4  
5 #include <iostream>  
6 #include <assert.h>  
7  
8 //same as in example 02\_03\_template\_way.cpp  
9 template<typename T, uint32\_t N>  
10 class template\_class {  
11 public:  
12 typedef uint32\_t size\_type; //use typedefs to make later changes easy  
13 T & operator[](size\_type const & index) {  
14 assert(index >= 0 and index < N);  
15 return array\_[index];  
16 }  
17 T const & operator[](size\_type const & index) const {  
18 //you almost ever need two access operators, one const and the other normal  
19 assert(index >= 0 and index < N);  
20 return array\_[index];  
21 }  
22 size\_type size() const { //can also be done as a static, but is inlined  
23 anyway  
24 return N;  
25 }  
26 private:  
27 T array\_[N];  
28 };

Status g++-4.7 -std=c++11 -O2 -o "08\_05\_static\_asserts" "08\_05\_static\_asserts.cpp" (in directory: /home/msk/Desl  
Compiler 08\_05\_static\_asserts.cpp: In function 'int main(int, char\*\*)':  
Messages 08\_05\_static\_asserts.cpp:41:5: error: static assertion failed: Error: some nice message  
Scribble Compilation failed.  
Terminal

line: 14 / 56 col: 41 sel: 33 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: template\_class::operator []



- Documents
- Files
- ~/Deskt...e\_basics
  - 03\_03\_...on.cpp
- ~/Deskt...classes
  - 03\_01\_...er.hpp
  - 03\_01\_...impl.cpp
  - 03\_01\_...es.cpp
- ~/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\_...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

```
23         anyway
24         return N;
25     }
26 private:
27     T array_[N];
28 };
29 //demonstration of default arguments
30 template< typename T = int
31           , typename U = int
32           , bool b = false>
33 struct default_template {
34     //does nothing...
35 };
36
37 // +-----+
38 // |                               |
39 // +-----+
40 int main(int argc, char* argv[]) {
41     //the template parameters are placed in the <> brakets right after
42     //the name
43     template_class<int, 5> a; //right
44     //~ template_class a<int, 5>; //wrong
45
46     //you can use default args for templates, but you still need <>
47     default_template<> all_default; //works
48     //~ default_template all_default_wrong; //doesn't
49
50     default_template<void> partial_default; //void is usable as a type
```

Status: g++-4.7 -std=c++11 -O2 -o "08\_05\_static\_asserts" "08\_05\_static\_asserts.cpp" (in directory: /home/msk/Des)

Compiler: 08\_05\_static\_asserts.cpp: In function 'int main(int, char\*\*)':

Messages: 08\_05\_static\_asserts.cpp:41:5: error: static assertion failed: Error: some nice message

Scribble: Compilation failed.

Terminal:

line: 43 / 56 col: 4 sel: 22 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main

```
08_07_function_template.cpp - /home/msk/Desktop/ProgTechFinal/08_summary - Geany
Documents Files
~/Deskt...e_basics
  03_03_...on.cpp
~/Deskt...classes
  03_01_...er.hpp
  03_01_...impl.cpp
  03_01_...es.cpp
~/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_...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
37 // |                                     main                                     |
38 // +-----+-----+-----+-----+-----+-----+-----+-----+-----+
39 int main(int argc, char* argv[]) {
40     PRINT_CYAN("press enter to continue")
41
42
43     int i = 10;
44     double d = 3.14;
45     std::string s = "hello world";
46
47     WAIT_FOR_INPUT()//just hit the enter key to continue
48     print(i); //deduce the type. It's possible since the first arg contains the T
49
50     WAIT_FOR_INPUT()
51     print(s);
52
53     WAIT_FOR_INPUT()
54     print(d);
55
56     WAIT_FOR_INPUT()
57     print<int>(d); //call with given type
58
59     WAIT_FOR_INPUT()
60     print( mean<double>(1, 2) ); //we have to pass T (the compiler
61     //cannot guess what T is), but U (=int) is deducible again
62
63     return 0;
64 }
65

Status g++-4.7 -std=c++11 -O2 -o "08_07_function_template" "08_07_function_template.cpp" (in directory: /home
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 60 / 66 col: 16 sel: 6 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main
```

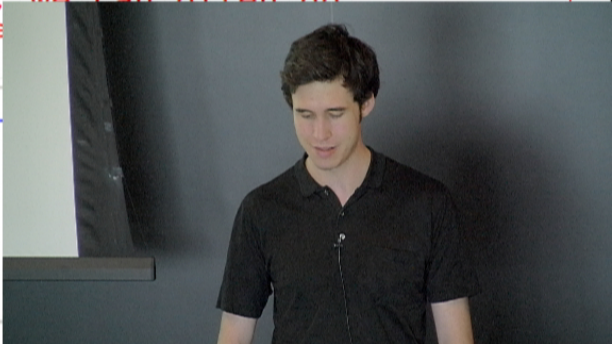


\*08\_10\_deductio...constructor.cpp - /home/msk/Desktop/ProgTechFinal/08\_summary - Geany

```
45  std::pair<int, int> p1(1, 2);
46  //~ std::pair p1(1, 2) //doesn't work, but why exactly?
47  //~functions can deduce the types...
48  p1 = make_pair(1, 1);
49
50  print_pair(p1); //deduction is no problem
51
52  WAIT_FOR_INPUT()
53  print_pair(std::pair<int, int>(3, 4)); //quite cumbersome...
54
55  WAIT_FOR_INPUT()
56  print_pair(make_pair(5, 6)); //deduction via function
57
58  WAIT_FOR_INPUT()
59  //~the std already has make_pair
60  print_pair(std::make_pair(7, 8)); //deduction via function
61
62  //~the reason for not having constructor deduction:
63  //~The C++ Programming Language, Third Edition by Bjarne Stroustr
64  /*
65
66  Note that class template parameters are never deduced. The reason is
67  that the flexibility provided by several constructors for a class
68  would make such deduction impossible in many cases and obscure in
69  many more. Specialization provides a mechanism for implicitly
70  choosing between different implementations of a class (§13.5).
71  If we need to create an object of a deduced type, we can often do
72  that by calling a function to do the creation; see
73
```

Status g++-4.7 -std=c++11 -O2 -o "08\_07\_function\_template" "08\_07\_function\_...  
Compiler Compilation finished successfully.  
Messages  
Scribble  
Terminal

line: 53 / 86 col: 15 sel: 25 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main




08\_10\_deductio...constructor.cpp - /home/msk/Desktop/ProgTechFinal/08\_summary - Geany

```
59  WAIT_FOR_INPUT()
60  //the std already has make_pair
61  print_pair(std::make_pair(7, 8)); //deduction via function
62
63  //the reason for not having constructor deduction:
64  //The C++ Programming Language, Third Edition by Bjarne Stroustr
65  /*
66
67  Note that class template parameters are never deduced. The reason is
68  that the flexibility provided by several constructors for a class
69  would make such deduction impossible in many cases and obscure in
70  many more. Specialization provides a mechanism for implicitly
71  choosing between different implementations of a class (§13.5).
72  If we need to create an object of a deduced type, we can often do
73  that by calling a function to do the creation; see make_pair()
74
75  */
76
77  WAIT_FOR_INPUT()
78  //Why you want c++11:
79
80  //auto deduces the type that is returned from a function
81  //we never have to write std::pair<double ,std::string>
82  auto p2 = 4;
83  //~ print_pair(p2);
84
85  return 0;
86 }
87
```

Status g++-4.7 -std=c++11 -O2 -o "08\_10\_deduction\_constructor" "08\_10\_dedu  
Compiler Compilation finished successfully.  
Messages  
Scribble  
Terminal

line: 83 / 87 col: 20 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main





```
04_01_functions.cpp - /home/msk/Desktop/ProgTechFinal/04_mean_trait - Geany
Documents Files
~/Deskt...e_basics
  03_03_...on.cpp
~/Deskt...classes
  03_01_...er.hpp
  03_01_...impl.cpp
  03_01_...es.cpp
~/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_...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
  http://www.cplusplus.com/forum/unices/36461/ for details
21
22 //===== includes =====
23 #include <iostream>
24
25 //===== mean function =====
26 inline double mean(int const & a, int const & b) {
27     PRINT_RED("mean fct version int with arg " << a << " and " << b);
28     return (a + b) / 2.0;
29 }
30
31
32 //~ inline double mean(double const & a, double const & b) {
33 //~ PRINT_GREEN("mean fct version double with arg " << a << " and " << b);
34 //~ return (a + b) / 2.0;
35 //~ }
36
37
38 //~ inline long double mean(long double const & a, long double const & b) {
39 //~ PRINT_MAGENTA("mean fct version long double with arg " << a << " and "
40 //~ << b);
41 //~ return (a + b) / 2.0;
42 //~ }
43
44 // +-----+
45 // |                               main                               |
46 // +-----+
47 int main(int argc, char* argv[]) {
Status g++-4.7 -std=c++11 -O2 -o "08_10_deduction_constructor" "08_10_deduction_constructor.cpp" (in director
Compiler Compilation finished successfully.
Messages
Scribble
Terminal
line: 26 / 69 col: 6 sel: 7 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown
```

04\_01\_functions.cpp - /home/msk/Desktop/ProgTechFinal/04\_mean\_trait - Geany

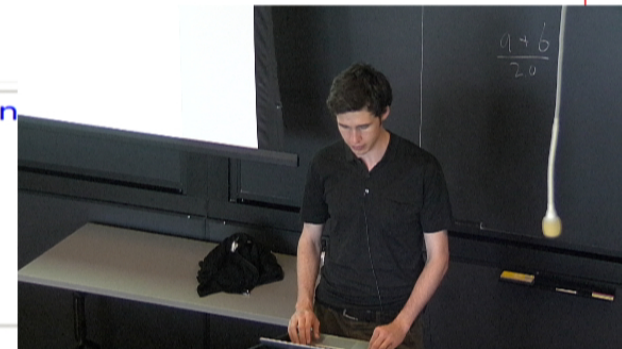
```

32 //~ inline double mean(double const & a, double const & b) {
33 //~ PRINT_GREEN("mean fct version double with arg " << a << " and " << b);
34 //~ return (a + b) / 2.0;
35 //~ }
36
37
38 //~ inline long double mean(long double const & a, long double const & b) {
39 //~ PRINT_MAGENTA("mean fct version long double with arg " << a << " and "
40 //~ return (a + b) / 2.0;
41 //~ }
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 PRINT_NAMED( mean(1, 2) )
53
54 WAIT_FOR_INPUT()
55 PRINT_NAMED( mean(1.8, 2.0) )
56
57 WAIT_FOR_INPUT()
58 //~ PRINT_NAMED( mean(1.8l, 2.0l) )
59

```

Status **g++-4.7 -std=c++11 -O2 -o "04\_01\_functions" "04\_01\_functions.cpp" (in**  
**Compilation finished successfully.**

line: 32 / 69 col: 29 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown





04\_01\_functions.cpp - /home/msk/Desktop/ProgTechFinal/04\_mean\_trait - Geany

```

39     //~ PRINT_MAGENTA("mean fct version long double with arg " << a << " and "
40     << b);
41     //~ return (a + b) / 2.0;
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         PRINT_NAMED( mean(1, 2) )
53
54         WAIT_FOR_INPUT()
55         PRINT_NAMED( mean(1.8, 2.0) )
56
57         WAIT_FOR_INPUT()
58         PRINT_NAMED( mean(1.8l, 2.0l) )
59
60         WAIT_FOR_INPUT()
61         //~ PRINT_RED("        sizeof(int) = " << sizeof(int) << " bytes")
62         //~ PRINT_GREEN("        sizeof(double) = " << sizeof(double) << " bytes")
63         //~ PRINT_MAGENTA("sizeof(long double) = " << sizeof(long double) << "
64         //we cannot just use the mean version double for all types since it
65         //is not efficient for smaller types and not accurate for larger types

```

Status  
Compiler  
Messages  
Scribble  
Terminal

```

04_01_functions.cpp: In function 'int main(int, char**)':
04_01_functions.cpp:58:5: error: call of overloaded 'mean(long double, long double)' is ambiguous
04_01_functions.cpp:58:5: note: candidates are:
04_01_functions.cpp:26:15: note: double mean(const int&, const int&)
04_01_functions.cpp:32:15: note: double mean(const double&, const double&)
Compilation failed.

```

line: 62 / 69 col: 66 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main

```

Terminal
press enter to continue
40 return (a + b) / 2.0;
41 //~ }
42
43
44
45
46 // -----+
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68 }
}

mean fct version double with arg 1 and 2
mean(1, 2) = 1.5 |
main |
-----+

mean fct version double with arg 1.8 and 2
mean(1.8, 2.0) = 1.9
PRINT_CYAN("press enter to continue")

mean fct version double with arg 1.8 and 2
mean(1.8l, 2.0l) = 1.9
PRINT_RED("press enter key to continue")
PRINT_NAMED( mean(1, 2) )
PRINT_NAMED( mean(1.8l, 2.0l) )

sizeof(int) = 4 bytes
sizeof(double) = 8 bytes
sizeof(long double) = 16 bytes

----- PRINT_RED(" sizeof(int) = " << sizeof(int) << " bytes")
(program exited with code: 0)
Press return to continue
PRINT_GREEN(" sizeof(double) = " << sizeof(double) << " bytes")
PRINT_MAGENTA( sizeof(long double) = " << sizeof(long double) << " bytes")
//we cannot just use the mean version double for all types since it
//is not efficient for smaller types and not accurate for larger types

return 0;
}

Status g++-4.7 -std=c++11 -O2 -o "04_01_functions" "04_01_functions.cpp" (in directory: /home/msk/Desktop/Pro
Compiler Compilation finished successfully.
Messages
Scribble
Terminal
line: 63 / 69 col: 71 sel: 212 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main

```



```
04_01_functions.cpp - /home/msk/Desktop/ProgTechFinal/04_mean_trait - Geany
Documents Files
~/Deskt...e_basics
  03_03_...on.cpp
~/Deskt...classes
  03_01_...er.hpp
  03_01_...impl.cpp
  03_01_...es.cpp
~/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_...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
24
25 //===== mean function =====
26 inline double mean(int const & a, int const & b) {
27     PRINT_RED("mean fct version int with arg " << a << " and " << b);
28     return (a + b) / 2.0;
29 }
30
31
32 inline double mean(double const & a, double const & b) {
33     PRINT_GREEN("mean fct version double with arg " << a << " and " << b);
34     return (a + b) / 2.0;
35 }
36
37
38 inline long double mean(long double const & a, long double const & b) {
39     PRINT_MAGENTA("mean fct version long double with arg " << a << " and " << b);
40     return (a + b) / 2.0;
41 }
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     PRINT_NAMED( mean(1, 2) )
Status g++-4.7 -std=c++11 -O2 -o "04_01_functions" "04_01_functions.cpp" (in directory: /home/msk/Desktop/ProgTechFinal/04_mean_trait)
Compiler Compilation finished successfully.
Messages
Scribble
Terminal
line: 26 / 69 col: 0 sel: 16L INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown
```

05\_02\_is\_class.cpp - /home/msk/Desktop/ProgTechFinal/05\_sfinae - Geany

```

26 //===== is_class trait =====
27 template<typename T>
28 struct is_class {
29     //second templates is needed since U::* is illegal for non class types
30     //SFINAE prevents illegal code from being instantiated
31     //U::* is sufficient to identify classes and structs since they have a
32     //destructor that isn't allowed to take parameter hence void(U::*)(void)
33     //and NULL is convertible to the method-pointer
34     template<typename U>
35     static char check(void(U::*)());
36     template<typename U>
37     static double check(...); //just catch anything else (variadic function)
38
39     enum { value = (sizeof(char) == sizeof(check<T>(NULL))) };
40 };
41 //===== some classes =====
42 struct foo_struct {
43 };
44
45 class bar_class {
46 };
47
48 enum baz_enum {
49 };
50
51 template<typename T>
52 void class_check(T const & t) {
53     if(is_class<T>::value)
54         PRINT_YELLOW(TYPE(T) << " is a class-type")

```

Status: g++-4.7 -std=c++11 -O2 -o "04\_01\_functions" "04\_01\_functions.cpp" (n  
 Compiler: Compilation finished successfully.  
 Messages:  
 Scribble  
 Terminal

line: 28 / 86 col: 15 sel: 8 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown

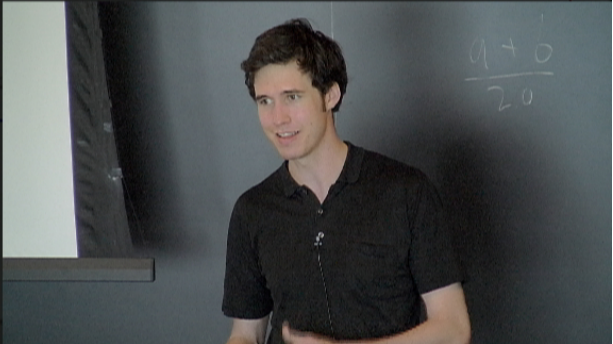




```
Terminal
press enter to continue ===== includes =====
23 #include <iostream>
24 #include <typeinfo>
25
mean fct version i with arg 1 and 2
mean(1, 2) = 1.5
mean fct version d with arg 1.8 and 2
mean(1.8, 2.0) = 1.9
mean fct version e with arg 1.8 and 2
mean(1.8l, 2.0l) = 1.9
fails bc mean<int>(1, 2) has int* as return type and not double...
(program exited with code: 0)
Press return to continue
----- WAIT FOR INPUT() //just hit the enter key to continue
41 PRINT_NAMED(mean(1, 2))
42
43
44 WAIT_FOR_INPUT()
45 PRINT_NAMED(mean(1.8, 2.0))
46
47 WAIT_FOR_INPUT()
48 PRINT_NAMED(mean(1.8l, 2.0l))
49
50 WAIT_FOR_INPUT()

Status g++-4.7 -std=c++11 -O2 -o "04_02_template_attempt" "04_02_template
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 45 / 55 col: 7 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main
```



```
04_03_simple_trait.cpp - /home/msk/Desktop/ProgTechFinal/04_mean_trait - Geany
Documents Files
01_04...me.cpp
01_05...me.cpp
01_06...pl.cpp
~/Deskt...tivation
02_01...ay.cpp
02_02...ay.cpp
02_03...ay.cpp
~/Deskt...e_basics
03_03...on.cpp
~/Deskt...classes
03_01...er.hpp
03_01_impl.cpp
03_01...es.cpp
~/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...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
25
26
27 //===== mean trait =====
28 //the mean_trait only exists during compiletime.
29 template<typename T>
30 struct mean_trait {
31     typedef T type;
32 };|
33 //----- full specialization -----
34 //a matching specialisation is always preferred over the general template
35 template<>
36 struct mean_trait<int> {
37     typedef double type;
38 };
39
40 //===== mean template attempt 2 =====
41 template<typename T>
42 inline typename mean_trait<T>::type mean(T const & a, T const & b) {
43     PRINT_YELLOW("mean fct version " << TYPE(T) <<
44                 " with arg " << a << " and " << b);
45     return (a + b) / 2.0;
46 }
47
48 // +-----+
49 // |                               main                               |
50 // +-----+
51 int main(int argc, char* argv[]) {
52     CLR_SCR()
53     PRINT_CYAN("press enter to continue")

```

Status g++-4.7 -std=c++11 -O2 -o "04\_02\_template\_attempt" "04\_02\_template\_attempt.cpp" (in directory: /home/

Compiler Compilation finished successfully.

Messages

Scribble

Terminal

line: 32 / 72 col: 2 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: PRINT\_MAGENTA



```
04_03_simple_trait.cpp - /home/msk/Desktop/ProgTechFinal/04_mean_trait - Geany
Documents Files
01_04...me.cpp
01_05...me.cpp
01_06...pl.cpp
~/Deskt...tivation
02_01...ay.cpp
02_02...ay.cpp
02_03...ay.cpp
~/Deskt...e_basics
03_03...on.cpp
~/Deskt...classes
03_01...er.hpp
03_01_impl.cpp
03_01...es.cpp
~/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...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
34 //a matching specialisation is always preferred over the general template
35 template<>
36 struct mean_trait<int> {
37     typedef double type;
38 };
39
40 //===== mean template attempt 2 =====
41 template<typename T>
42 inline typename mean_trait<T>::type mean(T const & a, T const & b) {
43     PRINT_YELLOW("mean fct version " << TYPE(T) <<
44                 " with arg " << a << " and " << b);
45     return (a + b) / 2.0;
46 }
47
48 // +-----+
49 // |                               main                               |
50 // +-----+
51 int main(int argc, char* argv[]) {
52     CLR_SCR()
53     PRINT_CYAN("press enter to continue")
54
55     WAIT_FOR_INPUT() //just hit the enter key to continue
56     PRINT_NAMED(mean(1, 2))
57
58     WAIT_FOR_INPUT()
59     PRINT_NAMED(mean(1.8, 2.0))
60
61     WAIT_FOR_INPUT()
62     PRINT_NAMED(mean(1.8l, 2.0l))

```

Status g++-4.7 -std=c++11 -O2 -o "04\_03\_simple\_trait" "04\_03\_simple\_trait.cpp" (in directory: /home/msk/Desktop/ProgTechFinal/04\_mean\_trait)
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 42 / 72 col: 35 sel: 28 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown

```
bool use_double(T)
```

return true

```
trait( ; T)
```

if (use\_d(T))  
return d,  
else  
return T,



```
*04_04_concept_trait.cpp - /home/msk/Desktop/ProgTechFinal/04_mean_trait - Geany
Documents Files
01_04...me.cpp
01_05...me.cpp
01_06...pl.cpp
~/Deskt...tivation
02_01...ay.cpp
02_02...ay.cpp
02_03...ay.cpp
~/Deskt...e_basics
03_03...on.cpp
~/Deskt...lasses
03_01...er.hpp
03_01_impl.cpp
03_01...es.cpp
~/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...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

21 http://www.cplusplus.com/forum/unices/36461/ for details
22 //===== includes =====
23 #include <iostream>
24 #include <typeinfo>
25
26 //===== mean_type_identifier =====
27 template<typename T>
28 struct use_double_identifier {
29     static T t;
30
31     static char check(T);
32     static double check(double);
33
34     enum {value = (sizeof(check((t+t)/double(1))) == sizeof(double))};
35 };
36 //----- full specialisation -----
37 template<>
38 struct use_double_identifier<double> {
39     enum {value = true}; //doesn't matter if true or false
40 };
41
42 //===== meta_if =====
43 template<bool cond, typename T, typename F> //default true
44 struct meta_if {
45     typedef T type;
46 };
47 //----- partial specialisation for true -----
48 template<typename T, typename F>
```

Status g++-4.7 -std=c++11 -O2 -o "04\_03\_simple\_trait" "04\_03\_simple\_trait.cpp"
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 29 / 109 col: 15 sel: 11 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: PRINT\_MAGENTA



```
*04_04_concept_trait.cpp - /home/msk/Desktop/ProgTechFinal/04_mean_trait - Geany
Documents Files
01_04...me.cpp
01_05...me.cpp
01_06...pl.cpp
~/Deskt...tivation
02_01...ay.cpp
02_02...ay.cpp
02_03...ay.cpp
~/Deskt...e_basics
03_03...on.cpp
~/Deskt...lasses
03_01...er.hpp
03_01_impl.cpp
03_01...es.cpp
~/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...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

21 http://www.cplusplus.com/forum/unices/36461/ for details
22 //===== includes =====
23 #include <iostream>
24 #include <typeinfo>
25
26 //===== mean_type_identifier =====
27 template<typename T>
28 struct use_double_identifier {
29     static T t;
30
31     static char check(T);
32     static double check(double);
33
34     enum {value = (sizeof(check((t+t)/double(1))) == sizeof(double))};
35 };
36 //----- full specialisation -----
37 template<>
38 struct use_double_identifier<double> {
39     enum {value = true}; //doesn't matter if true or false
40 };
41
42 //===== meta_if =====
43 template<bool cond, typename T, typename F> //default true
44 struct meta_if {
45     typedef T type;
46 };
47 //----- partial specialisation for true -----
48 template<typename T, typename F>
```

Status g++-4.7 -std=c++11 -O2 -o "04\_03\_simple\_trait" "04\_03\_simple\_trait.cpp"
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 28 / 109 col: 0 sel: 0 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown





```
*04_04_concept_trait.cpp - /home/msk/Desktop/ProgTechFinal/04_mean_trait - Geany
Documents Files
01_04...me.cpp
01_05...me.cpp
01_06...pl.cpp
~/Deskt...tivation
02_01...ay.cpp
02_02...ay.cpp
02_03...ay.cpp
~/Deskt...e_basics
03_03...on.cpp
~/Deskt...lasses
03_01...er.hpp
03_01_impl.cpp
03_01...es.cpp
~/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...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

21 http://www.cplusplus.com/forum/unices/36461/ for details
22 //===== includes =====
23 #include <iostream>
24 #include <typeinfo>
25
26 //===== mean_type_identifier =====
27 template<typename T>
28 struct use_double_identifier {
29     static T t;
30
31     static char check(T);
32     static double check(double);
33
34     enum {value = (sizeof(check((t+t)/double(1))) == sizeof(double))};
35 };
36 //----- full specialisation -----
37 template<>
38 struct use_double_identifier<double> {
39     enum {value = true}; //doesn't matter if true or false
40 };
41
42 //===== meta_if =====
43 template<bool cond, typename T, typename F> //default true
44 struct meta_if {
45     typedef T type;
46 };
47 //----- partial specialisation for true -----
48 template<typename T, typename F>
```

Status g++-4.7 -std=c++11 -O2 -o "04\_03\_simple\_trait" "04\_03\_simple\_trait.cpp"
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 33 / 109 col: 4 sel: 0 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: PRINT\_MAGENTA



\*04\_04\_concept\_trait.cpp - /home/msk/Desktop/ProgTechFinal/04\_mean\_trait - Geany

Documents Files

01\_04...me.cpp  
01\_05...me.cpp  
01\_06...pl.cpp  
~/Deskt...tivation  
02\_01...ay.cpp  
02\_02...ay.cpp  
02\_03...ay.cpp  
~/Deskt...e\_basics  
03\_03...on.cpp  
~/Deskt...classes  
03\_01...er.hpp  
03\_01\_impl.cpp  
03\_01...es.cpp  
~/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...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

```

21 http://www.cplusplus.com/forum/unices/36461/ for details
22 //===== includes =====
23 #include <iostream>
24 #include <typeinfo>
25
26 //===== mean_type_identifier =====
27 template<typename T>
28 struct use_double_identifier {
29     static T t;
30
31     static char check(T);
32     static double check(double);
33
34     enum {value = (sizeof(check((t+t)/double(1))) == sizeof(double))};
35
36     enum color {red = 1, blue = 2, green = 3};
37 };
38
39
40 class color {
41 public:
42
43 private:
44 };
45
46
47
48

```

Status g++-4.7 -std=c++11 -O2 -o "04\_03\_simple\_trait" "04\_03\_simple\_trait.cpp"

Compiler Compilation finished successfully.

Messages

Scribble

Terminal

line: 42 / 126 col: 4 sel: 0 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown





```
*04_04_concept_trait.cpp - /home/msk/Desktop/ProgTechFinal/04_mean_trait - Geany
Documents Files
01_04...me.cpp
01_05...me.cpp
01_06...pl.cpp
~/Deskt...tivation
02_01...ay.cpp
02_02...ay.cpp
02_03...ay.cpp
~/Deskt...e_basics
03_03...on.cpp
~/Deskt...classes
03_01...er.hpp
03_01_impl.cpp
03_01...es.cpp
~/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...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

25
26 //===== mean_type_identifier =====
27 template<typename T>
28 struct use_double_identifier {
29     static T t;
30
31     static char check(T);
32     static double check(double);
33
34     enum {value = (sizeof(check((t+t)/double(1))) == sizeof(double))};
35
36     enum color {red = 1, blue = 2, green = 3};
37 };
38 };
39
40 //----- full specialisation -----
41 template<>
42 struct use_double_identifier<double> {
43     enum {value = true}; //doesn't matter if true or false
44 };
45
46 //===== meta_if =====
47 template<bool cond, typename T, typename F> //default true
48 struct meta_if {
49     typedef T type;
50 };
51 //----- partial specialisation for true -----
52 template<typename T, typename F>
53 struct meta_if<false, T, F> {
```

Status g++-4.7 -std=c++11 -O2 -o "04\_03\_simple\_trait" "04\_03\_simple\_trait.cpp" (in directory: /home/msk/Desktop  
Compiler Compilation finished successfully.  
Messages  
Scribble  
Terminal

line: 34 / 113 col: 70 sel: 0 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: PRINT\_MAGENTA

```
*04_04_concept_trait.cpp - /home/msk/Desktop/ProgTechFinal/04_mean_trait - Geany
Documents Files
01_04...me.cpp
01_05...me.cpp
01_06...pl.cpp
~/Deskt...tivation
02_01...ay.cpp
02_02...ay.cpp
02_03...ay.cpp
~/Deskt...e_basics
03_03...on.cpp
~/Deskt...classes
03_01...er.hpp
03_01_impl.cpp
03_01...es.cpp
~/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...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

26 //===== mean_type_identifier =====
27 template<typename T>
28 struct use_double_identifier {
29     static T t;
30
31     static char check(T);
32     static double check(double);
33
34     enum {value = (sizeof(check((t+t)/double(1))) == sizeof(double))};
35 };
36
37 //----- full specialisation -----
38 template<>
39 struct use_double_identifier<double> {
40     enum {value = true}; //doesn't matter if true or false
41 };
42
43 //===== meta_if =====
44 template<bool cond, typename T, typename F> //default true
45 struct meta_if {
46     typedef T type;
47 };
48 //----- partial specialisation for true -----
49 template<typename T, typename F>
50 struct meta_if<false, T, F> {
51     typedef F type;
52 };
53
54 //===== better_mean_trait =====

Status g++-4.7 -std=c++11 -O2 -o "04_03_simple_trait" "04_03_simple_trait.cpp" (in directory: /home/msk/Desktop
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 41 / 110 col: 2 sel: 111 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown
```



```

*04_04_concept_trait.cpp - /home/msk/Desktop/ProgTechFinal/04_mean_trait - Geany
Documents Files
01_04...me.cpp
01_05...me.cpp
01_06...pl.cpp
~/Deskt...tivation
02_01...ay.cpp
02_02...ay.cpp
02_03...ay.cpp
~/Deskt...e_basics
03_03...on.cpp
~/Deskt...lasses
03_01...er.hpp
03_01_impl.cpp
03_01...es.cpp
~/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...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

39 struct use_double_identifier<double> {
40     enum {value = true}; //doesn't matter if true or false
41 };
42
43 //===== meta_if =====
44 template<bool cond, typename T, typename F> //default true
45 struct meta_if {
46     typedef T type;
47 };
48 //----- partial specialisation for true -----
49 template<typename T, typename F>
50 struct meta_if<false, T, F> {
51     typedef F type;
52 };
53
54 //===== better_mean_trait =====
55 template<typename T>
56 struct better_mean_trait {
57     typedef
58         typename
59             meta_if<use_double_identifier<T>::value
60                 , double
61                 , T
62             >::type
63         type;
64 };
65 //===== mean template attempt 3 =====
66 template<typename T>
67 inline typename better_mean_trait<T>::type mean(T const & a, T const & b) {

```

Status g++-4.7 -std=c++11 -O2 -o "04\_03\_simple\_trait" "04\_03\_simple\_trait.cpp" (in directory: /home/msk/Desktop)

Compiler Compilation finished successfully.

Messages

Scribble

Terminal

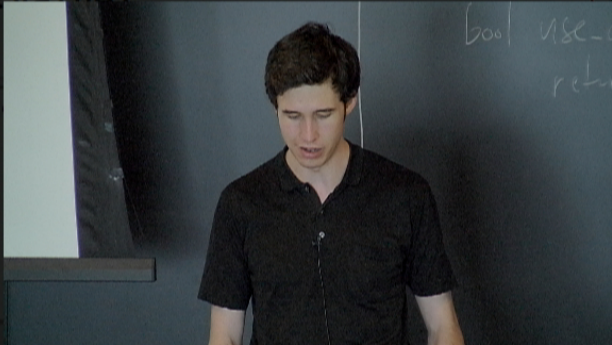
line: 44 / 110 col: 0 sel: 0 INS SP MOD mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown

Terminal

```
press enter to continue
62     >::type
63     type;
64 };
65     template attempt 3 =====
66     template<typename T>
67     struct better_trait<T>::type mean(T const & a, T const & b) {
68         PRINT_YELLOW("mean fct version " << TYPE(T) <<
69             " with arg " << a << " and " << b);
70     };
71 }
72
73 // -----+-----
74 // |                               |
75 // +-----+-----+
76 int main(int argc, char* argv[]) {
77     CLR_SCR()
78     PRINT_CYAN("press enter to continue")
79
80     WAIT_FOR_INPUT() //just hit the enter key to continue
81     PRINT_NAMED(use_double_identifier<int>::value)
82     WAIT_FOR_INPUT()
83     PRINT_NAMED(use_double_identifier<double>::value)
84     WAIT_FOR_INPUT()
85     PRINT_NAMED(use_double_identifier<uint>::value)
86     WAIT_FOR_INPUT()
87     PRINT_NAMED(use_double_identifier<long double>::value)
88
89     WAIT_FOR_INPUT()
```

Status g++-4.7 -std=c++11 -O2 -o "04\_04\_concept\_trait" "04\_04\_concept\_trait.  
Compiler Compilation finished successfully.  
Messages  
Scribble  
Terminal

line: 89 / 110 col: 20 sel: 0 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: main





Wiki

[Start Page](#) [Index](#) [History](#) [Last Change](#)

## Mean Value Type concept

Let:

- M be a model of Mean Value Type concept, m1, m2 be objects of type M
- d be an object of type double
- out be an object of type std::ostream

Valid expressions:

- M m1()
- M m1(int())
- M m1(m2)
- m1 + m2, where m1 and/or m2 are allowed to be const
- m1 \* m2, where m1 and/or m2 are allowed to be const
- m1 / d, returns a type M or a type double
- m1 = m2
- m1 += m2
- out << m1

[back](#)

Download in other formats:



Let:

- M be a model of Mean Value Type concept, m1, m2 be objects of type M
- d be an object of type double
- out be an object of type std::ostream

Valid expressions:

- M m1()
- M m1(int())
- M m1(m2)
- m1 + m2, where m1 and/or m2 are allowed to be const
- m1 \* m2, where m1 and/or m2 are allowed to be const
- m1  $\neq$  d, returns a type M or a type double
- m1 = m2
- m1 += m2
- out << m1

[back](#)

Download in other formats:

Plain Text



Powered by **Trac 0.11.7**  
By Edgewall Software.





```
05_01_funct...e.cpp /home/mrk/Desktop/ProgTech/Final/05/05final/05_01_funct...e.cpp - Geany 16:34:10
Documents Files
01_04...me.cpp
01_05...me.cpp
01_06...pl.cpp
~/Deskt...tivation
02_01...ay.cpp
02_02...ay.cpp
02_03...ay.cpp
~/Deskt...e_basics
03_03...on.cpp
~/Deskt...classes
03_01...er.hpp
03_01_impl.cpp
03_01...es.cpp
~/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...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
26 //===== includes =====
27 #include <iostream>
28 #include <typeinfo>
29
30 //===== normal functions =====
31 void print(int i, bool b) {
32     PRINT_RED("normal function")
33 }
34 //===== full template function =====
35 template<typename T, typename U>
36 void print(T i, U b) {
37     PRINT_MAGENTA("base template normal")
38 }
39 //full specialisation of template normal
40 template<>
41 void print<>(int * i, bool b) {
42     PRINT_MAGENTA("spec template normal")
43 }
44 template<typename T, typename U>
45 void print(T * i, U b) {
46     PRINT_CYAN("base template pointer")
47 }
48 //full specialisation of template pointer
49 //~ template<>
50 //~ void print<>(int * i, bool b) {
51     //~ PRINT_CYAN("spec template pointer")
52 //~ }
53 //===== partial template function =====
54 //because of function overloading there is no partial specialisation

Status g++-4.7 -std=c++11 -O2 -o "04_04_concept_trait" "04_04_concept_trait.cpp" (in directory: /home/mrk/Desk
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 31 / 95 col: 27 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
Documents Files
01_04...me.cpp
01_05...me.cpp
01_06...pl.cpp
~/Deskt...tivation
02_01...ay.cpp
02_02...ay.cpp
02_03...ay.cpp
~/Deskt...e_basics
03_03...on.cpp
~/Deskt...classes
03_01...er.hpp
03_01_impl.cpp
03_01...es.cpp
~/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...me.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

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 //doesn't work for foo bc the argument given is 10 and not convertible to foo
33 template<typename T>
34 void fct(T t) {
35     PRINT_RED(TYPE(T) << " has no size_type")
36 }
37 //===== a type that has a typedef size_type =====
38 struct foo {
39     typedef int size_type;
40 };
41
42 // +-----+
43 // |                    main                    |
44 // +-----+
45 int main(int argc, char* argv[]) {
46     CLR_SCR()
47     PRINT_CYAN("press enter to continue")
48
49     WAIT_FOR_INPUT() //just hit the enter key to continue
50     fct<foo>(10);
51
52     WAIT_FOR_INPUT()
53     fct<int>(10);
54
Status g++-4.7 -std=c++11 -O2 -o "04_04_concept_trait" "04_04_concept_trait.cpp" (in directory: /home/msk/Desk
Compiler Compilation finished successfully.
Messages
Scribble
Terminal

line: 29 / 61 col: 30 sel: 21 INS SP mode: Unix (LF) encoding: UTF-8 filetype: C++ scope: unknown
```