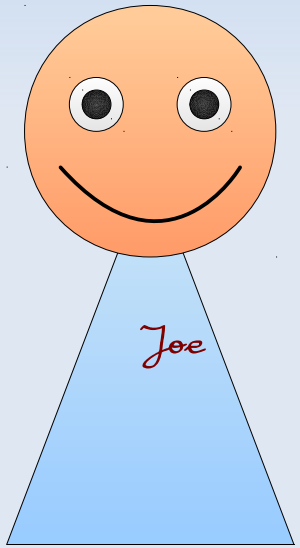
Interactive metaprogramming shell based on Clang

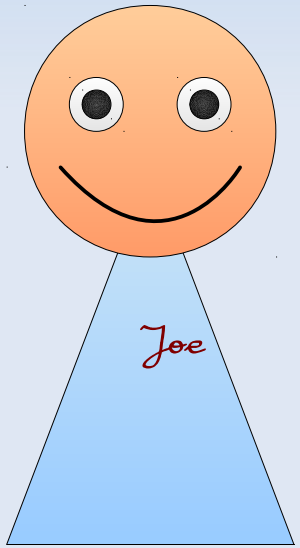
Ábel Sinkovics

Agenda

Agenda

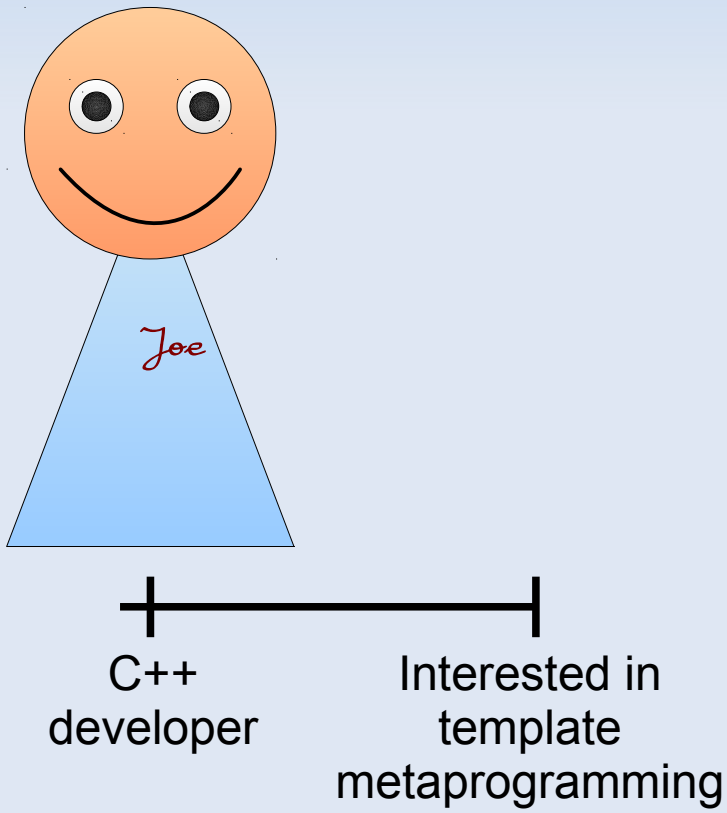


Agenda

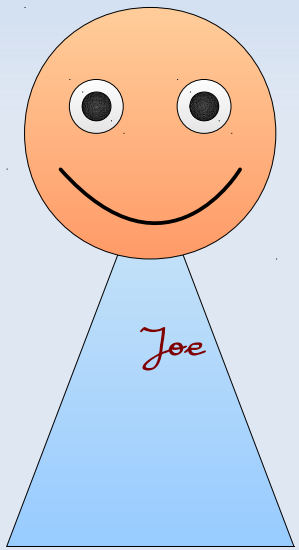


C++
developer

Agenda



Agenda

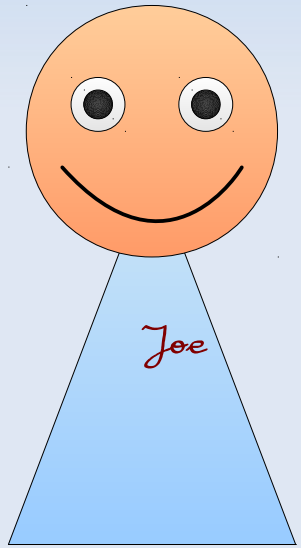


C++ developer

Interested in template metaprogramming

Advanced template metaprogrammer

Agenda

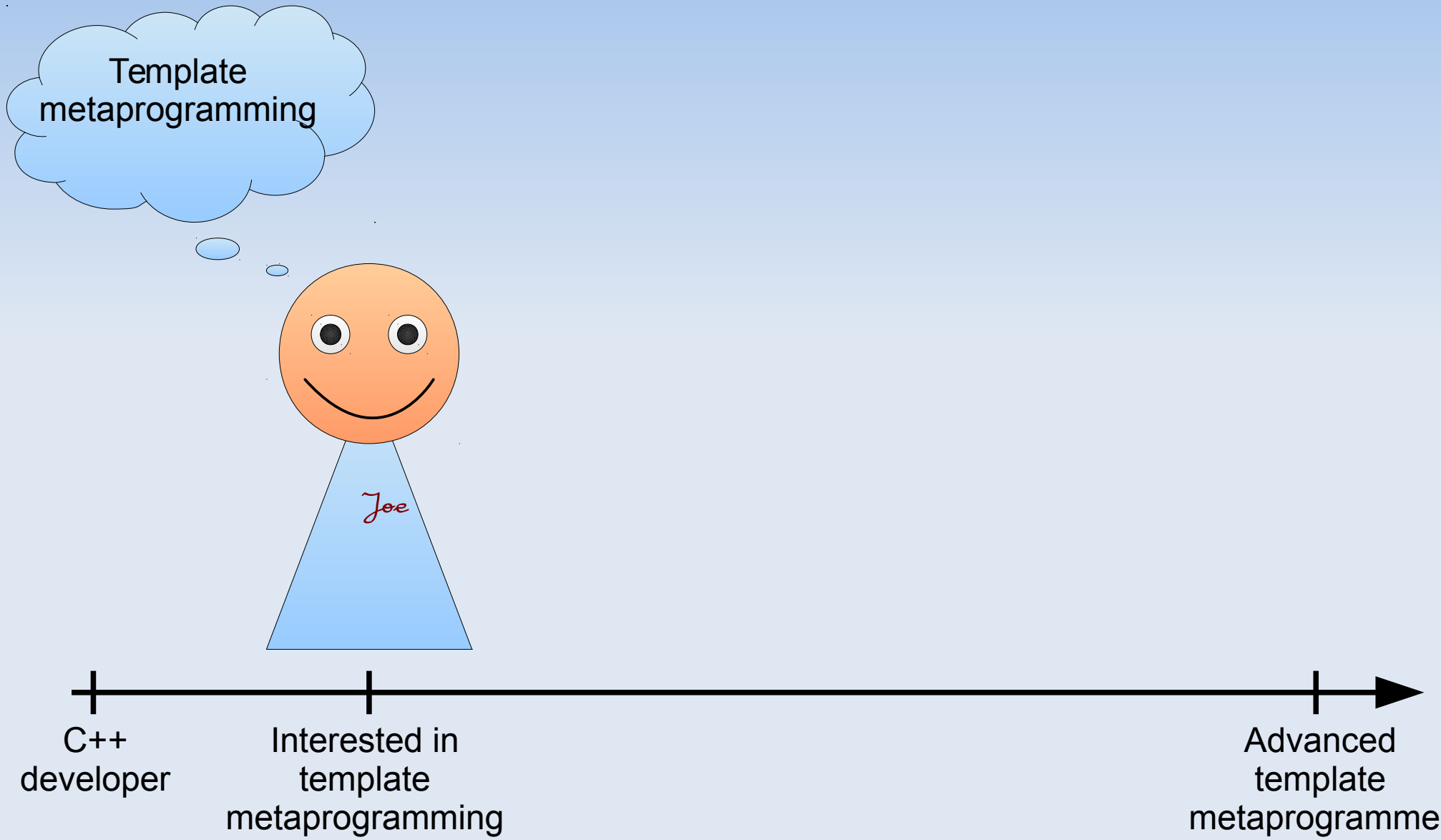


C++ developer

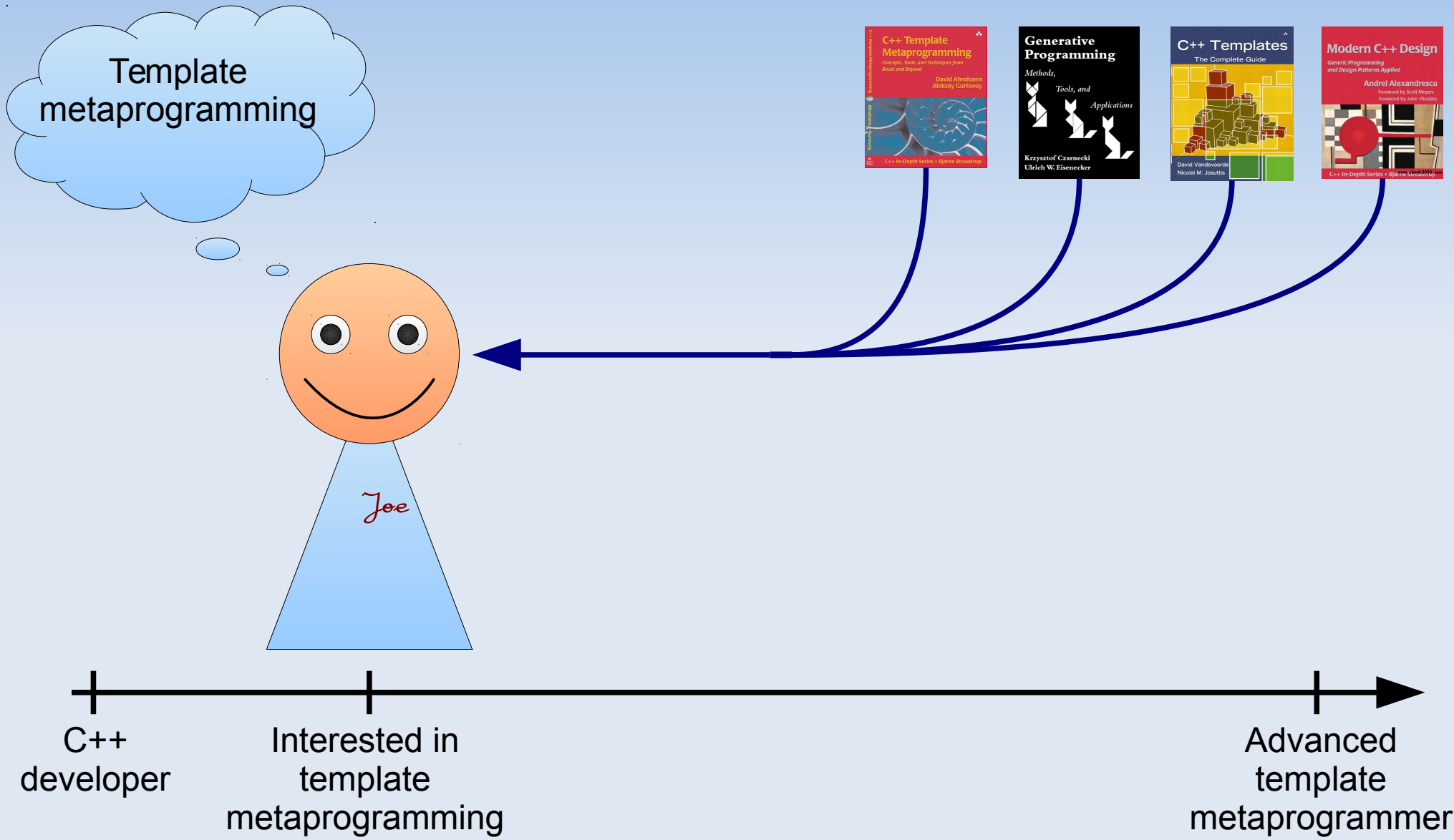
Interested in template metaprogramming

Advanced template metaprogrammer

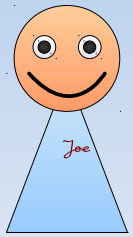
Agenda


Template metaprogramming
Joe
C++ developer
Interested in template metaprogramming
Advanced template metaprogrammer

Agenda

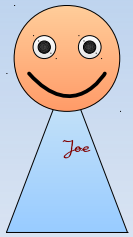

Template metaprogramming
C++ Template Metaprogramming
Concepts, Tools, and Techniques from Boost and Beyond
David Abrahams
Aleksey Gurtovoy
C++ In-Depth Series • Bjarne Stroustrup
Generative Programming
Methods, Tools, and Applications
Krzysztof Czarnecki
Ulrich W. Eisenecker
C++ Templates
The Complete Guide
David Vandevoorde
Nicolai M. Josuttis
Modern C++ Design
Generic Programming and Design Patterns Applied
Andrei Alexandrescu
Foreword by Scott Meyers
Foreword by John Vlissides
C++ In-Depth Series • Bjarne Stroustrup
Joe
C++ developer
Interested in template metaprogramming
Advanced template metaprogrammer

Template metaprogramming




```
template <int N>
struct fact {
  enum { value = N * fact<N – 1>::value };
};

template <>
struct fact<0> {
  enum { value = 1  };
};
```

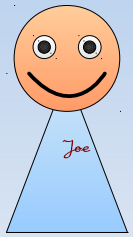
Template metaprogramming




```
template <int N>
struct fact {
  enum { value = N * fact<N – 1>::value };
};

template <>
struct fact<0> {
  enum { value = 1  };
};
```

fact<3>

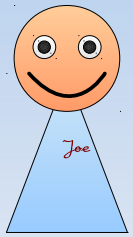
Template metaprogramming




```
template <int N>
struct fact {
  enum { value = N * fact<N – 1>::value };
};

template <>
struct fact<0> {
  enum { value = 1  };
};
```

```
fact<3>::value
```

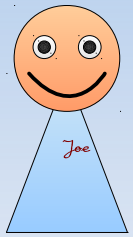
Template metaprogramming




```
template <int N>
struct fact {
  enum { value = N * fact<N – 1>::value };
};

template <>
struct fact<0> {
  enum { value = 1  };
};
```

```
int main() {
  std::cout << fact<3>::value << std::endl;
}
```

Template metaprogramming

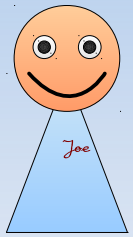



```
template <int N>
struct fact {
  enum { value = N * fact<N - 1>::value };
};

template <>
struct fact<0> {
  enum { value = 1  };
};
```

```
int main() {
  std::cout << fact<3>::value << std::endl;
}
```

```
$
```

Template metaprogramming

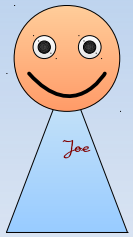



```
template <int N>
struct fact {
  enum { value = N * fact<N – 1>::value };
};

template <>
struct fact<0> {
  enum { value = 1  };
};
```

```
int main() {
  std::cout << fact<3>::value << std::endl;
}
```

```
$ g++ test_fact.cpp
$
```

Template metaprogramming




```
template <int N>
struct fact {
  enum { value = N * fact<N – 1>::value };
};

template <>
struct fact<0> {
  enum { value = 1  };
};
```

```
int main() {
  std::cout << fact<3>::value << std::endl;
}
```

```
$ g++ test_fact.cpp
$ ./a.out
6
$
```

C++ template metafunction

Argument list

Name

Body

C++ template metafunction

```
template <class T>
struct add_const
{
  typedef const T type;
};
```

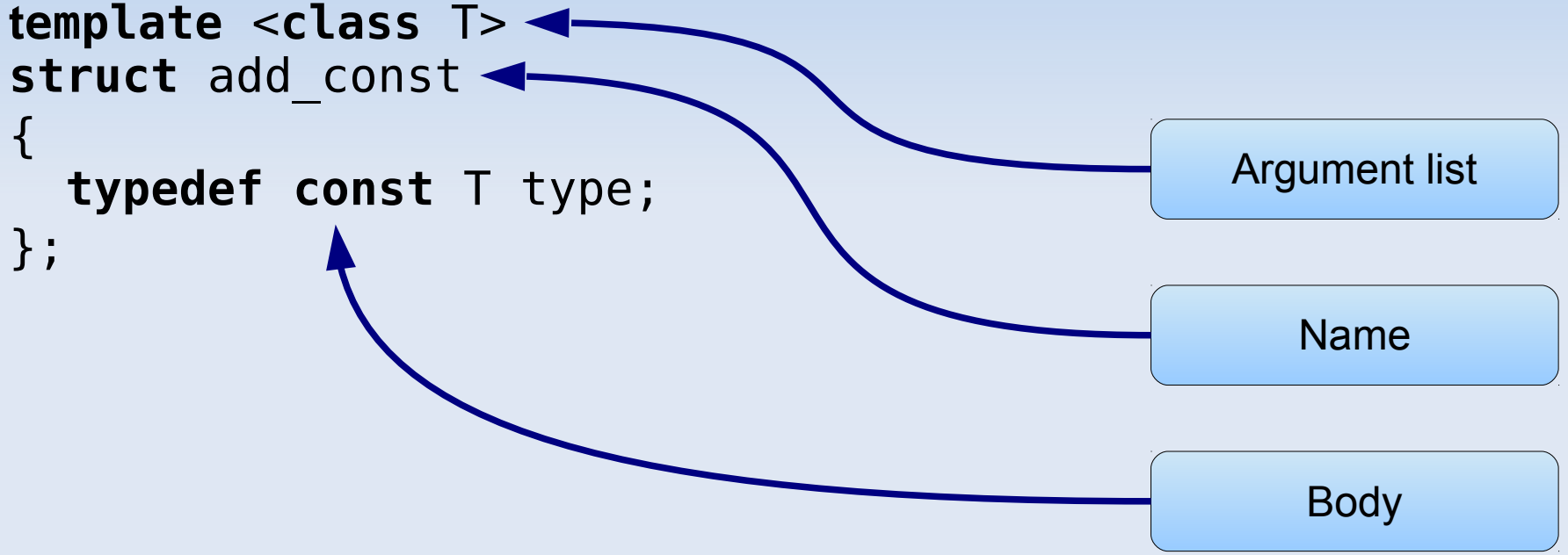
Argument list

Name

Body

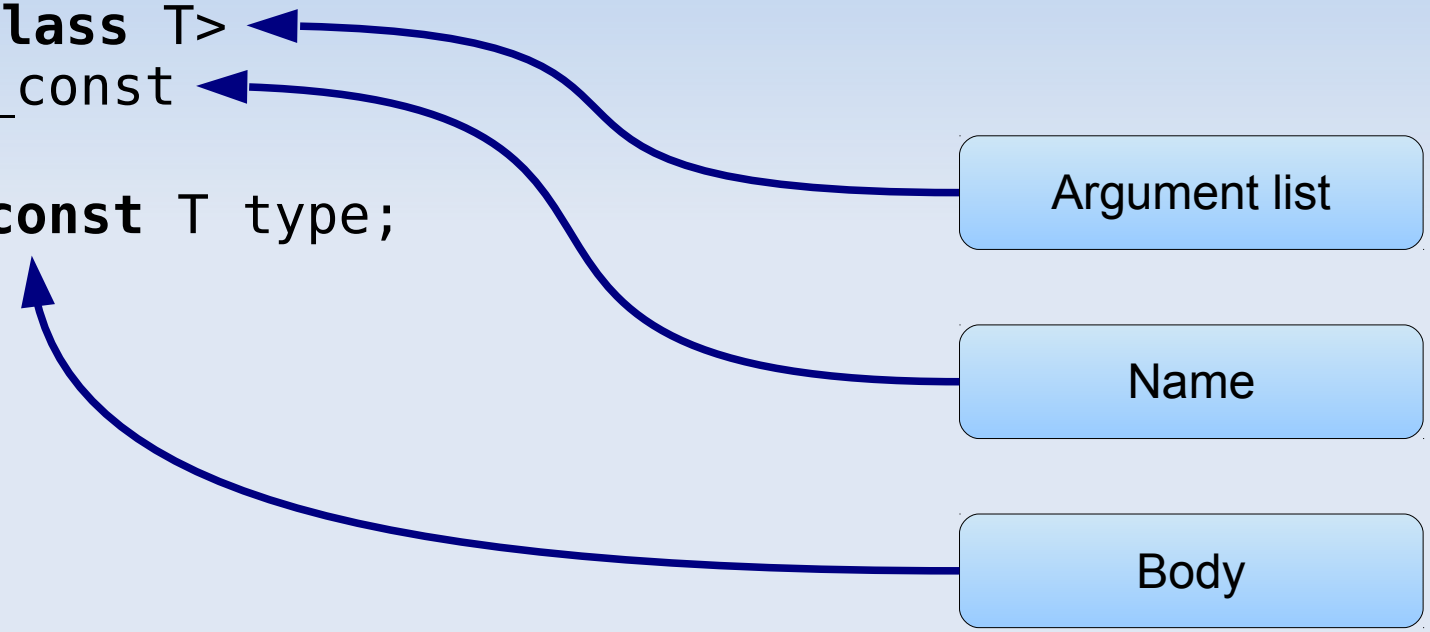
C++ template metafunction

```
template <class T>
struct add_const
{
  typedef const T type;
};
```



C++ template metafunction

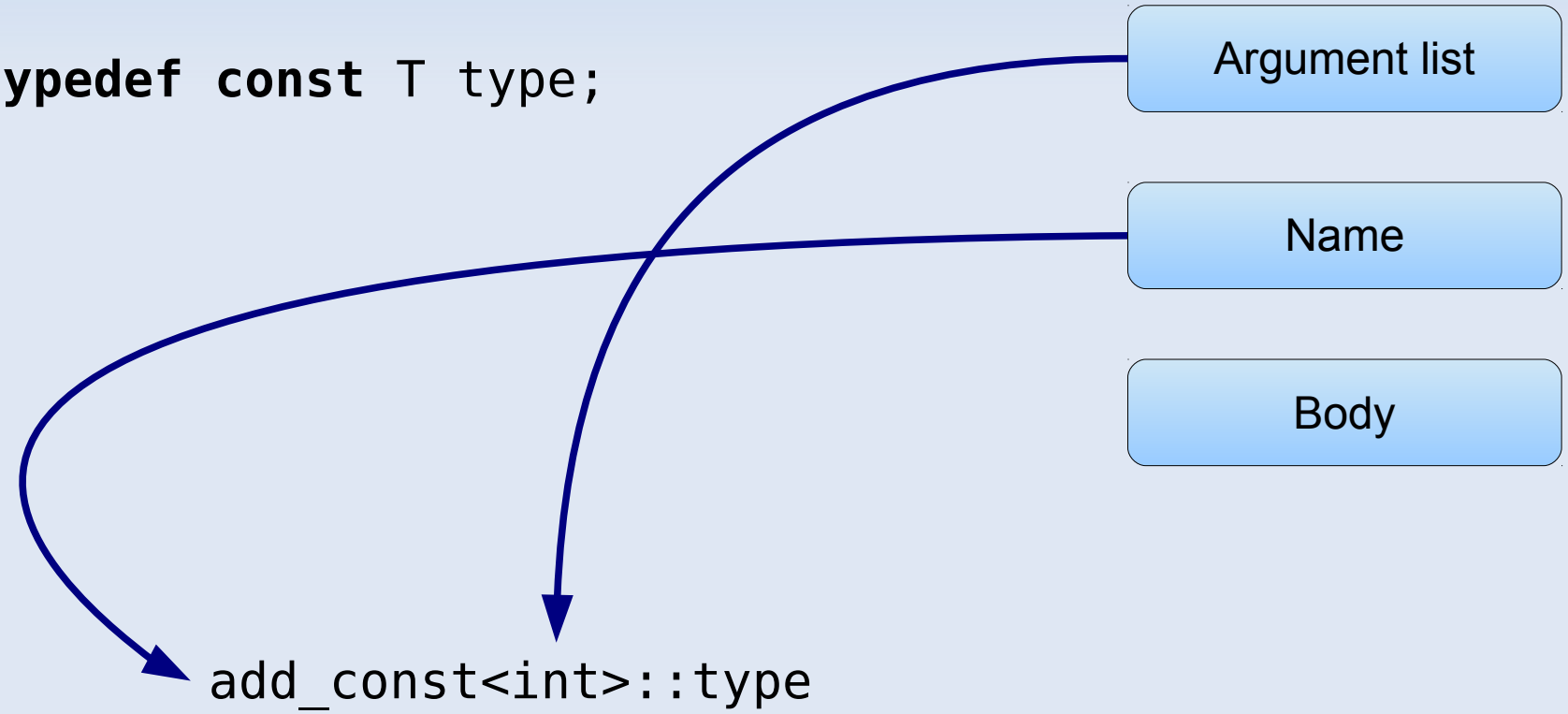
```
template <class T>
struct add_const
{
  typedef const T type;
};
```



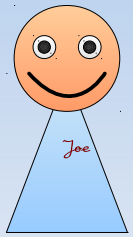

```
add_const<int>::type
```

C++ template metafunction

```
template <class T>
struct add_const
{
  typedef const T type;
};
```

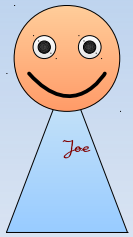


Template metaprogramming



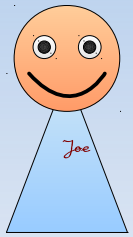

```
template <class T>
struct add_const {
  typedef    const T type;
};
```

Template metaprogramming



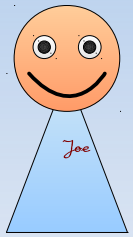

```
template <class T>
struct add_const {
  typedef    const T type;
};
```

Template metaprogramming




```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

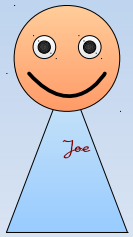
Template metaprogramming




```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
int main() {
  add_const<int>::type x = 11;
  x = 13;
}
```

Template metaprogramming

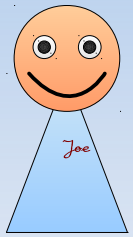



```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
int main() {
  add_const<int>::type x = 11;
  x = 13;
}
```

Compiles...

Template metaprogramming

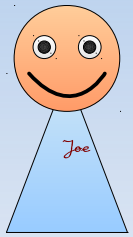



```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
int main() {
  add_const<int>::type x = 11;
  x = 13;
}
```

Compiles...

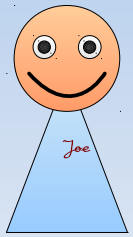
Template metaprogramming




```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
int main() {

    std::is_same<
      const int,
      add_const<int>::type
    >::type::value


}
```

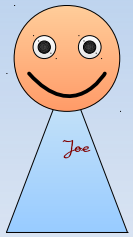
Template metaprogramming




```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
int main() {
  static_assert(
    std::is_same<
      const int,
      add_const<int>::type
    >::type::value,
    "Testing my metafunction"
  );
}
```

Template metaprogramming

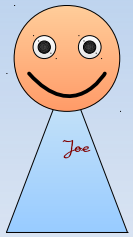



```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
int main() {
  static_assert(
    std::is_same<
      const int,
      add_const<int>::type
    >::type::value,
    "Testing my metafunction"
  );
}
```

```
v.cpp: In function ‘int main()’:
v.cpp:12:3: error: static assertion failed: Testing my metafunction
   static_assert(
   ^
```

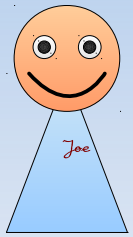
Template metaprogramming




```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
int main() {
  mpllibs::metamonad::fail_with_type<
    add_const<int>::type
  >();
}
```

Template metaprogramming

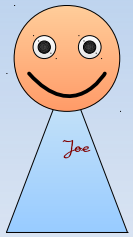



```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
int main() {
  mpllibs::metamonad::fail_with_type<
    add_const<int>::type
  >();
}
```

```
In file included from mpllibs/metamonad/fail_with_type.hpp:9:0,
                 from test.cpp:1:
mpllibs/metamonad/v1/fail_with_type.hpp: In instantiation of ‘void mpllibs::
metamonad::v1::fail_with_type() [with T = volatile int]’:
v.cpp:13:5:   required from here
mpllibs/metamonad/v1/fail_with_type.hpp:26:70: error: ‘f’ is not a member of
 ‘mpllibs::metamonad::v1::impl::
FAIL_WITH_TYPE_____________________________<volatile int>’
         impl::FAIL_WITH_TYPE_____________________________<T>::f();
                                                                ^
```

Template metaprogramming

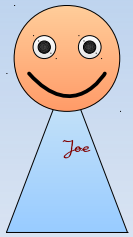



```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
int main() {
  mpllibs::metamonad::fail_with_type<
    add_const<int>::type
  >();
}
```

```
In file included from mpllibs/metamonad/fail_with_type.hpp:9:0,
                 from test.cpp:1:
mpllibs/metamonad/v1/fail_with_type.hpp: In instantiation of ‘void mpllibs::
metamonad::v1::fail_with_type() [with T = volatile int]’:
v.cpp:13:5:   required from here
mpllibs/metamonad/v1/fail_with_type.hpp:26:70: error: ‘f’ is not a member of
 ‘mpllibs::metamonad::v1::impl::
FAIL_WITH_TYPE_____________________________<volatile int>’
         impl::FAIL_WITH_TYPE_____________________________<T>::f();
                                                                ^
```

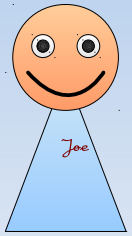
Template metaprogramming




```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

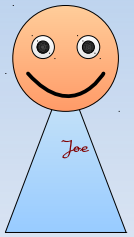
Template metaprogramming

```
$ python
>>>
```



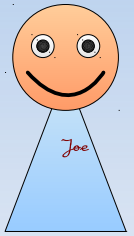
Template metaprogramming

```
$ python
>>> def fact(n):
...   if n == 0:
...     return 1
...   else:
...     return n * fact(n – 1)
...
>>>
```



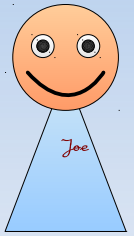
Template metaprogramming

```
$ python
>>> def fact(n):
...   if n == 0:
...     return 1
...   else:
...     return n * fact(n – 1)
...
>>> fact(3)
6
>>>
```



Template metaprogramming

```
$ python
>>> def fact(n):
...   if n == 0:
...     return 1
...   else:
...     return n * fact(n – 1)
...
>>> fact(3)
6
>>>
```



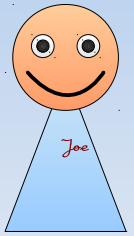
Erlang

Haskell

...

Template metaprogramming

```
$ python
>>> def fact(n):
...   if n == 0:
...     return 1
...   else:
...     return n * fact(n – 1)
...
>>> fact(3)
6
>>>
```



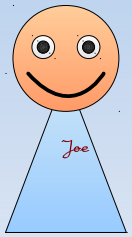
Erlang

Haskell

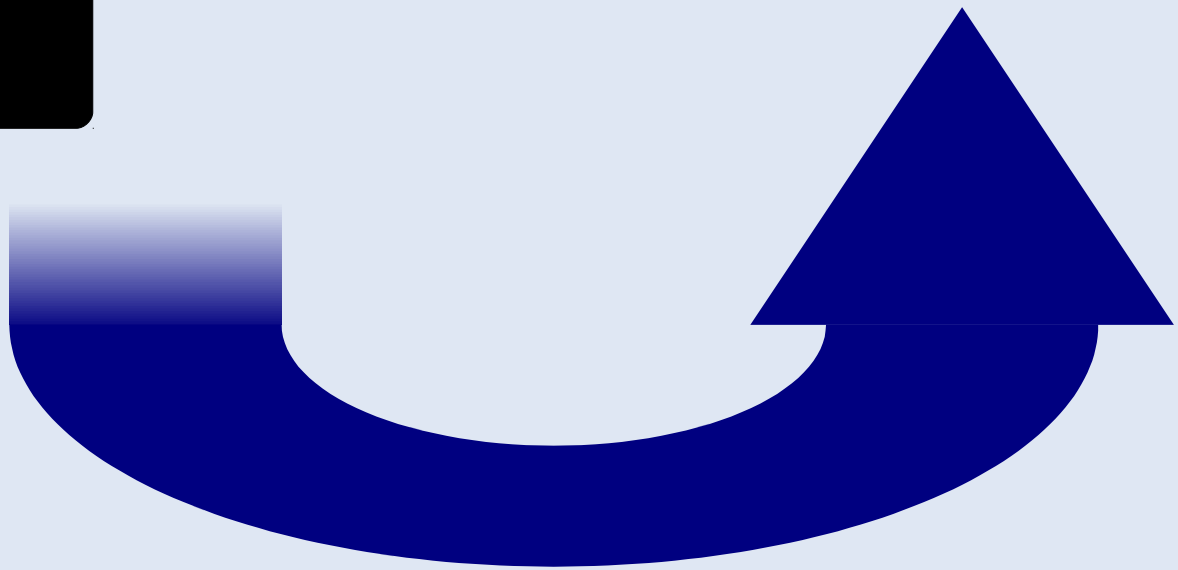
...

Template metaprogramming

```
– 1)
```

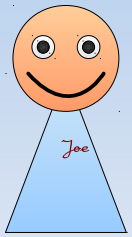



```
template <cla
struct add_co
  typedef vol
};
```

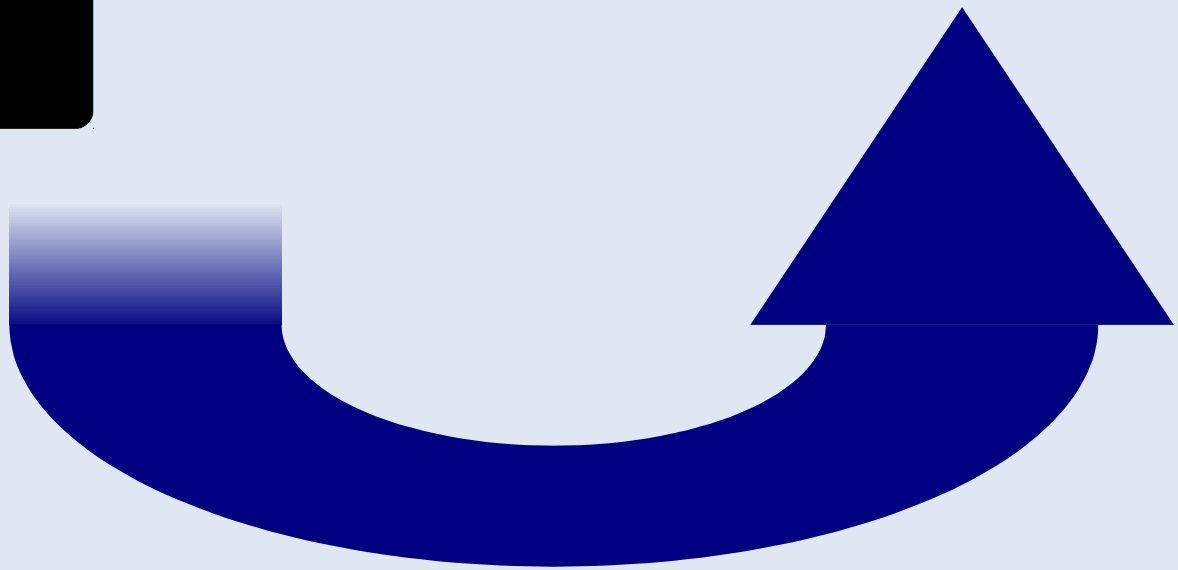


Template metaprogramming

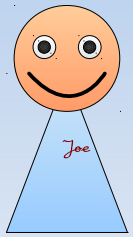
```
– 1)
```




```
template <cla
struct add_co
  typedef vol
};
```



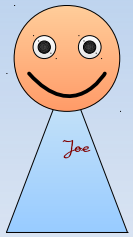
Template metaprogramming




```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
>
```

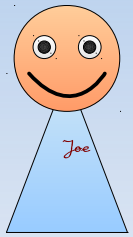
Template metaprogramming




```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
> add_const<int>::type
```

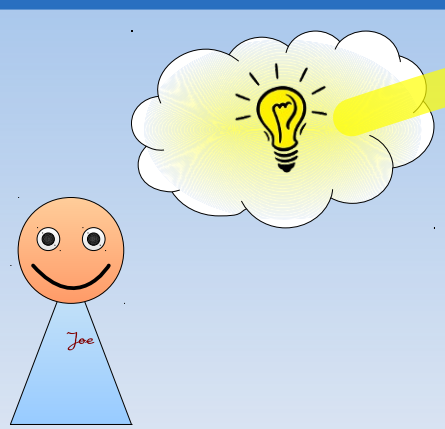
Template metaprogramming




```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
> add_const<int>::type
volatile int
>
```

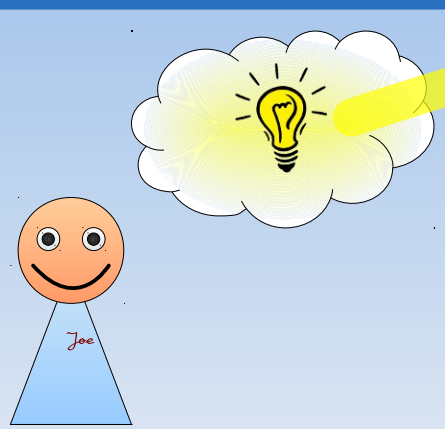
Template metaprogramming




```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
> add_const<int>::type
volatile int
>
```

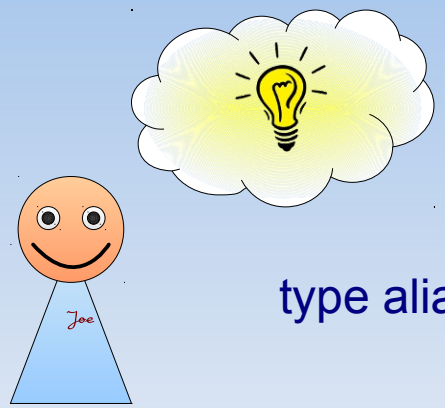
Template metaprogramming




```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
> add_const<int>::type
volatile int
>
```

Template metaprogramming

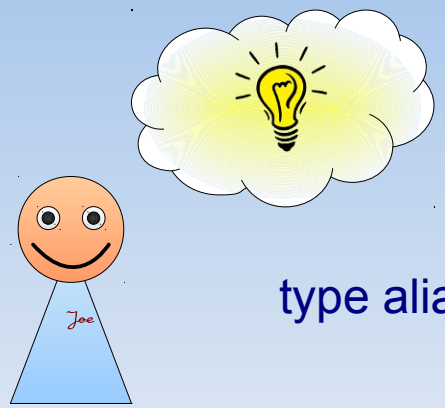


type alias

```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
> add_const<int>::type
volatile int
>
```

Template metaprogramming



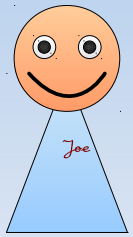
type alias

```
template <class T>
struct add_const {
  typedef volatile T type;
};
```

```
> add_const<int>::type
volatile int
>
```

Resolve type aliases

Template metaprogramming

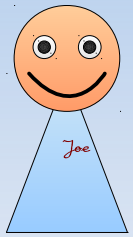



```
template <class T>
struct add_const {
  typedef const T type;
};
```

```
> add_const<int>::type
volatile int
>
```

Resolve type aliases

Template metaprogramming




```
template <class T>
struct add_const {
  typedef const T type;
};
```

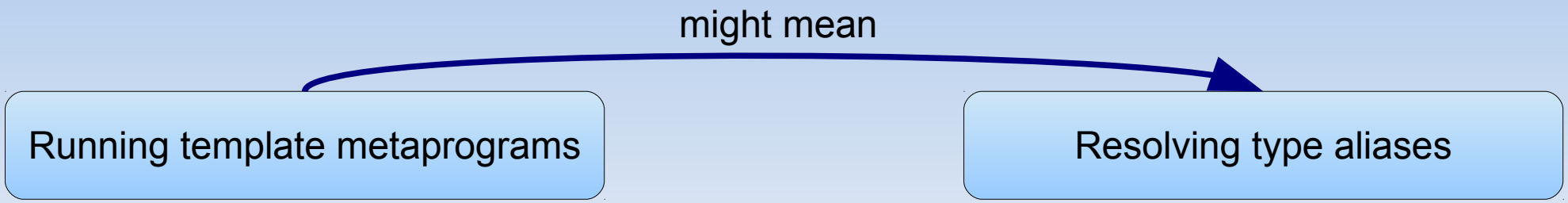
```
> add_const<int>::type
volatile int
> add_const<int>::type
const int
>
```

Resolve type aliases

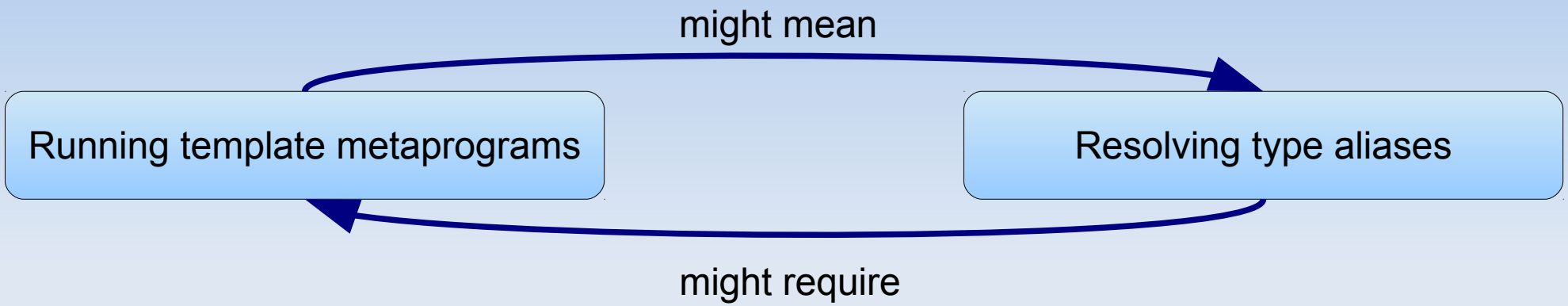
Template metaprogram evaluation

Running template metaprograms

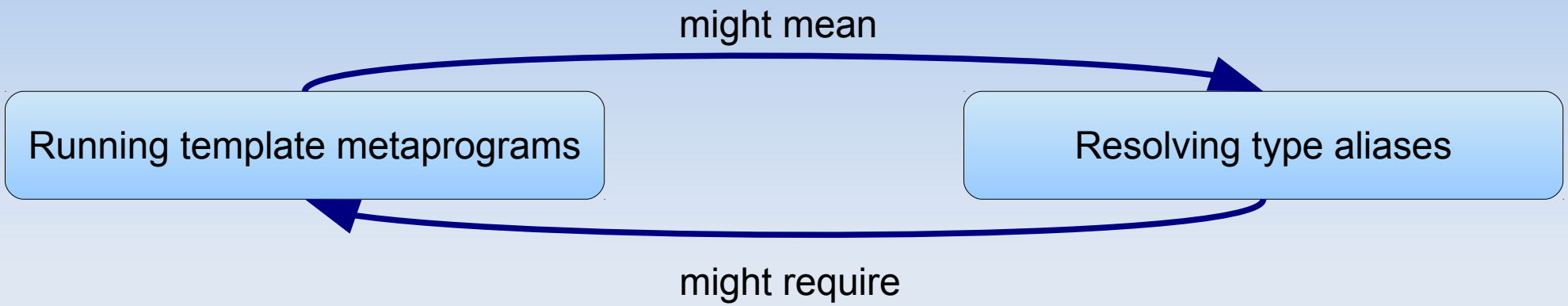
Template metaprogram evaluation



Template metaprogram evaluation



Template metaprogram evaluation



and it might mean running a Haskell interpreter...

Template metaprogram evaluation

might mean

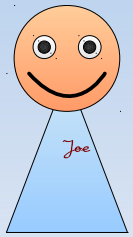
Running template metaprograms

Resolving type aliases

might require

and it [illegible] erpreter...

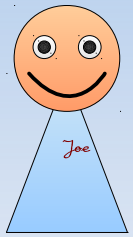
Template metaprogramming




```
template <class T>
struct add_const {
  typedef const T type;
};
```

```
>
```

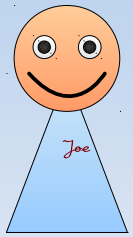
Template metaprogramming




```
template <class T>
struct add_const {
  typedef const T type;
};
```

```
> add_const<int>::type
```

Template metaprogramming

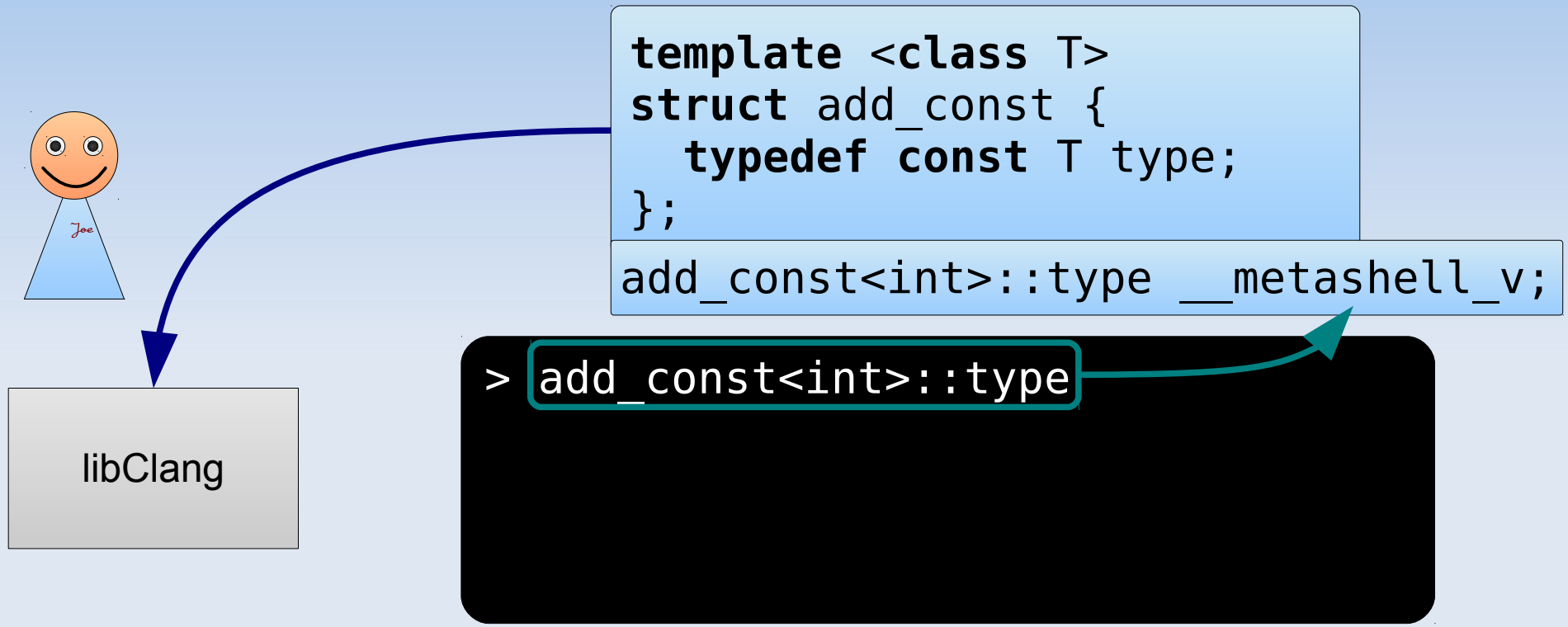



```
template <class T>
struct add_const {
  typedef const T type;
};
```

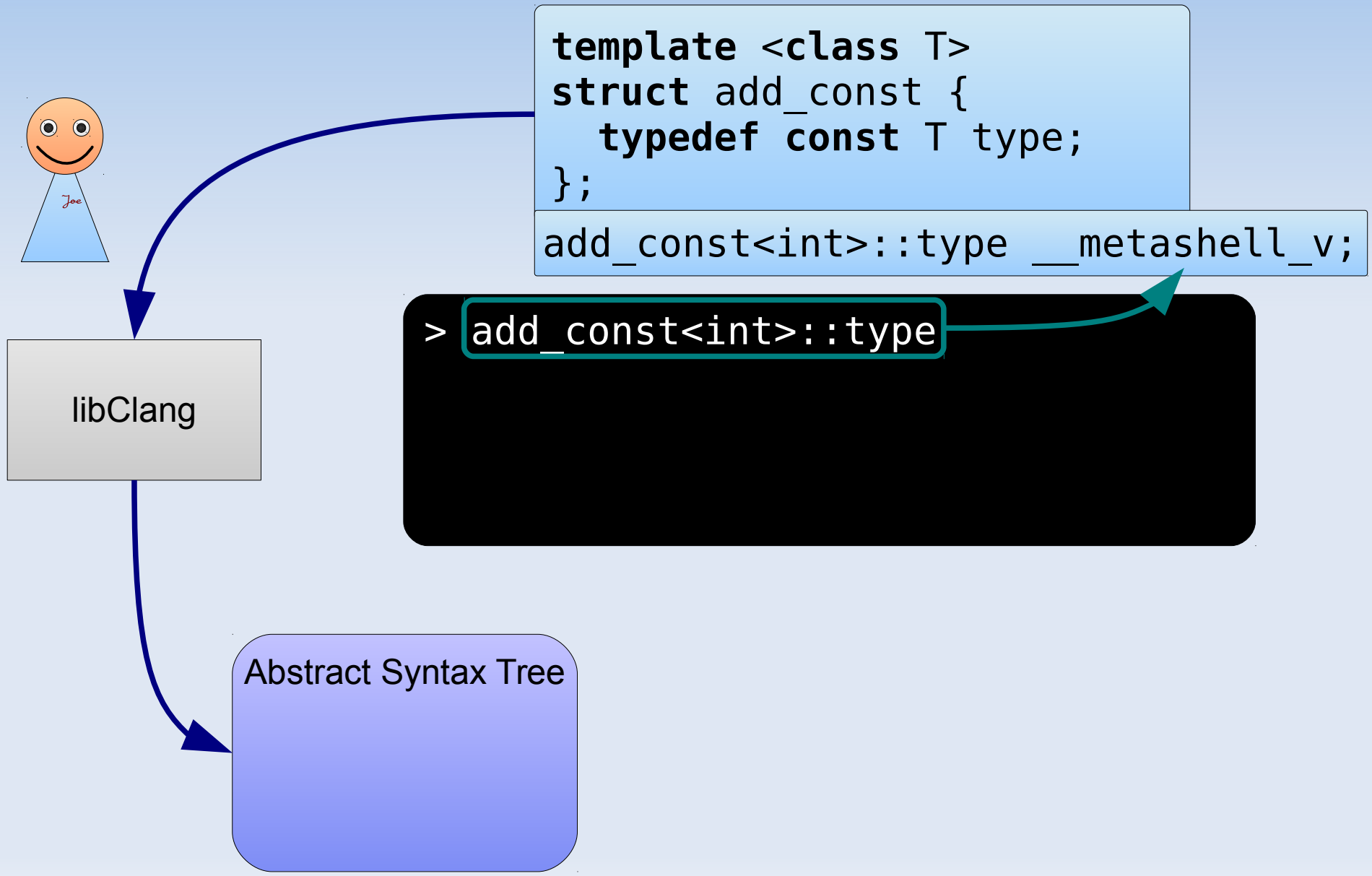
```
add_const<int>::type __metashell_v;
```

```
> add_const<int>::type
```

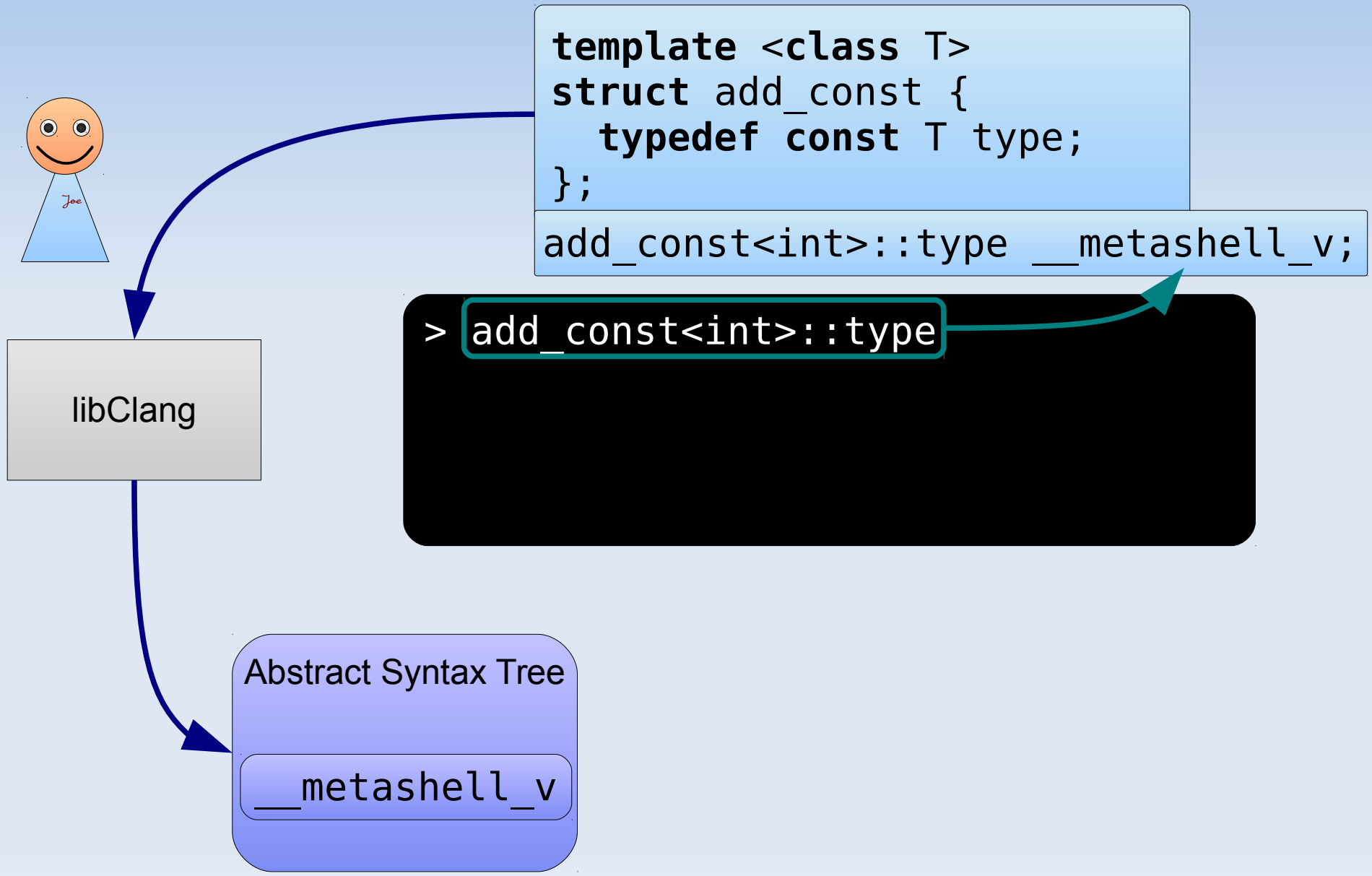
Template metaprogramming



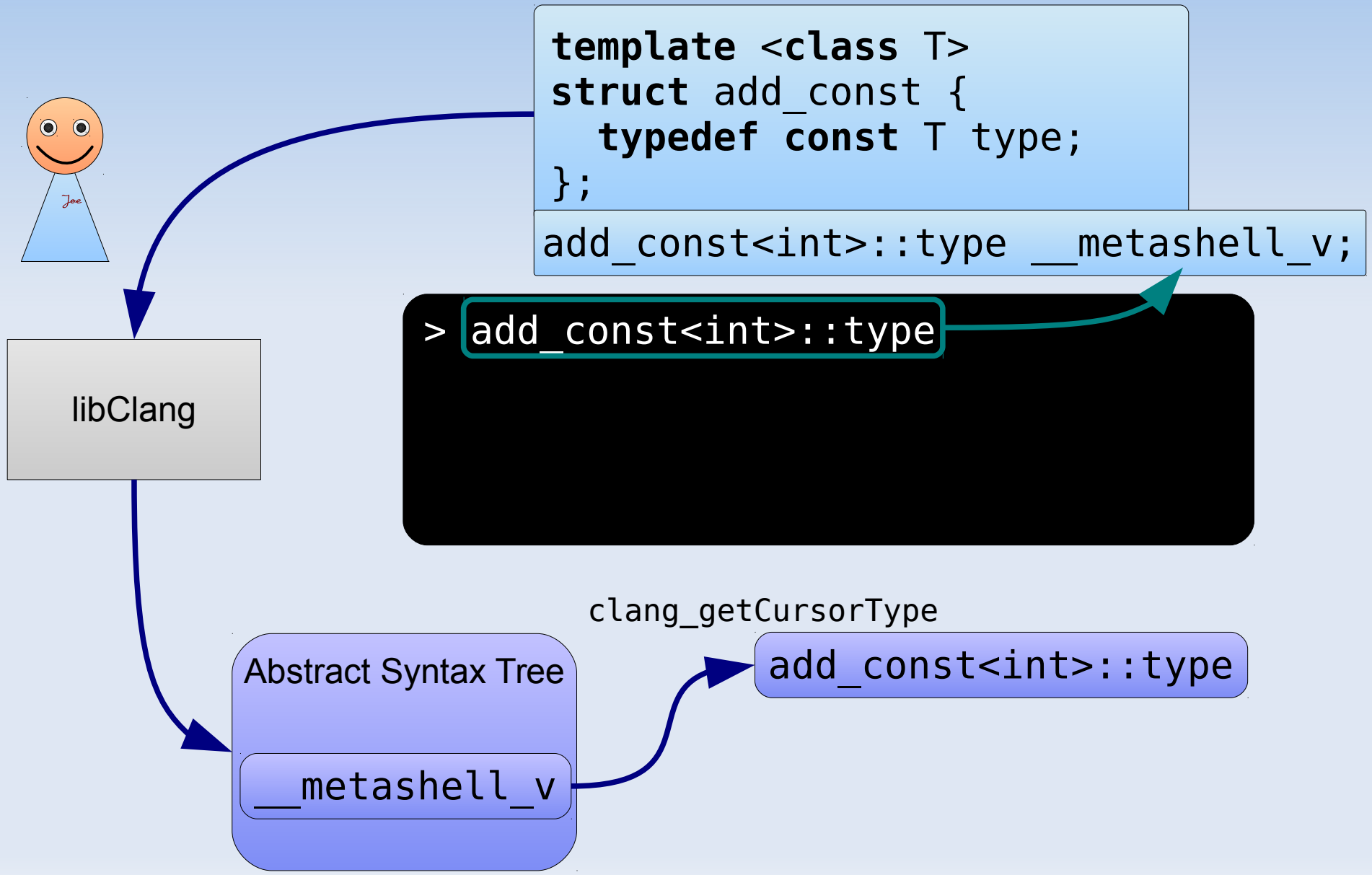
Template metaprogramming



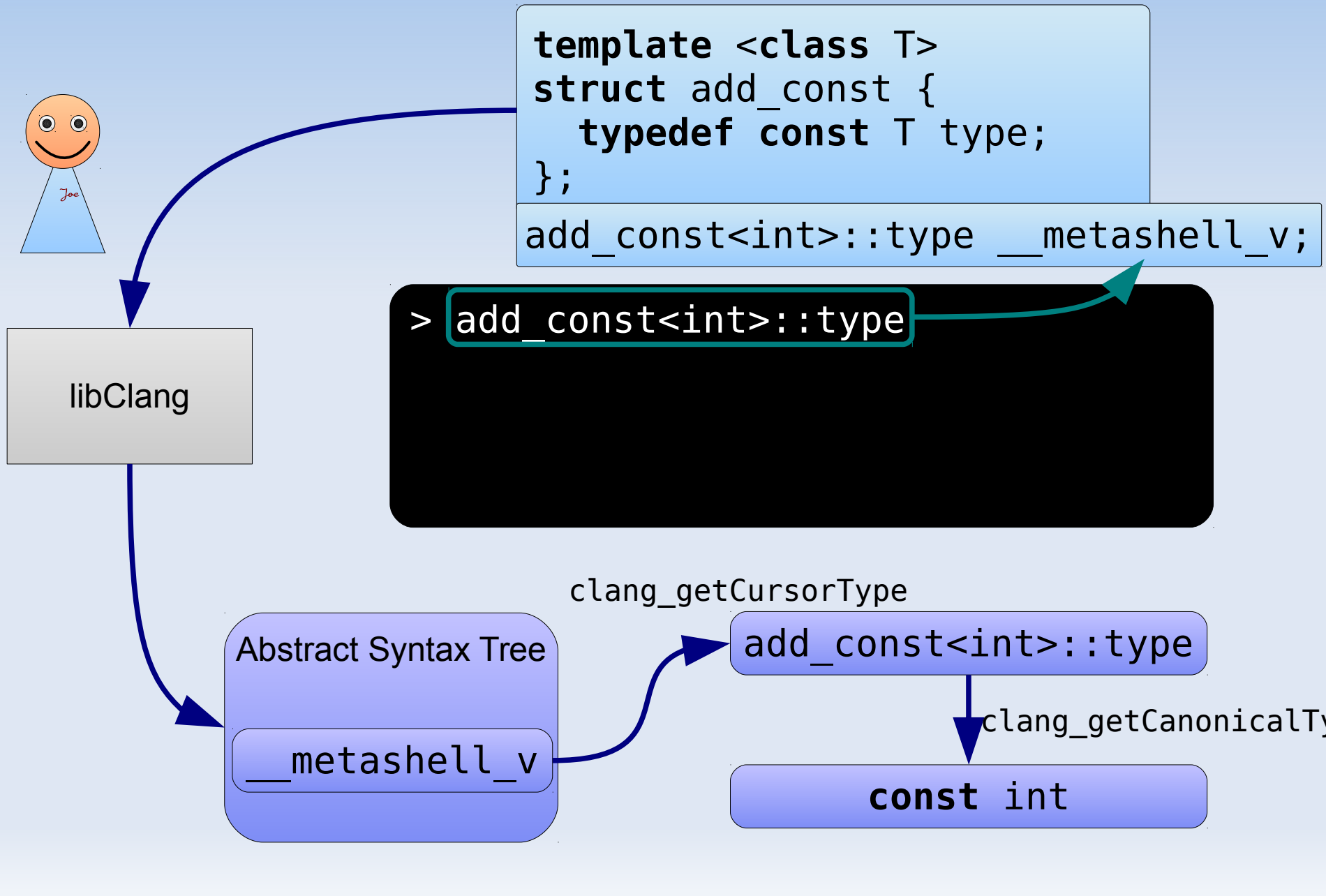
Template metaprogramming



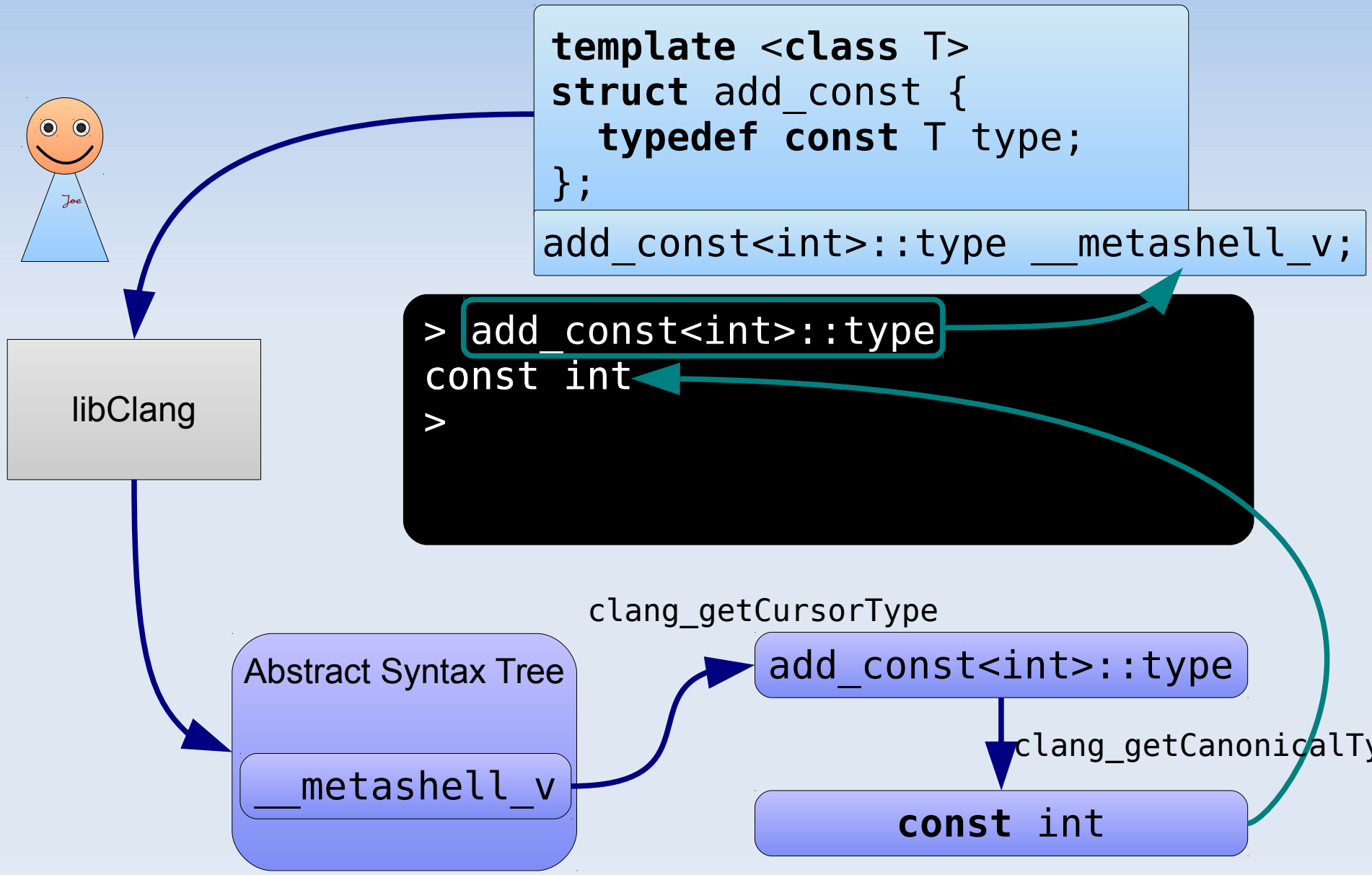
Template metaprogramming



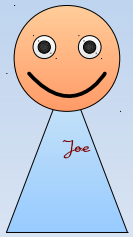
Template metaprogramming



Template metaprogramming

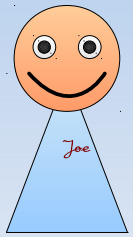


constexpr




```
constexpr int fact(int n) {
  return n == 0 ? 1 : n * fact(n – 1);
}
```

constexpr




```
constexpr int fact(int n) {
  return n == 0 ? 1 : n * fact(n – 1);
}
```

Syntax highlighting

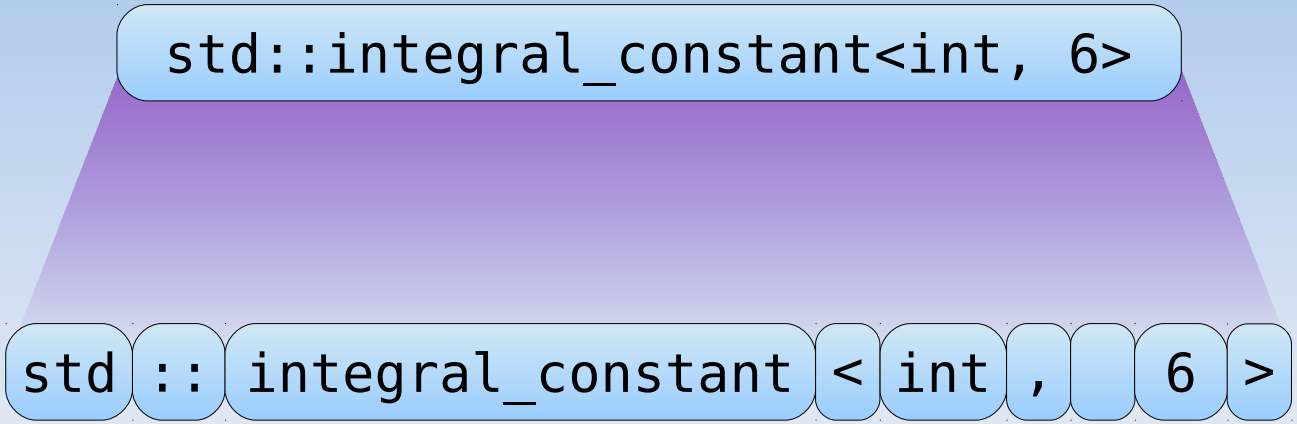
```
std::integral_constant<int, 6>
```

Syntax highlighting

```
std::integral_constant<int, 6>
```

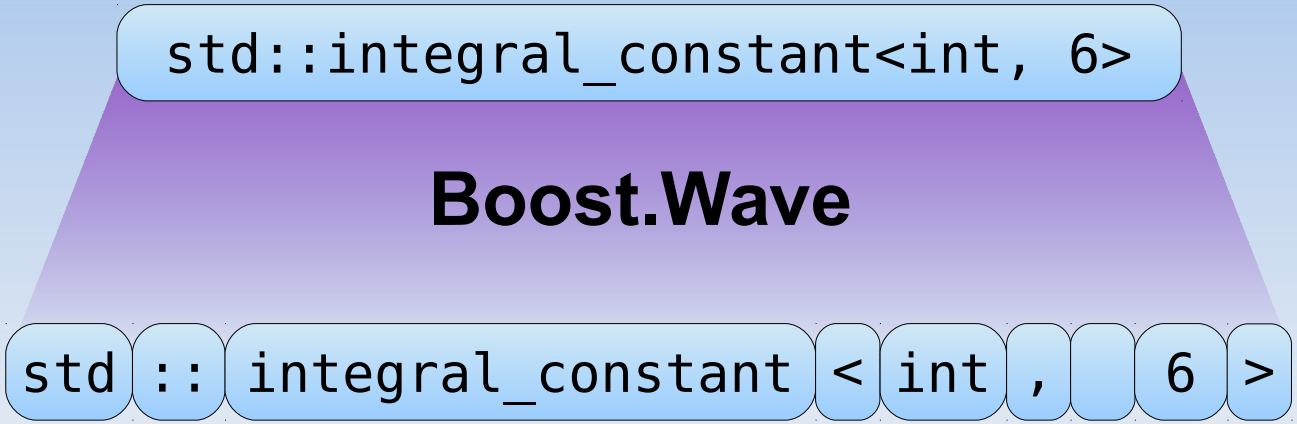
```
std::integral_constant<int, 6>
```

Syntax highlighting



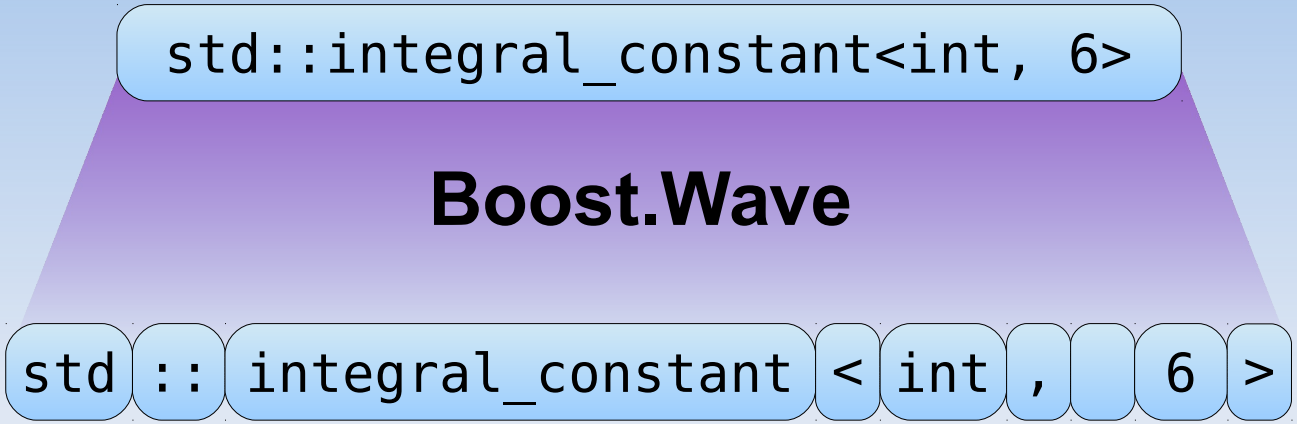

```
std::integral_constant<int, 6>
```

Syntax highlighting




```
std::integral_constant<int, 6>
```

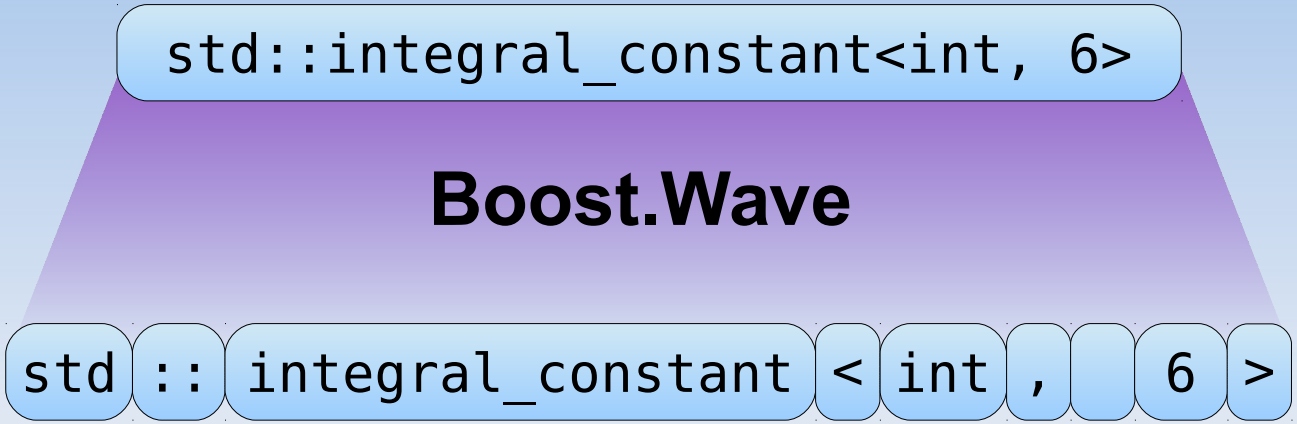
Syntax highlighting




```
for (token_iterator i = begin_tokens(s), e; i != e; ++i)
{




}
```

```
std::integral_constant<int, 6>
```

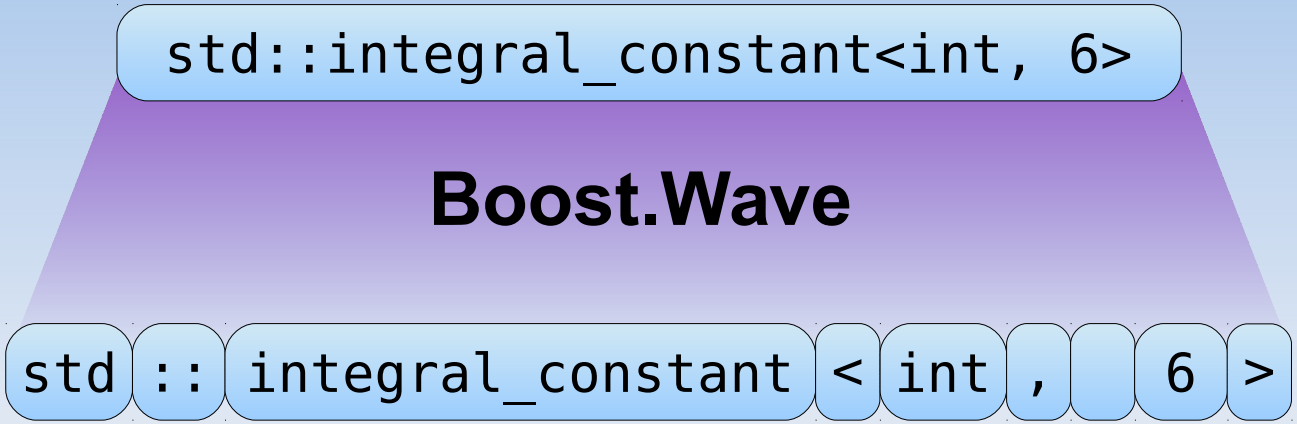
Syntax highlighting




```
for (token_iterator i = begin_tokens(s), e; i != e; ++i)
{




  std::cout << i->get_value();
}
```

```
std::integral_constant<int, 6>
```

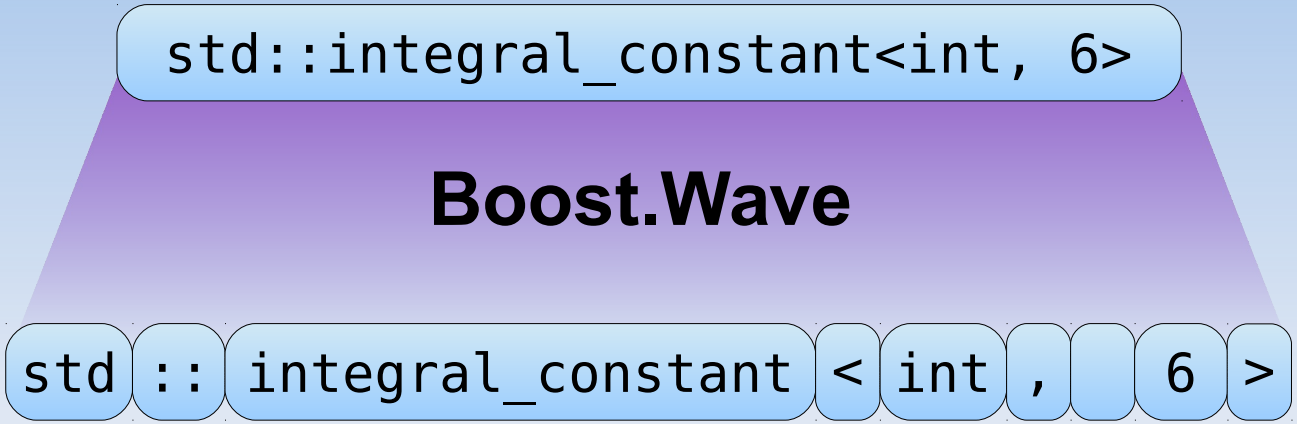
Syntax highlighting




```
for (token_iterator i = begin_tokens(s), e; i != e; ++i)
{
  if (IS_CATEGORY(*i, wave::IntegerLiteralTokenType))
  {
    set_color(purple);
  }

  std::cout << i->get_value();
}
```

```
std::integral_constant<int, 6>
```

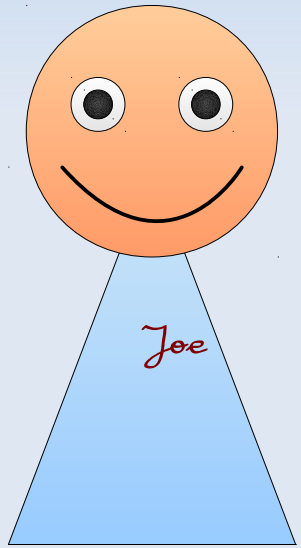
Syntax highlighting




```
for (token_iterator i = begin_tokens(s), e; i != e; ++i)
{
  if (IS_CATEGORY(*i, wave::IntegerLiteralTokenType))
  {
    set_color(purple);
  }
  // ...
  std::cout << i->get_value();
}
```

```
std::integral_constant<int, 6>
```

Agenda

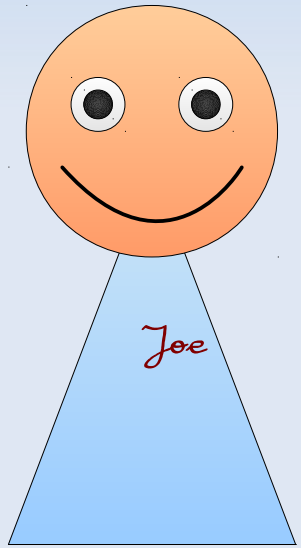


C++ developer

Interested in template metaprogramming

Advanced template metaprogrammer

Agenda

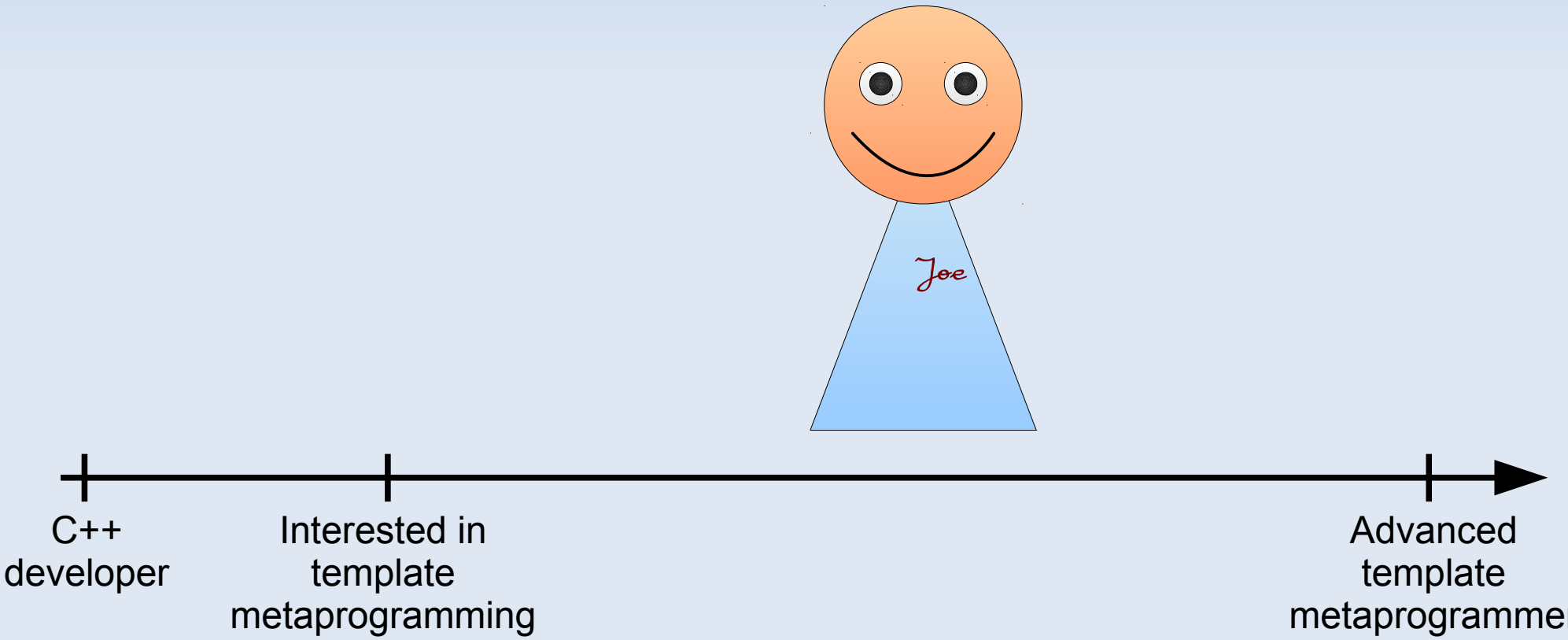


C++ developer

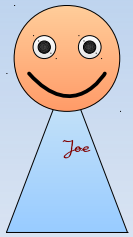
Interested in template metaprogramming

Advanced template metaprogrammer

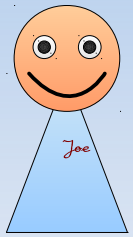
Agenda


Joe
C++ developer
Interested in template metaprogramming
Advanced template metaprogrammer

Boost.MPL



Boost.MPL



DEMO

Boost.MPL

```
namespace boost_ {
  namespace mpl {
    template <class... Ts> struct vector;
  }
}
```

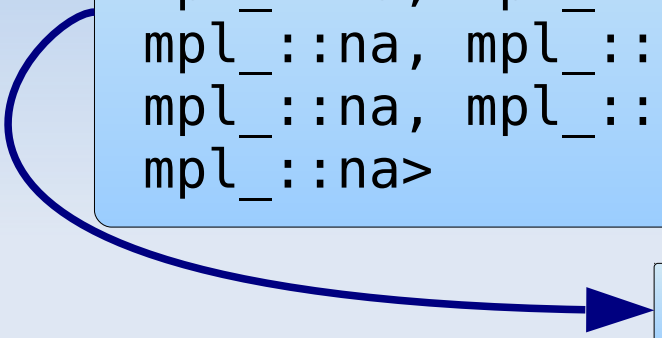
Boost.MPL

```
boost_::mpl::vector<int, char>
```

```
namespace boost_ {
  namespace mpl {
    template <class... Ts> struct vector;
  }
}
```

Boost.MPL

```
boost::mpl::vector<int, char, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na>
```

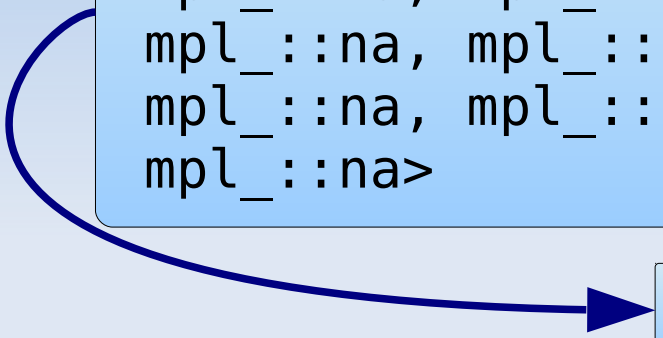


```
boost_::mpl::vector<int, char>
```

```
namespace boost_ {
  namespace mpl {
    template <class... Ts> struct vector;
  }
}
```

Boost.MPL

```
boost::mpl::vector<int, char, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na>
```



```
boost_::mpl::vector<int, char>
```

```
boost::mpl::v_item<double, boost::mpl::vector
<int, char, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na>, 1>
```

Boost.MPL

```
boost::mpl::vector<int, char, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na>
```

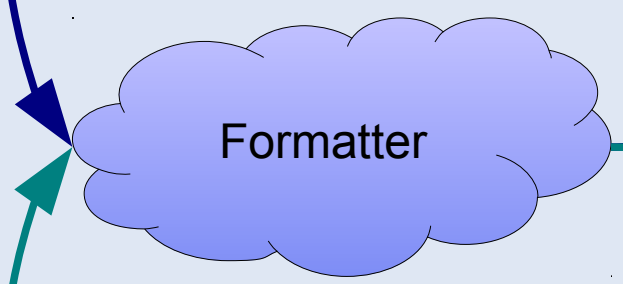
```
boost_::mpl::vector<int, char>
```

```
boost_::mpl::vector<double, int, char>
```

```
boost::mpl::v_item<double, boost::mpl::vector
<int, char, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na>, 1>
```

Boost.MPL

```
boost::mpl::vector<int, char, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na>
```

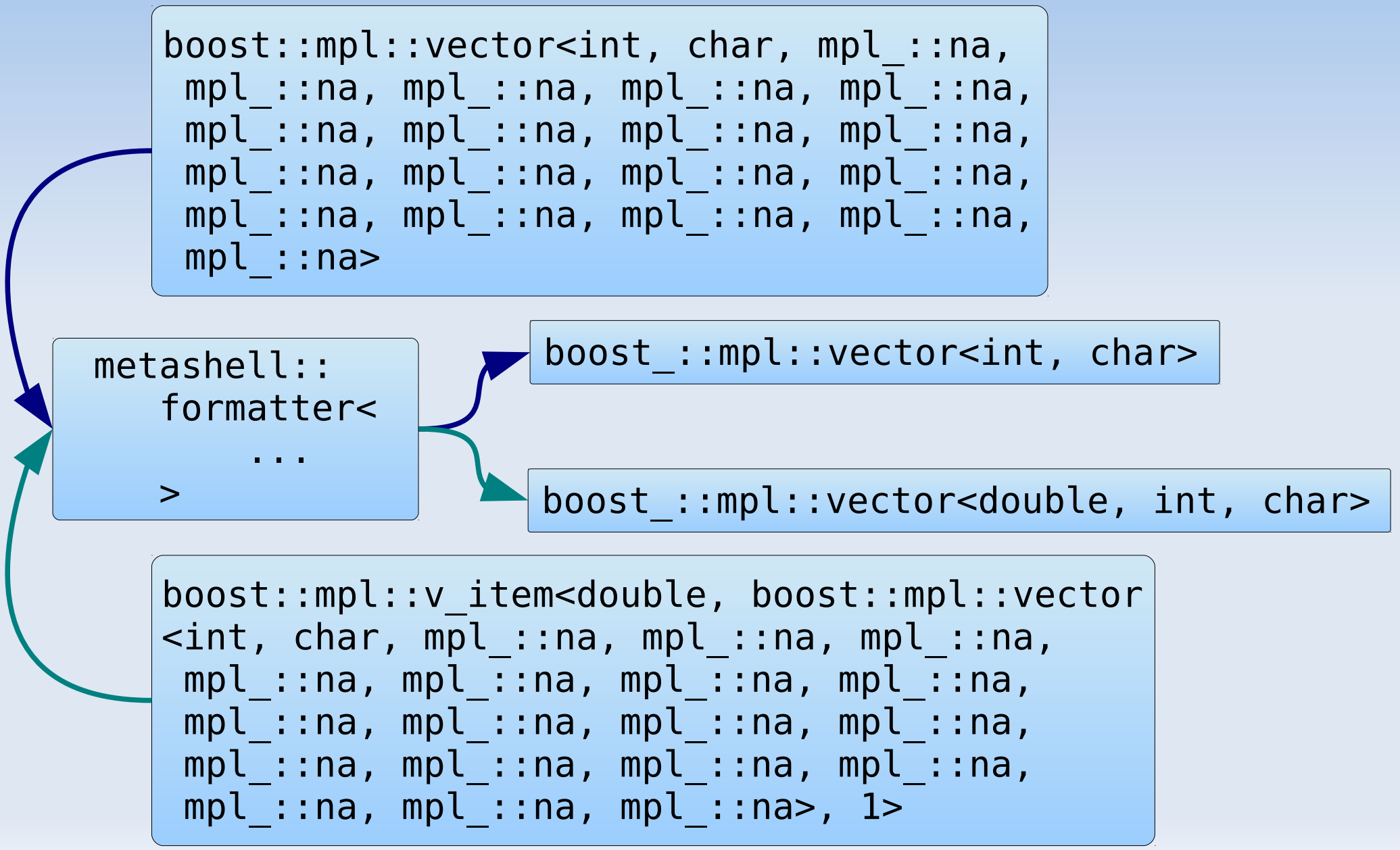



```
boost_::mpl::vector<int, char>
```

```
boost_::mpl::vector<double, int, char>
```

```
boost::mpl::v_item<double, boost::mpl::vector
<int, char, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na>, 1>
```

Boost.MPL


boost::mpl::vector<int, char, mpl_::na,
mpl_::na, mpl_::na, mpl_::na, mpl_::na,
mpl_::na, mpl_::na, mpl_::na, mpl_::na,
mpl_::na, mpl_::na, mpl_::na, mpl_::na,
mpl_::na, mpl_::na, mpl_::na, mpl_::na,
mpl_::na>
metashell::
formatter<
...
>
boost_::mpl::vector<int, char>
boost_::mpl::vector<double, int, char>
boost::mpl::v_item<double, boost::mpl::vector
<int, char, mpl_::na, mpl_::na, mpl_::na,
mpl_::na, mpl_::na, mpl_::na, mpl_::na,
mpl_::na, mpl_::na, mpl_::na, mpl_::na,
mpl_::na, mpl_::na, mpl_::na, mpl_::na,
mpl_::na, mpl_::na, mpl_::na>, 1>

Boost.MPL

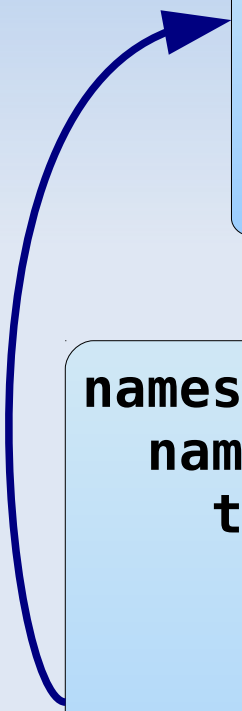
```
boost::mpl::vector<int, char, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na>
```

```
boost_::mpl::vector<int, char>
```

Boost.MPL

```
boost::mpl::vector<int, char, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na>
```

```
boost_::mpl::vector<int, char>
```



```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}
```

Boost.MPL

```
boost::mpl::vector<int, char, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na, mpl_::na, mpl_::na, mpl_::na,
 mpl_::na>
```

```
boost_::mpl::vector<int, char>
```

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}
```

```
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
```

Boost.MPL

”real” implementation of the library
works with old compilers

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}
```

```
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
```

Boost.MPL

”real” implementation of the library
works with old compilers


```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}
```

addition to the library
used for ”pretty-printing” only

```
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
```

Boost.MPL

vector.hpp

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}
```

```
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
```

Boost.MPL

vector.hpp

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}
```

```
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
```

Boost.MPL

vector.hpp

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}
```

```
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
```

Boost.MPL

vector.hpp

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}
```

```
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
```

Boost.MPL

vector.hpp

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}


namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
```

Boost.MPL

vector.hpp

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}
```

```
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
```

Needs variadic templates

Boost.MPL

vector.hpp

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}

#if Used in Metashell
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
#endif
```

Needs variadic templates

Boost.MPL

vector.hpp

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}

#if defined __METASHELL
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
#endif
```

Needs variadic templates

Boost.MPL

vector.hpp

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}

#if defined __METASHELL
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
#endif
```

```
> #include <metashell/formatter/vector.hpp>
> #include <metashell/formatter/list.hpp>
// …
```

Needs variadic templates

Boost.MPL

vector.hpp

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}

#if defined __METASHELL
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
#endif
```

```
> #include <metashell/formatter/vector.hpp>
> #include <metashell/formatter/list.hpp>
// …
> #include <metashell/formatter.hpp>
```

Needs variadic templates

Boost.MPL

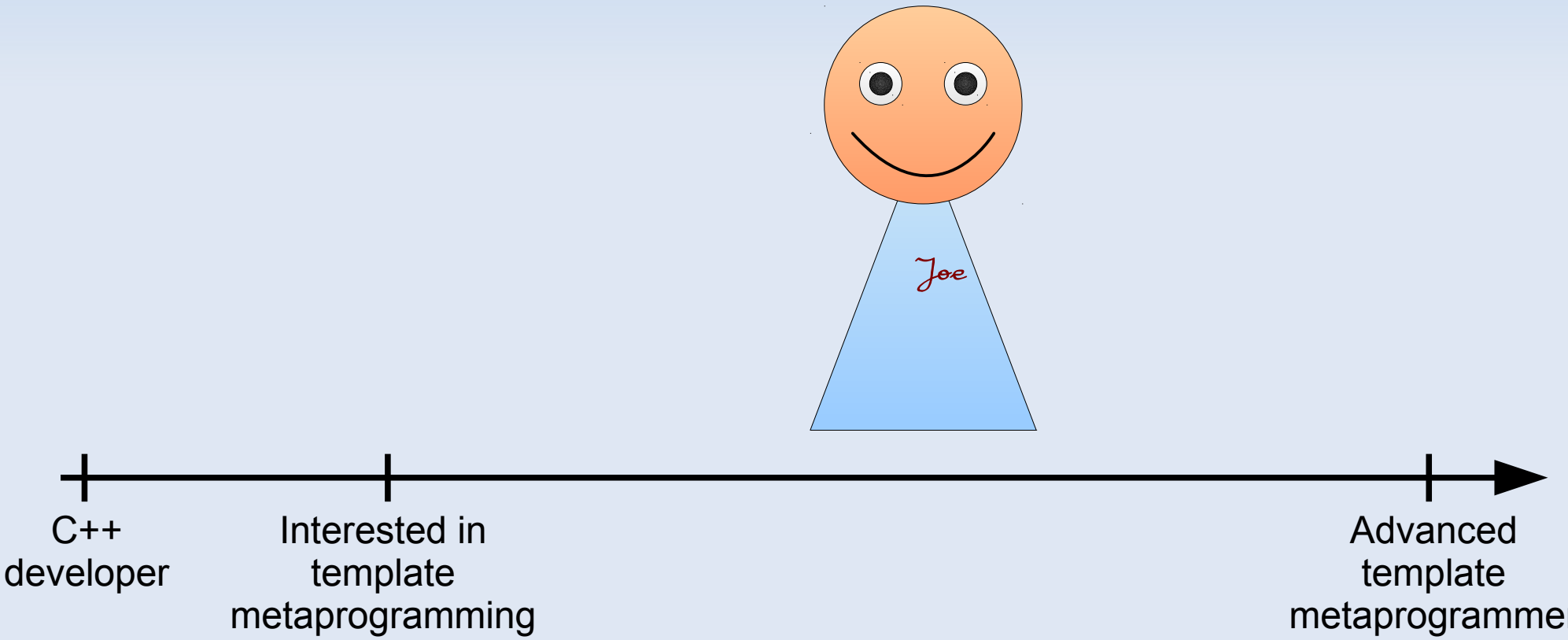
vector.hpp

```
namespace boost {
  namespace mpl {
    template <
      class T0 = mpl_::na,
      class T1 = mpl_::na,
      // ...
      class T20 = mpl_::na
    >
    struct vector;
  }
}

#if defined __METASHELL
namespace boost_ {
  namespace mpl {
    template <class... Ts>
    struct vector;
  }
}
#endif
```

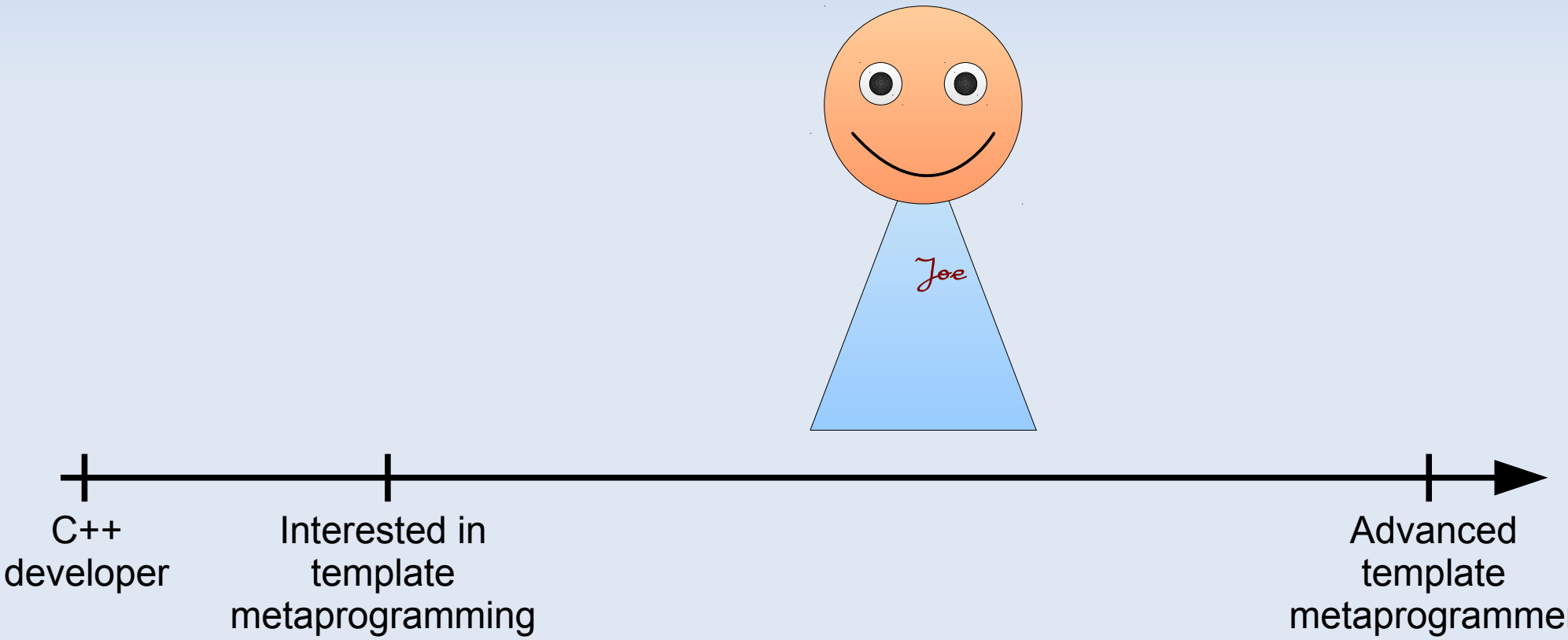

```
> #include <metashell/formatter/vector.hpp>
> #include <metashell/formatter/list.hpp>
// …
> #include <metashell/formatter.hpp>
```

Needs variadic templates

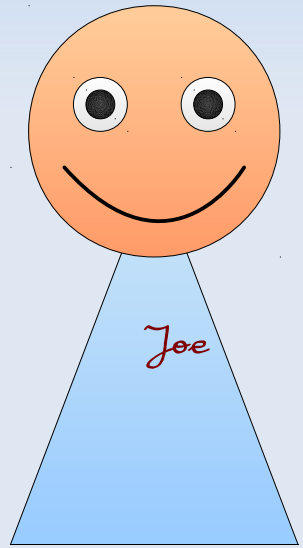
Agenda


Joe
C++ developer
Interested in template metaprogramming
Advanced template metaprogrammer

Agenda


Joe
C++ developer
Interested in template metaprogramming
Advanced template metaprogrammer

Agenda



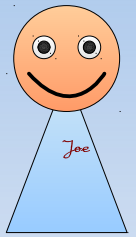
C++ developer

Interested in template metaprogramming

Advanced template metaprogrammer

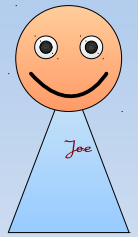
Custom data-type: list

```
struct nil { typedef nil type; };
```



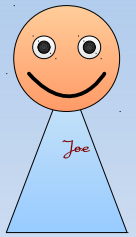
Custom data-type: list

```
struct nil { typedef nil type; };

template <class Head, class Tail>
struct cons { typedef cons type; };
```



Custom data-type: list

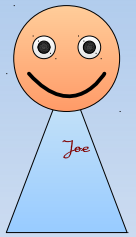
```
struct nil { typedef nil type; };

template <class Head, class Tail>
struct cons { typedef cons type; };
```




```
// [int, char, int_<13>]
```

Custom data-type: list

```
struct nil { typedef nil type; };

template <class Head, class Tail>
struct cons { typedef cons type; };
```

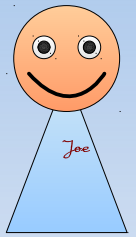


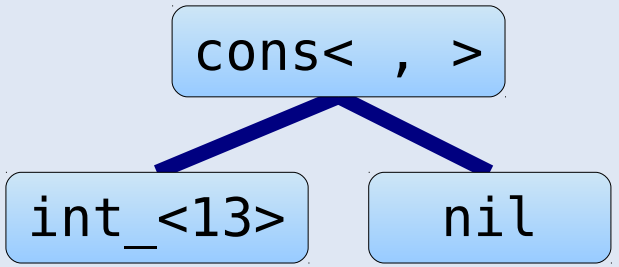
nil

```
// [int, char, int_<13>]
                                        nil
```

Custom data-type: list

```
struct nil { typedef nil type; };

template <class Head, class Tail>
struct cons { typedef cons type; };
```

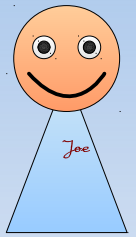


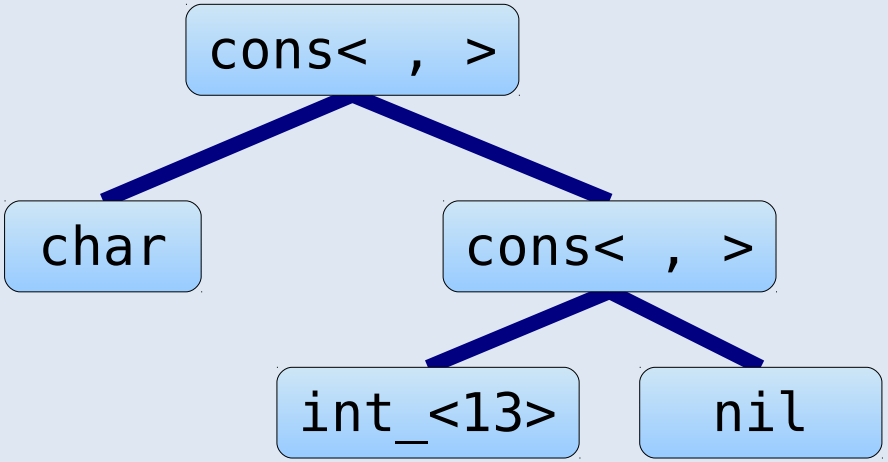



```
// [int, char, int_<13>]
                        cons<int_<13>, nil>
```

Custom data-type: list

```
struct nil { typedef nil type; };

template <class Head, class Tail>
struct cons { typedef cons type; };
```

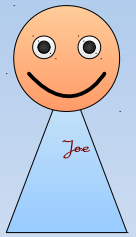


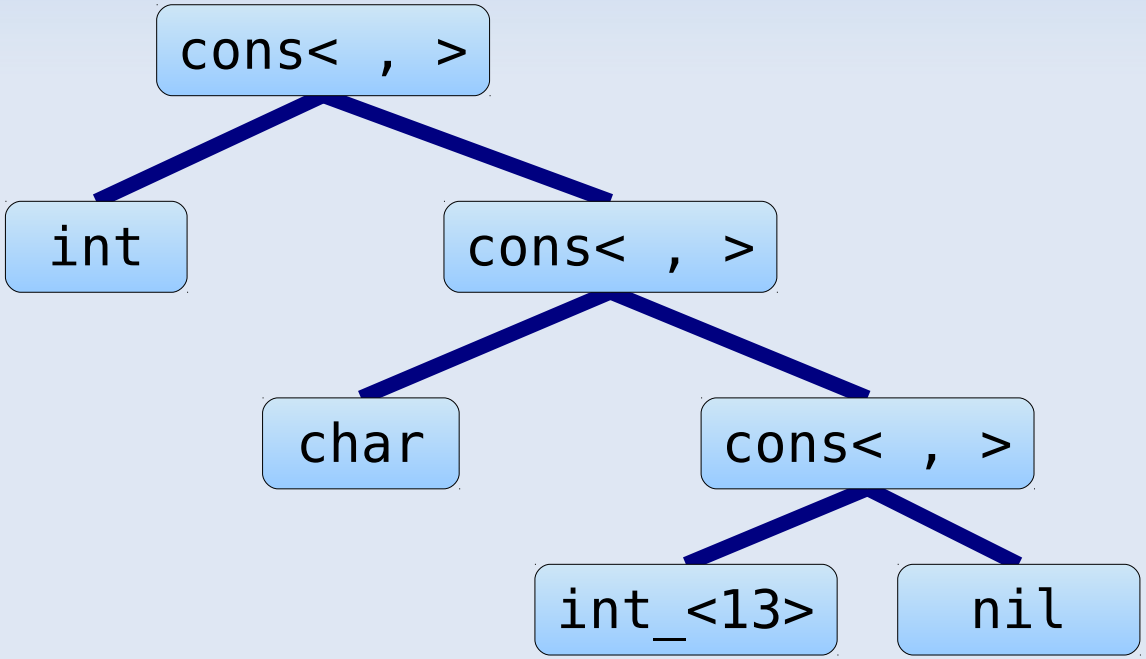



```
// [int, char, int_<13>]
          cons<char, cons<int_<13>, nil>>
```

Custom data-type: list

```
struct nil { typedef nil type; };

template <class Head, class Tail>
struct cons { typedef cons type; };
```

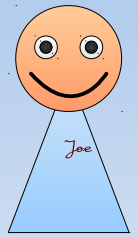


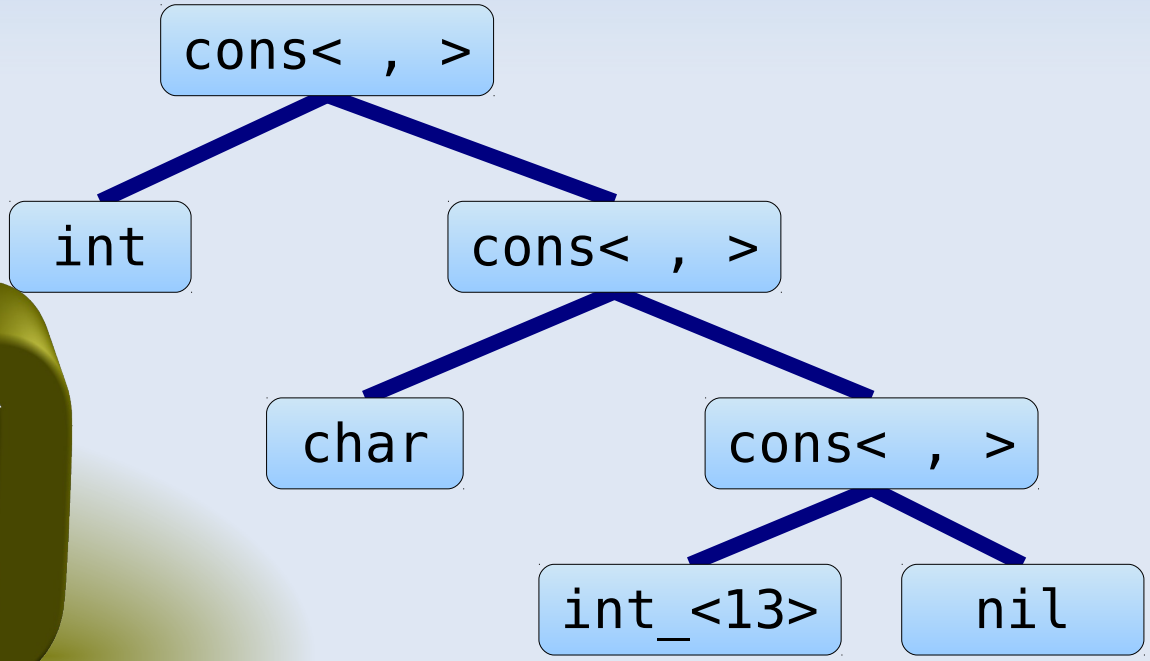



```
// [int, char, int_<13>]
cons<int, cons<char, cons<int_<13>, nil>>>
```

Custom data-type: list

```
struct nil { typedef nil type; };

template <class Head, class Tail>
struct cons { typedef cons type; };
```





DEMO

```
// [int, char, int_<13>]
cons<int, cons<char, cons<int_<13>, nil>>>
```

What else is it good for?

What is the type of x?

What else is it good for?

What is the type of x?

```
auto x = some_function(11.13, "03", 21);
```

What else is it good for?

What is the type of x?

```
auto x = some_function(11.13, "03", 21);
```

```
a<double>::handle x;
```

What else is it good for?

What is the type of x?

```
auto x = some_function(11.13, "03", 21);
```


```
a<double>::handle x;
```

Getting metashell

- https://github.com/sabel83/metashell
  - Getting the source code
  - Pre-built binaries

Getting metashell

- https://github.com/sabel83/metashell
  - Getting the source code
  - Pre-built binaries
- http://abel.web.elte.hu/shell
  - Trying out online

Getting metashell

- https://github.com/sabel83/metashell
  - Getting the source code
  - Pre-built binaries
- http://abel.web.elte.hu/shell
  - Trying out online

Challenges

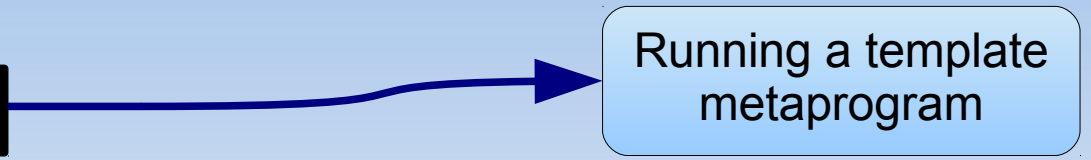
```
>
```

Challenges

```
> add_const<int>::type
```

Challenges

```
> add_const<int>::type
```



Challenges

```
> add_const<int>::type
```

Running a template metaprogram

```
> #include <type_traits>
```

Challenges

```
> add_const<int>::type
```

Running a template metaprogram

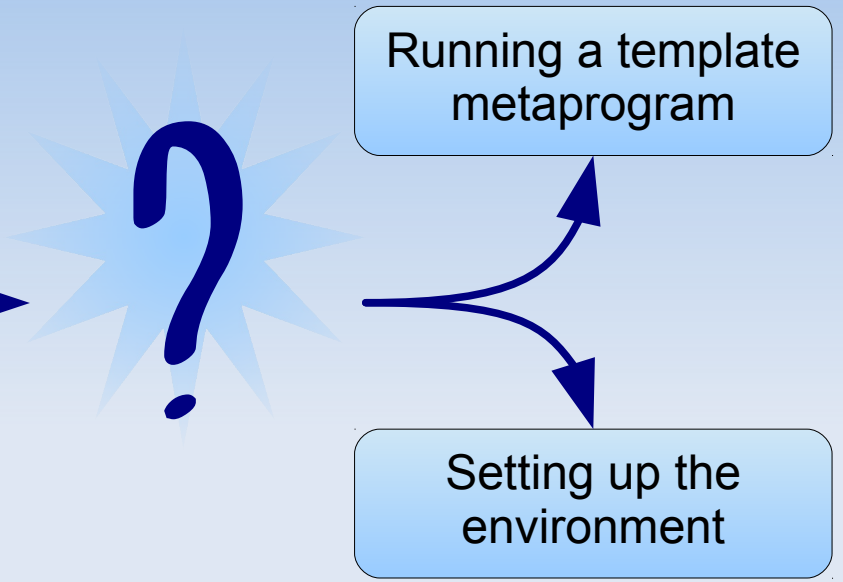
```
> #include <type_traits>
```

Setting up the environment

Challenges

```
> add_const<int>::type
```

```
> #include <type_traits>
```

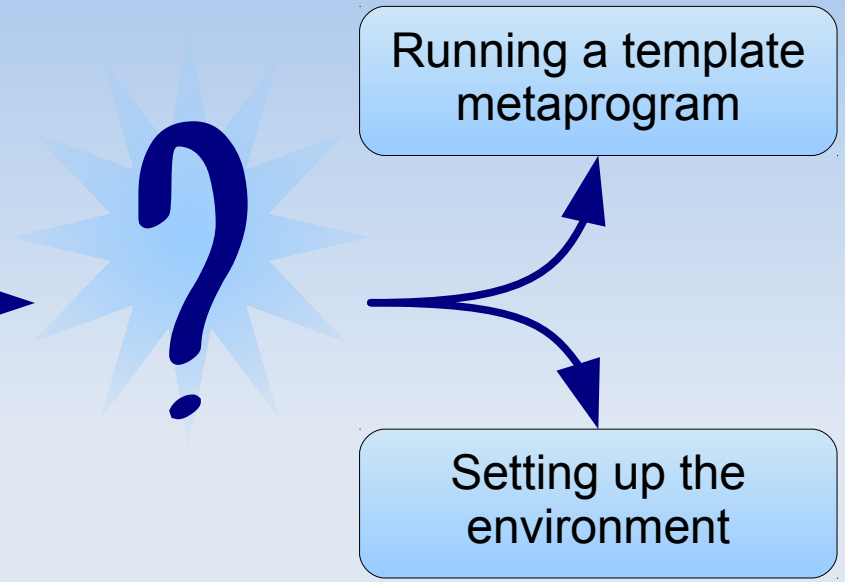


Challenges

```
> add_const<int>::type
```

```
> #include <type_traits>
```

```
> using namespace boost::mpl;
```

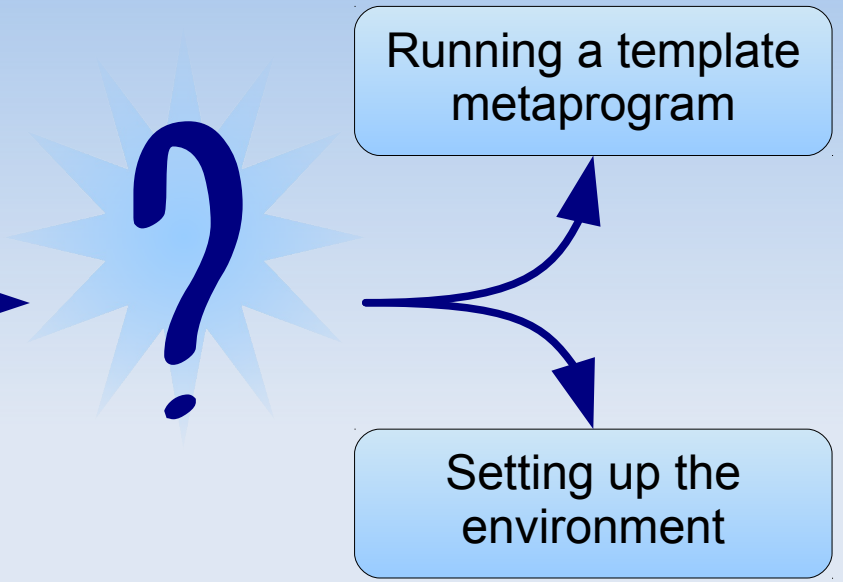


Challenges

```
> add_const<int>::type
```

```
> #include <type_traits>
```

```
> using namespace boost::mpl;
```




```
> template <class T> struct add_const {typedef const T type;};
```

Challenges

```
> add_const<int>::type
```

```
> #include <type_traits>
```

```
> using namespace boost::mpl;
```

Running a template metaprogram

Setting up the environment

Current solution: ad-hoc rules....

```
> template <class T> struct add_const {typedef const T type;};
```

Challenges

```
> add_const<int>::type
```

```
> #include <type_traits>
```

```
> using namespace boost::mpl;
```

Running a template metaprogram

Setting up the environment

Current solution: ad-hoc rules....

- Parsing the line?

```
> template <class T> struct add_const {typedef const T type;};
```

Challenges

```
> add_const<int>::type
```

```
> #include <type_traits>
```

```
> using namespace boost::mpl;
```

Running a template metaprogram

Setting up the environment

Current solution: ad-hoc rules....

- Parsing the line?
- Asking Clang if this is a type?

```
> template <class T> struct add_const {typedef const T type;};
```

Challenges

```
> add_const<int>::type
```

```
> #include <type_traits>
```

```
> using namespace boost::mpl;
```

Running a template metaprogram

Setting up the environment

Current solution: ad-hoc rules....

- Parsing the line?
- Asking Clang if this is a type?
- << *your idea goes here....* >> ?

```
> template <class T> struct add_const {typedef const T type;};
```

Challenges

```
> add_const<int>::type
```

```
> #include <type_traits>
```

```
> using namespace boost::mpl;
```

Running a template metaprogram

Setting up the environment

Current solution: ad-hoc rules....

- Parsing the line?
- Asking Clang if this is a type?
- << *your idea goes here....* >> ?

```
> template <class T> struct add_const {typedef const T type;};
```

Setting up the environment

Environment

```
#define __METASHELL

// …
```

Setting up the environment

Environment

```
#define __METASHELL

// …
```

Setting up the environment

Environment

```
#define __METASHELL

// …
```

```
> #inculde <type_traits>
```

Setting up the environment

Environment

```
#define __METASHELL

// …
```

```
> #inculde <type_traits>
```

```
#define __METASHELL

// …
```

Setting up the environment

Environment

```
#define __METASHELL

// …
```

```
> #inculde <type traits>
```

```
#define __METASHELL

// …



#inculde <type_traits>
```

Setting up the environment

Environment

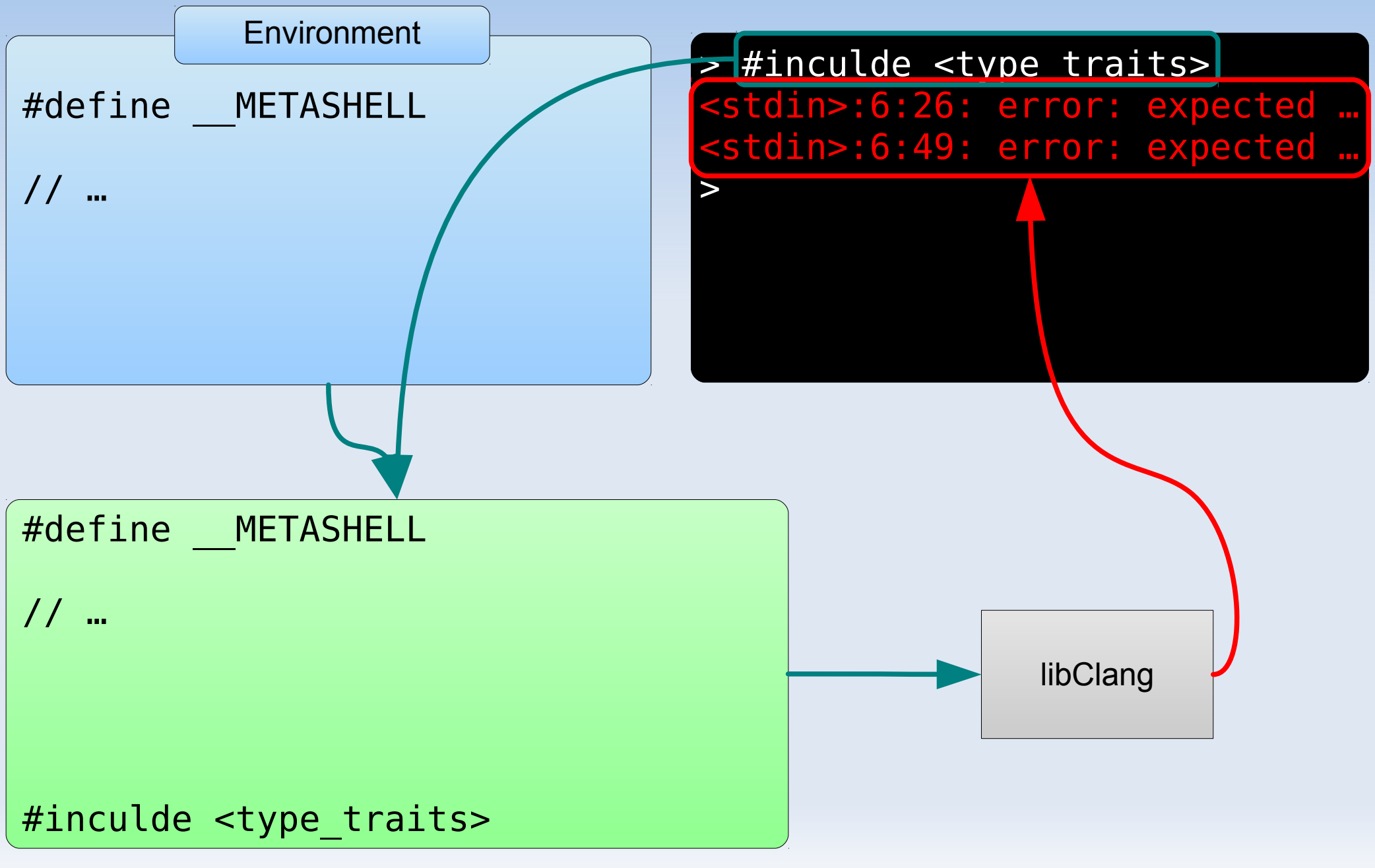
```
#define __METASHELL

// …
```

```
> #inculde <type traits>
```

```
#define __METASHELL

// …



#inculde <type_traits>
```

libClang

Setting up the environment



Setting up the environment

Environment

```
#define __METASHELL

// …
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
```

libClang

Setting up the environment

Environment

```
#define __METASHELL

// …
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
```

```
#define __METASHELL

// …
```

libClang

Setting up the environment

Environment

```
#define __METASHELL

// …
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
```

```
#define __METASHELL

// …



#include <type_traits>
```

libClang

Setting up the environment

Environment

```
#define __METASHELL

// …
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
```

```
#define __METASHELL

// …



#include <type_traits>
```

libClang

Setting up the environment

Environment

```
#define __METASHELL

// …

#include <type_traits>
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
>
```

```
#define __METASHELL

// …



#include <type_traits>
```

libClang

Setting up the environment

Environment

```
#define __METASHELL

// …

#include <type_traits>
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
> template <class T> struct …
```

libClang

Setting up the environment

Environment

```
#define __METASHELL

// …

#include <type_traits>
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
> template <class T> struct …
```

```
#define __METASHELL

// …

#include <type_traits>
```

libClang

Setting up the environment

Environment

```
#define __METASHELL

// …

#include <type_traits>
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
> template <class T> struct …
```

```
#define __METASHELL

// …

#include <type_traits>


template <class T> struct …
```

libClang

Setting up the environment

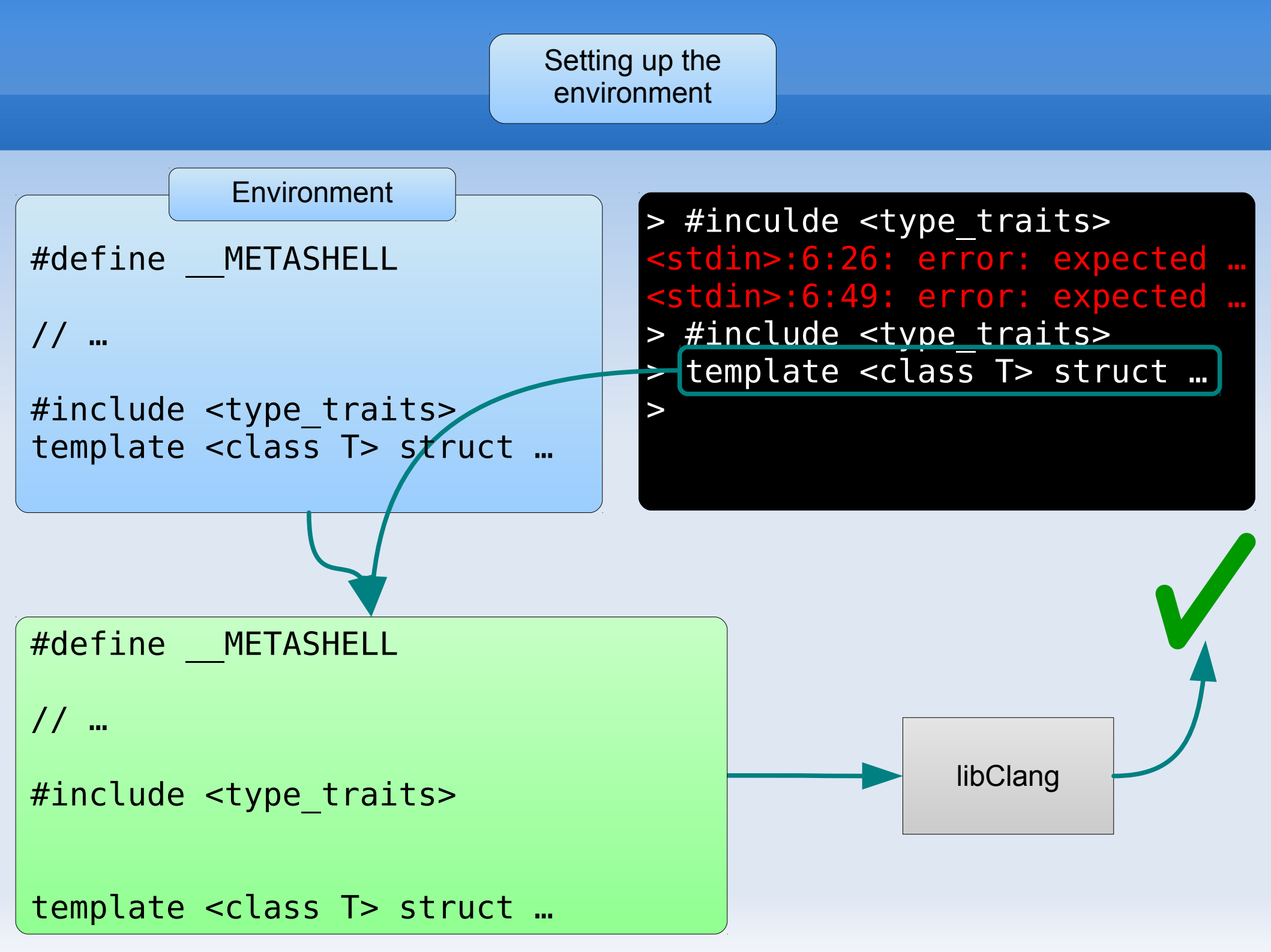
Environment

```
#define __METASHELL

// …

#include <type_traits>
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
> template <class T> struct …
```

```
#define __METASHELL

// …

#include <type_traits>


template <class T> struct …
```

libClang


Setting up the environment
Environment
#define __METASHELL
// …
#include <type_traits>
template <class T> struct …
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
> template <class T> struct …
>
#define __METASHELL
// …
#include <type_traits>
template <class T> struct …
libClang

Setting up the environment

Environment

```
#define __METASHELL

// …

#include <type_traits>
template <class T> struct …
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
> template <class T> struct …
> add_const<int>::type
```

libClang

Setting up the environment

Environment

```
#define __METASHELL

// …

#include <type_traits>
template <class T> struct …
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
> template <class T> struct …
> add_const<int>::type
```

```
#define __METASHELL

// …

#include <type_traits>
template <class T> struct …
```

libClang

Setting up the environment

Environment

```
#define __METASHELL

// …

#include <type_traits>
template <class T> struct …
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
> template <class T> struct …
> add_const<int>::type
```

```
#define __METASHELL

// …

#include <type_traits>
template <class T> struct …

add_const<int>::type __metashell_v;
```

libClang

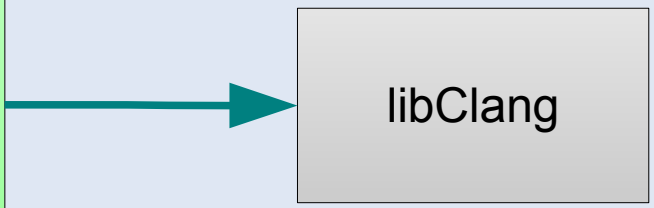
Setting up the environment

Environment

```
#define __METASHELL

// …

#include <type_traits>
template <class T> struct …
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
> template <class T> struct …
> add_const<int>::type
```

```
#define __METASHELL

// …

#include <type_traits>
template <class T> struct …

add_const<int>::type __metashell_v;
```



Setting up the environment

Environment

```
#define __METASHELL

// …

#include <type_traits>
template <class T> struct …
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
> template <class T> struct …
> add_const<int>::type
const int
>
```

```
#define __METASHELL

// …

#include <type_traits>
template <class T> struct …

add_const<int>::type __metashell_v;
```

libClang

Setting up the environment

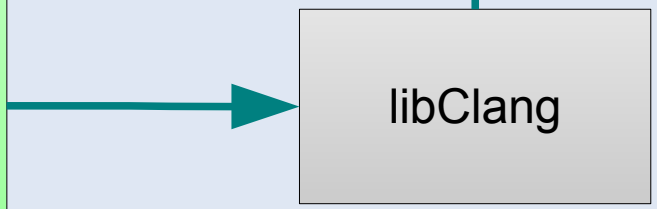
Environment

```
#define __METASHELL

// …

#include <type_traits>
template <class T> struct …
```

```
> #inculde <type_traits>
<stdin>:6:26: error: expected …
<stdin>:6:49: error: expected …
> #include <type_traits>
> template <class T> struct …
> add_const<int>::type
const int
>
```

The whole thing is compiled over and over again...

```
#define __METASHELL

// …

#include <type_traits>
template <class T> struct …

add_const<int>::type __metashell_v;
```



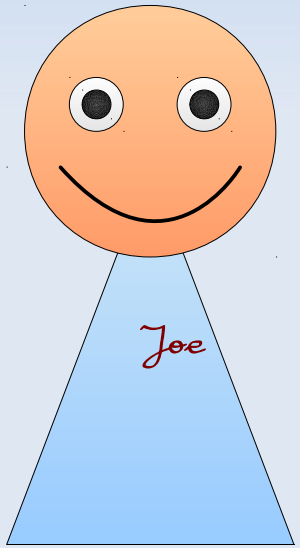
What is next?

- Looking into metaprogram execution
  - Templight integration

What is next?

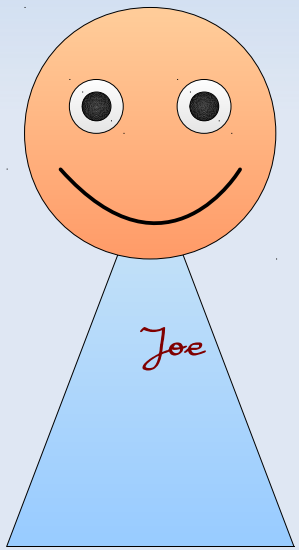
- Looking into metaprogram execution
  - Templight integration
- Windows build

A story for another day...



C/C++
developer

A story for another day...

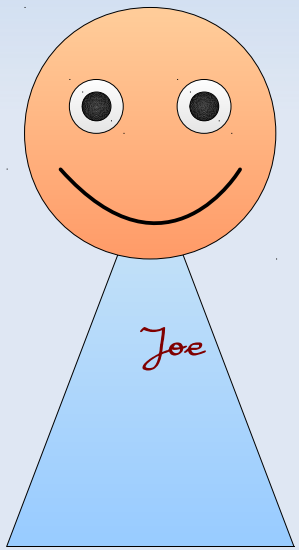


C/C++ developer

Interested in preprocessor metaprogramming

Advanced preprocessor metaprogrammer

A story for another day...



C/C++ developer

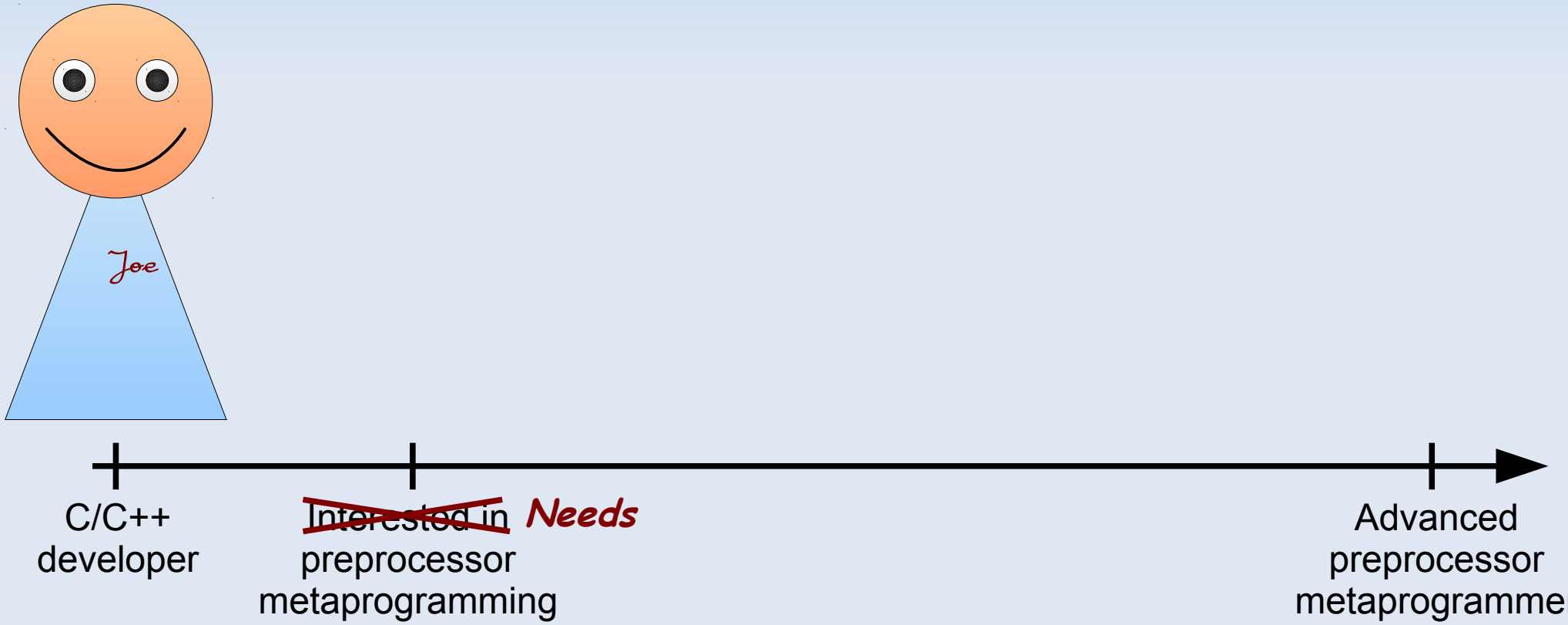
~~Interested in~~ preprocessor metaprogramming

Needs

Advanced preprocessor metaprogrammer

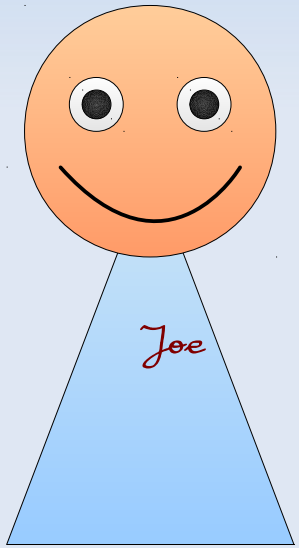
A story for another day...

```
> template <BOOST_PP_ENUM_PARAMS(3, class T)> struct vector;
```



A story for another day...

```
> template <BOOST_PP_ENUM_PARAMS(3, class T)> struct vector;
template <class T0  , class T1  , class T2> struct vector;
>
```



C/C++ developer

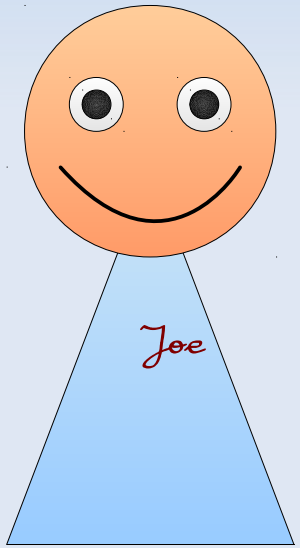
~~Interested in~~ preprocessor metaprogramming

Needs

Advanced preprocessor metaprogrammer

A story for another day...

```
> template <BOOST_PP_ENUM_PARAMS(3, class T)> struct vector;
template <class T0  , class T1  , class T2> struct vector;
>
```



Preshell

C/C++ developer

~~Interested in~~ preprocessor metaprogramming

Needs

Advanced preprocessor metaprogrammer

Q & A

abel@sinkovics.hu

Metashell: http://github.com/sabel83/metashell
Preshell: http://github.com/sabel83/preshell

Online demo: http://abel.web.elte.hu/shell