Post-Modern TODOs

Copyright © 2019 Daniel Frey

Typical TODOs

```
int main()
{
    // TODO: World domination!
    // TODO: Get a cat
}
```

Copyright © 2019 Daniel Frey

Typical Problem

- “We’ll fix them later!”
- …and “later” never happens.
- In my experience, every sufficiently large enough code base contains a lot of TODOs that no one ever works on.
- How about an expiry date?
- Luckily, the compiler knows the current `__DATE__`.

Copyright © 2019 Daniel Frey

```
constexpr unsigned digit( const std::string_view s, const unsigned i )
{
   return ( s[ i ] >= '0' && s[ i ] <= '9' )
      ? ( s[ i ] - '0' ) : throw "invalid character";
}

constexpr unsigned two( const std::string_view s, const unsigned i )
{
   return digit( s, i ) * 10 + digit( s, i + 1 );
}

constexpr unsigned four( const std::string_view s, const unsigned i )
{
   return two( s, i ) * 100 + two( s, i + 2 );
}
```

Copyright © 2019 Daniel Frey

```
// __DATE__ => "Jul 16 2019”, note: format guaranteed by the C++ standard!

constexpr unsigned day_cpp( const std::string_view s )
{
   return ( ( s[ 4 ] == ' ' ) ? digit( s, 5 ) : two( s, 4 ) ) - 1;
}

constexpr unsigned month_cpp( const std::string_view s )
{
   return ( s == "Jan" ) ? 0
        : ( s == "Feb" ) ? 1
        : ( s == "Mar" ) ? 2

        …

        : ( s == "Oct" ) ? 9
        : ( s == "Nov" ) ? 10
        : ( s == "Dez" ) ? 11
        : throw "invalid month";
}
```

Copyright © 2019 Daniel Frey

```
// ISO date => “2019-07-16"

constexpr bool is_leap_year( const unsigned y )
{
   return ( y % 4 == 0 ) && ( ( y % 400 == 0 ) || ( y % 100 != 0 ) );
}

constexpr unsigned days_of_month( const unsigned y, const unsigned m )
{
   if( m == 1 ) {
      return is_leap_year( y ) ? 29 : 28;
   }
   return ( m == 3 || m == 5 || m == 8 || m == 10 ) ? 30 : 31;
}
```

Copyright © 2019 Daniel Frey

```
// ISO date => “2019-07-16"

constexpr unsigned check( const unsigned i, const unsigned l )
{
   return ( i < l ) ? i : throw "out of range";
}

constexpr unsigned month_iso( const std::string_view s )
{
   return check( two( s, 5 ) - 1, 12 );
}

constexpr unsigned day_iso( const std::string_view s )
{
   return check( two( s, 8 ) - 1, days_of_month( four( s, 0 ), month_iso( s ) ) );
}
```

Copyright © 2019 Daniel Frey

```
constexpr unsigned total_cpp( const std::string_view s )
{
   if( s.size() != 11 || s[ 3 ] != ' ' || s[ 6 ] != ' ' ) {
      throw "invalid string";
   }
   return ( four( s, 7 ) * 12 + month_cpp( s.substr( 0, 3 ) ) ) * 31 + day_cpp( s );
}

constexpr unsigned total_iso( const std::string_view s )
{
   if( s.size() != 10 || s[ 4 ] != '-' || s[ 7 ] != '-' ) {
      throw "invalid string";
   }
   return ( four( s, 0 ) * 12 + month_iso( s ) ) * 31 + day_iso( s );
}

#define TODO( DATE, MSG ) \
   static_assert( total_cpp( __DATE__ ) < total_iso( DATE ), MSG )
```

Copyright © 2019 Daniel Frey

Post-Modern TODOs

```
#include "todo.hpp"

int main()
{
    TODO( "2021-01-01", "World domination!" );
    TODO( "2019-01-01", "Get a cat" );
}
```

Copyright © 2019 Daniel Frey

Post-Modern TODOs

```
#include "todo.hpp"

int main()
{
    TODO( "2021-01-01", "World domination!" );
    TODO( "2019-01-01", "Get a cat" );
}

$ g++ -std=c++17 example.cpp && ./a.out
In file included from example.cpp:1:0:
example.cpp: In function ‘int main()’:
todo.hpp:113:41: error: static assertion failed: Get a cat
  113 |     static_assert( total_cpp( __DATE__ ) < total_iso( DATE ), MSG )
      |                    ~~~~~~~~~~~~~~~~~~~~~^~~~~~~~~~~~~~~~~~
example.cpp:6:4: note: in expansion of macro ‘TODO’
    7 |     TODO( "2019-01-01", "Get a cat" );
      |     ^~~~
$
```

Copyright © 2019 Daniel Frey

Thank You!

https://github.com/d-frey/todo

Copyright © 2019 Daniel Frey

Questions?

https://github.com/d-frey/todo

Copyright © 2019 Daniel Frey