

# Good coding style

*A drama in 12 parts with an epilogue*

Konstantin Osipov  
Staff Engineer  
Sun/MySQL

[kostja@sun.com](mailto:kostja@sun.com)



*These slides released under the Creative Commons  
Attribution-Noncommercial-Share Alike License*



# Agenda

- It's all Monty's fault! Or maybe not?
- Define the problem
- What can be done
- Call to arms
- Thank you & References



# What's a good coding style?

- **C++ FAQ Lite:**

Thank you for reading this answer rather than just trying to set your own coding standards.

But beware that some people on `comp.lang.c++` are very sensitive on this issue. Nearly every software engineer has, at some point, been exploited by someone who used coding standards as a "power play." Furthermore some attempts to set C++ coding standards have been made by those who didn't know what they were talking about, so the standards end up being based on what was the state-of-the-art when the standards setters were writing code. Such impositions generate an attitude of mistrust for coding standards.



# Why is it a problem?

```
kostja@bodhi:~/work/mysql-6.0-runtime/sql$ grep '\<true\>' *.{h,cc}  
| grep -v sql_yacc.cc | wc -l
```

**356**

```
kostja@bodhi:~/work/mysql-6.0-runtime/sql$ grep '\<false\>' *.  
{h,cc} | grep -v sql_yacc.cc | wc -l
```

**212**



# Why is it a problem?

```
struct st_bstream_snapshot_info
```

```
class QUICK_RANGE
```

```
class DsMrr_impl
```

```
class MDL_context and class DDL_blocker_class
```

```
typedef struct st_table_ref TABLE_REF
```

```
class NdbScanOperation and class TableEvent
```

```
class sp_head and sp_instr_jump
```



# MySQL coding style gotchas: maintenance

```
Foo::Foo(int a_arg, int b_arg, int c_arg)
```

```
  : a(a_arg), b(b_arg), c(c_arg)
```

-> rename or add a member

```
int descriptive_identifier_name1= 1;
```

```
char not_so_descriptive= 2;
```

-> now add a declaration of very\_long\_and\_descriptive\_identifier3

```
My_long_class_name::My_long_class_name(int param1, Type1 param 2,  
                                         Type2 param3);
```

-> rename the class or the method



# Why is It a problem?

- sheer neglect to copy constructors and assignment operators
- all members are public, in new code as well
- -> C++ practice guide is missing



## Define a good coding style, attempt #2

- helps readability
- prevents coding mistakes
- simplifies maintenance
- easy to learn
- unambiguous
- comes with a best practices guide



# A change is called for!

## Possible immediate steps:

- prefer **new/delete** to `malloc()/free()` wherever possible
- **TRUE** -> **true**, **FALSE** -> **false** switch
- **m\_** prefixes for private class members
- remove `typedef st_name {} NAME` constructs
- your pet peeve can be here, if it's also everybody's



# The criticism

- what we have is good enough, and everybody is used to it
- there are more important problems (AKA re-engineering is our savior)
- a change means the code is always half-way through it. Doxygen comments adoption.
- changes in the style is a revision history noise



# What can be done?

- let's talk about it, but not too much
- accept and follow the current status quo
- improve what we have, with care



Thank you!



# References

- MySQL coding guidelines :  
[http://forge.mysql.com/wiki/MySQL\\_Internals\\_Coding\\_Guidelines](http://forge.mysql.com/wiki/MySQL_Internals_Coding_Guidelines)
- C++ FAQ lite: <http://www.parashift.com/c++-faq-lite/>
- Bjarne Stroustrup's FAQ: [http://www.research.att.com/~bs/bs\\_faq.html](http://www.research.att.com/~bs/bs_faq.html)
- C++ Style Guide from Geosoft.no: <http://geosoft.no/development/cppstyle.html>
- Collection of other styles: <http://www.chris-lott.org/resources/cstyle/>