

```
1: // $Id: hello.h,v 1.1 2018-01-05 14:33:39-08 - - $
2:
3: #ifndef __HELLO_H__
4: #define __HELLO_H__
5:
6: #include <iostream>
7: #include <string>
8:
9: class hello {
10:     private:
11:         std::string message {"Hello, World!"};
12:     public:
13:         hello();                                // default constructor
14:         hello (const hello&);                    // copy constructor
15:         hello& operator= (const hello&); // copy operator=
16:         hello (hello&);                          // move constructor
17:         hello& operator= (hello&&);              // move operator=
18:         ~hello();                                // destructor
19:         hello (const string&);                    // alternate constructor
20:         void say (ostream&);                      // member function
21: };
22:
23: #endif
24:
```

```
1: // $Id: hello.cpp,v 1.9 2019-04-03 13:52:31-07 - - $
2:
3: #include <iostream>
4: #include <string>
5: using namespace std;
6:
7: #include "hello.h"
8:
9: #define SHOW { \
10:     cout << this << "->" << __PRETTY_FUNCTION__ \
11:     << " {" << message << "}" << endl; \
12: }
13:
14: hello::hello() {
15:     SHOW;
16: }
17:
18: hello::hello (const hello& that): message (that.message) {
19:     SHOW;
20: }
21:
22: hello& hello::operator= (const hello& that) {
23:     SHOW;
24:     if (this != &that) message = that.message;
25:     return *this;
26: }
27:
28: hello::hello (hello&& that): message (std::move (that.message)) {
29:     SHOW;
30: }
31:
32: hello& hello::operator= (hello&& that) {
33:     SHOW;
34:     if (this != &that) message = std::move (that.message);
35:     return *this;
36: }
37:
38: hello::~~hello() {
39:     SHOW;
40: }
41:
42: hello::hello (const string& message_): message(message_) {
43:     SHOW;
44: }
45:
46: void hello::say (ostream& out) {
47:     SHOW;
48:     out << message << endl;
49: }
50:
```

```
1: // $Id: main.cpp,v 1.5 2018-06-19 17:36:22-07 - - $
2:
3: #include <algorithm>
4: #include <cstdlib>
5: #include <iostream>
6: #include <memory>
7: #include <string>
8: #include <vector>
9: using namespace std;
10:
11: #include <libgen.h>
12:
13: #include "hello.h"
14:
15: int main (int argc, char** argv) {
16:     string execname {basename (argv[0])};
17:     vector<string> args (&argv[1], &argv[argc]);
18:     cout << "Execname: " << execname << ", Compiled: "
19:         << __DATE__ << " " << __TIME__ << endl;
20:     cout << "This is: " << __FILE__ << "[" << __LINE__ << "]: "
21:         << __PRETTY_FUNCTION__ << endl;
22:     auto hello_ptr = make_shared<hello>();
23:     hello_ptr->say (cout);
24:     hello goodbye {"Goodbye, world!"};
25:     goodbye.say (cout);
26:     hello second {*hello_ptr};
27:     second.say (cout);
28:     for (const auto& arg: args) cout << "for: " << arg << endl;
29:     for_each (args.cbegin(), args.cend(),
30:         [](auto& arg) { cout << "for_each: " << arg << endl; });
31:     return EXIT_SUCCESS;
32: }
33:
```

```
1: # $Id: Makefile,v 1.19 2019-04-03 14:15:22-07 - - $
2:
3: MKFILE      = Makefile
4: DEFILE      = ${MKFILE}.dep
5: NOINCL      = ci clean spotless
6: NEEDINCL    = ${filter ${NOINCL}, ${MAKECMDGOALS}}
7: GMAKE       = ${MAKE} --no-print-directory
8: GPPWARN     = -Wall -Wextra -Werror -Wpedantic -Wshadow -Wold-style-cast
9: GPPPTS      = ${GPPWARN} -fdiagnostics-color=never
10: COMPILECPP  = g++ -g -O0 -std=gnu++17 ${GPPPTS}
11: MAKEDEPCPP  = g++ -MM -std=gnu++17 ${GPPPTS}
12: LOGCOMMAND   = (echo ""; echo [COMMAND] "${2}"; ${2}) >>${1} 2>&1
13:
14: CPPHEADER    = hello.h
15: CPPSOURCE    = hello.cpp main.cpp
16: ALLCPPSRC    = ${CPPHEADER} ${CPPSOURCE}
17: OBJECTS      = ${CPPSOURCE:.cpp=.o}
18: EXECBIN      = hello
19: OTHERS       = ${MKFILE}
20: ALLSOURCES   = ${ALLCPPSRC} ${OTHERS}
21: LISTING      = Listing.ps
22:
23: all : ${EXECBIN}
24:
25: ${EXECBIN} : ${OBJECTS}
26:     ${COMPILECPP} -o $@ ${OBJECTS}
27:
28: %.o : %.cpp
29:     - checksource $<
30:     - cpplint.py.perl $<
31:     ${COMPILECPP} -c $<
32:
33: ci : ${ALLSOURCES}
34:     - checksource ${ALLSOURCES}
35:     - cpplint.py.perl ${ALLCPPSRC}
36:     cid + ${ALLSOURCES}
37:
38: lis : ${ALLSOURCES} test file
39:     mkpspdf ${LISTING} ${ALLSOURCES} ${DEFILE} test.out files.log
40:
41: clean :
42:     - rm ${OBJECTS} ${DEFILE} core test.out files.log
43:
44: spotless : clean
45:     - rm ${EXECBIN} ${LISTING} ${LISTING:.ps=.pdf}
46:
47: test : ${EXECBIN}
48:     ./${EXECBIN} foo bar baz qux >test.out 2>&1
49:
50: file : ${EXECBIN}
51:     LC_TIME=C date >files.log
52:     ${call LOGCOMMAND,files.log,checksource ${ALLSOURCES}}
53:     ${call LOGCOMMAND,files.log,cpplint.py.perl ${ALLCPPSRC}}
54:     ${call LOGCOMMAND,files.log,file *}
55:
```

```
56:
57: dep : ${ALLCPPSRC}
58:     @ echo "# ${DEPFILE} created `LC_TIME=C date`" >${DEPFILE}
59:     ${MAKEDEPCPP} ${CPPSOURCE} >>${DEPFILE}
60:
61: ${DEPFILE} : ${MAKEFILE}
62:     @ touch ${DEPFILE}
63:     ${GMAKE} dep
64:
65: again :
66:     ${GMAKE} spotless dep ci all lis
67:
68: ifeq (${NEEDINCL}, )
69: include ${DEPFILE}
70: endif
```

```
1: # Makefile.dep created Wed Aug 21 14:11:30 PDT 2019
2: hello.o: hello.cpp hello.h
3: main.o: main.cpp hello.h
```

```
1: Execname: hello, Compiled: Aug 21 2019 14:11:31
2: This is: main.cpp[20]: int main(int, char**)
3: 0x11c7140->hello::hello() {"Hello, World!"}
4: 0x11c7140->void hello::say(std::ostream&) {"Hello, World!"}
5: Hello, World!
6: 0x7ffcb422c538->hello::hello(const string&) {"Goodbye, world!"}
7: 0x7ffcb422c538->void hello::say(std::ostream&) {"Goodbye, world!"}
8: Goodbye, world!
9: 0x7ffcb422c530->hello::hello(const hello&) {"Hello, World!"}
10: 0x7ffcb422c530->void hello::say(std::ostream&) {"Hello, World!"}
11: Hello, World!
12: for: foo
13: for: bar
14: for: baz
15: for: qux
16: for_each: foo
17: for_each: bar
18: for_each: baz
19: for_each: qux
20: 0x7ffcb422c530->hello::~~hello() {"Hello, World!"}
21: 0x7ffcb422c538->hello::~~hello() {"Goodbye, world!"}
22: 0x11c7140->hello::~~hello() {"Hello, World!"}
```

```
1: Wed Aug 21 14:11:32 PDT 2019
2:
3: [COMMAND] checksource hello.h hello.cpp main.cpp Makefile
4:
5: [COMMAND] cpplint.py.perl hello.h hello.cpp main.cpp
6: Done processing hello.h
7: Done processing hello.cpp
8: Done processing main.cpp
9:
10: [COMMAND] file *
11: HEADER.html:  HTML document, ASCII text
12: Makefile:      ASCII text
13: Makefile.dep:  ASCII text
14: RCS:           directory
15: files.log:     ASCII text
16: hello:         ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux),
dynamically linked (uses shared libs), for GNU/Linux 2.6.32, BuildID[sha1]=d58f
c5ccb99729bc2e6c21de0a274f56c828abb2, not stripped
17: hello.cpp:    C source, ASCII text
18: hello.h:      C++ source, ASCII text
19: hello.o:      ELF 64-bit LSB relocatable, x86-64, version 1 (GNU/Linux),
not stripped
20: main.cpp:     C source, ASCII text
21: main.o:      ELF 64-bit LSB relocatable, x86-64, version 1 (GNU/Linux),
not stripped
22: test.out:    C source, ASCII text
```