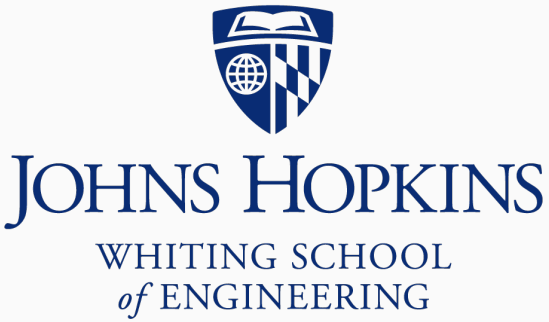
override

Ben Langmead

ben.langmead@gmail.com

www.langmead-lab.org



Source markdown available at github.com/BenLangmead/c-cpp-notes

override

```
#include <iostream>
#include <string>

using std::cout; using std::endl;

class Account {
  public:
    virtual std::string type() const { return "Account"; }
};
class CheckingAccount : public Account {
  public:
    virtual std::string type()       { return "CheckingAccount"; }
};

int main() {
    CheckingAccount checking;
    Account& acct = checking;
    cout << acct.type() << endl; // polymorphism FTW?
    return 0;
}
```

override

```
$ g++ -c override.cpp -std=c++11 -pedantic -Wall -Wextra
$ g++ -o override override.o
$ ./override
Account
```

Sometimes you *intend* to override a function in the base class. . .

. . . but you fail

override

In this case, it was just a matter of missing a const

```
class Account {
  public:
    virtual std::string type() const { return "Account"; }
};
class CheckingAccount : public Account {
  public:
    virtual std::string type()       { return "CheckingAccount"; }
    //                          ^^^^^ oops
};
```

override

```
$ g++ -c override_fix.cpp -std=c++11 -pedantic -Wall -Wextra
$ g++ -o override_fix override_fix.o
$ ./override_fix
CheckingAccount
```

That's better

This is a typical mistake, often because we:

- fail to match const status
- fail to exactly match parameter & return types

override

The `override` keyword helps

When you *intend* to override a function, add the `override` modifier:

```
class Account {
  public:
    virtual std::string type() const { return "Account"; }
};
class CheckingAccount : public Account {
  public:
    virtual std::string type()       override { return "CheckingAccount"; }
    //                       ^^^^^ oops
};
```

override

```
$ g++ -c override_fix2.cpp -std=c++11 -pedantic -Wall -Wextra
override_fix2.cpp:12:25: error: 'virtual std::__cxx11::string
CheckingAccount::type()' marked 'override', but does not override
    virtual std::string type() override { return "CheckingAccount"; }
                        ^~~~
```

override

Now we combine it with const to fix the problem:

```
class Account {
  public:
    virtual std::string type() const { return "Account"; }
};
class CheckingAccount : public Account {
  public:
    virtual std::string type() const override { return "CheckingAccount"; }
};
```

```
$ g++ -c override_fix2.cpp -std=c++11 -pedantic -Wall -Wextra
$ g++ -o override_fix2 override_fix2.o
$ ./override_fix2
CheckingAccount
```

No warnings or errors because override was successful