Ch10. Overriding 확인 방법

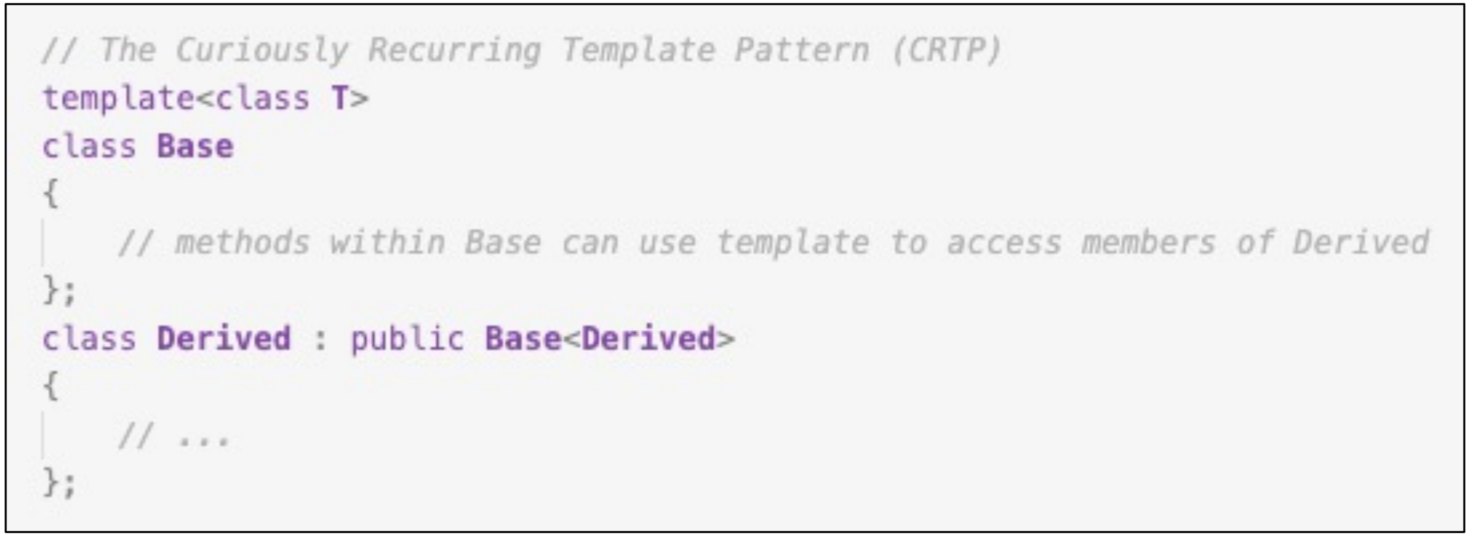
Jong-Hyeok Park
akindo19@gmail.com

SUNGKYUNKWAN UNIVERSITY
1398
성균관대학교

VLDB
VLDB Lab. @ SKKU
http://vldb.skku.ac.kr

CRTP

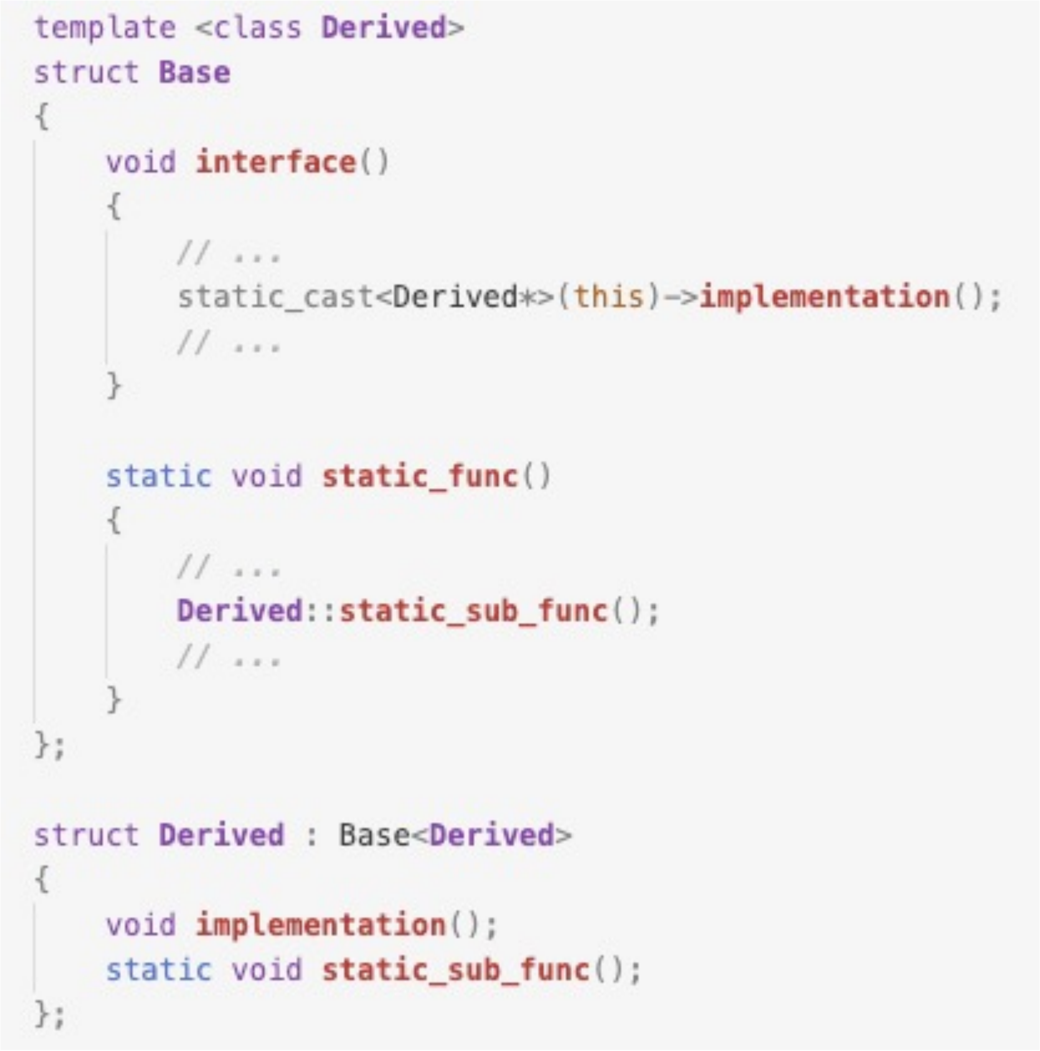
- **C**uriously **R**ecurring **T**emplate **P**attern
  - 클래스 x가 x 자신을 템플릿 인자로 사용하여 클래스 템플릿 인스턴스화로부터 파생되게 하는 것
  - 기반 클래스에서 파생 클래스의 이름을 사용할 수 있게 하는 기법



```
// The Curiously Recurring Template Pattern (CRTP)
template<class T>
class Base
{
    // methods within Base can use template to access members of Derived
};
class Derived : public Base<Derived>
{
    // ...
};
```

CRPT

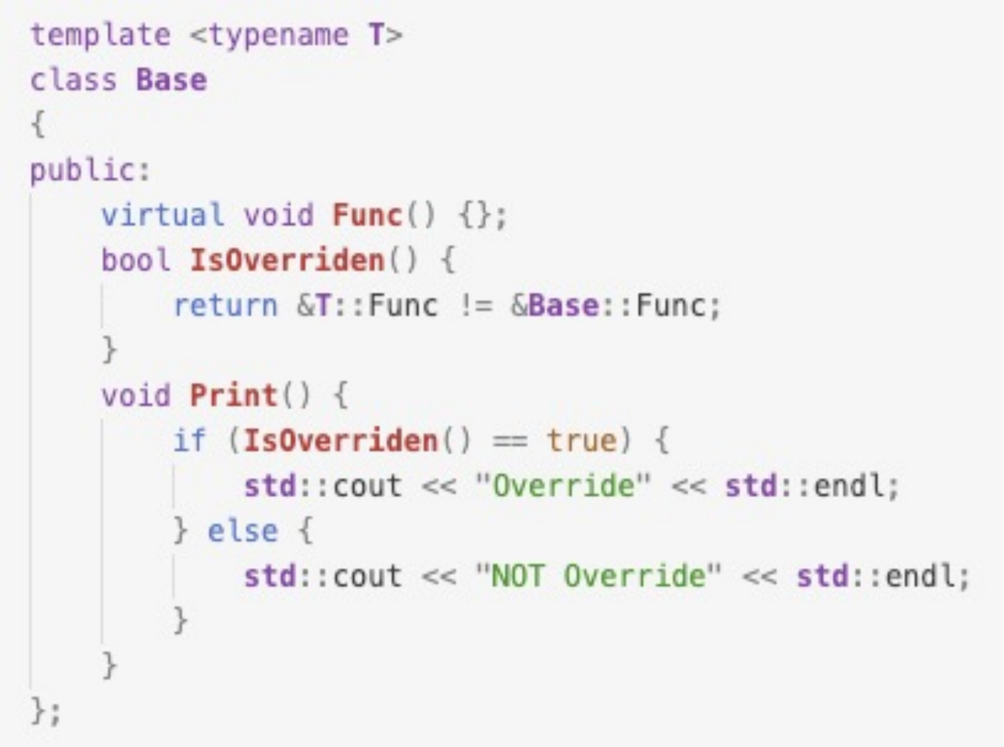
- Static Polymorphism
  - Virtual Function 효과
  - 동적 다형성 비용 없음
  - 컴파일때 결정



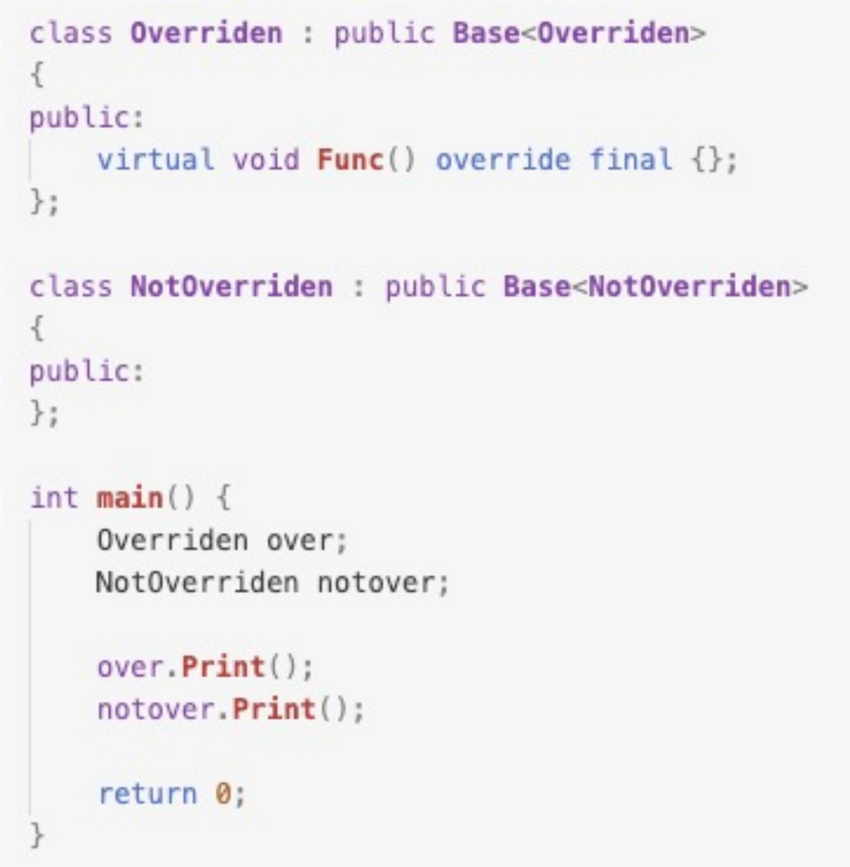
```
template <class Derived>
struct Base
{
    void interface()
    {
        // ...
        static_cast<Derived*>(this)->implementation();
        // ...
    }

    static void static_func()
    {
        // ...
        Derived::static_sub_func();
        // ...
    }
};

struct Derived : Base<Derived>
{
    void implementation();
    static void static_sub_func();
};
```

Overriding 확인용 Helper Class

- Override Helper Class



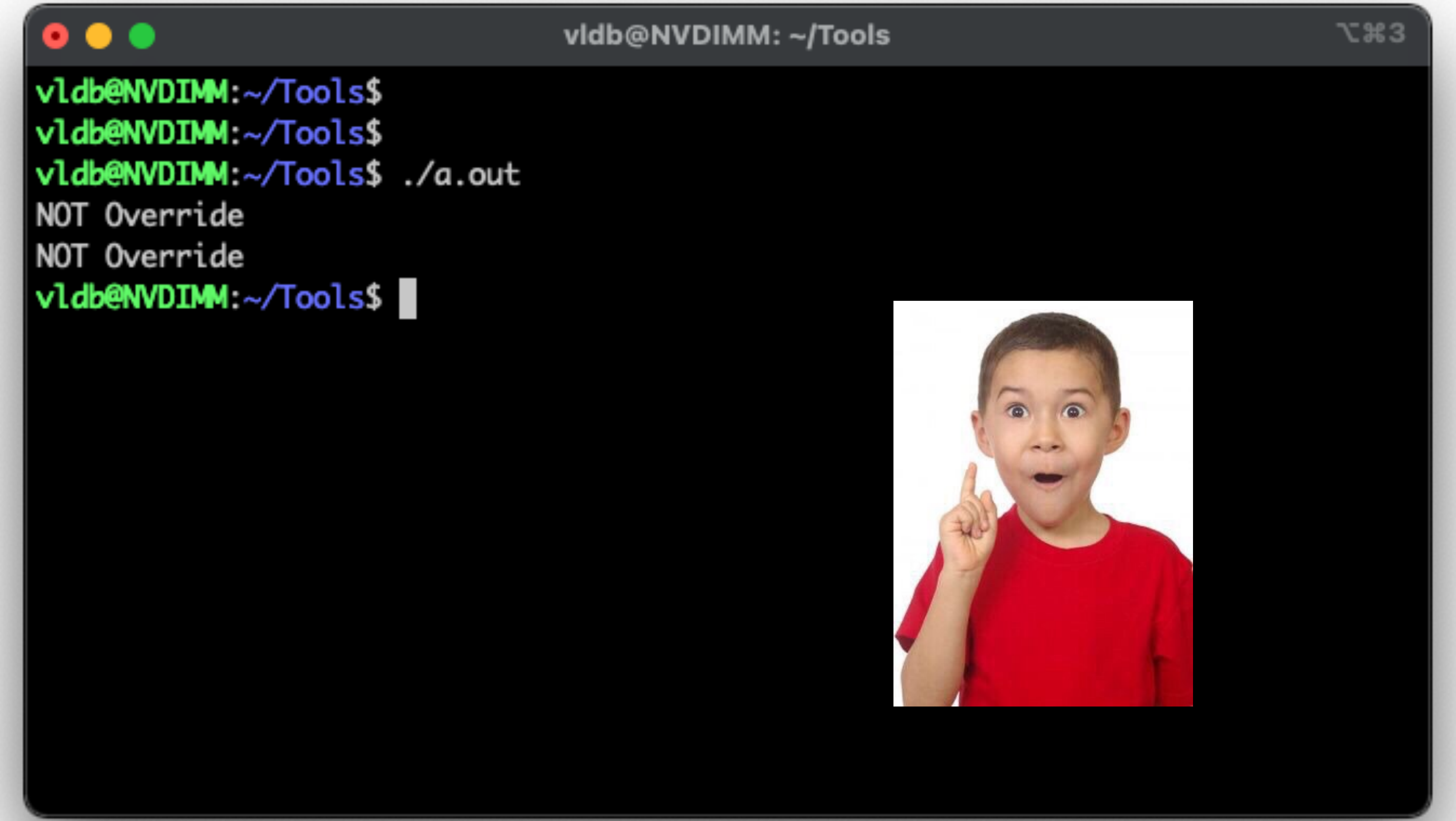
```cpp
template <typename T>
class Base
{
public:
    virtual void Func() {};
    bool IsOverriden() {
        return &T::Func != &Base::Func;
    }
    void Print() {
        if (IsOverriden() == true) {
            std::cout << "Override" << std::endl;
        } else {
            std::cout << "NOT Override" << std::endl;
        }
    }
};
```



```cpp
class Overriden : public Base<Overriden>
{
public:
    virtual void Func() override final {};
};

class NotOverriden : public Base<NotOverriden>
{
public:
};

int main() {
    Overriden over;
    NotOverriden notover;

    over.Print();
    notover.Print();

    return 0;
}
```

Overriding 확인용 Helper Class

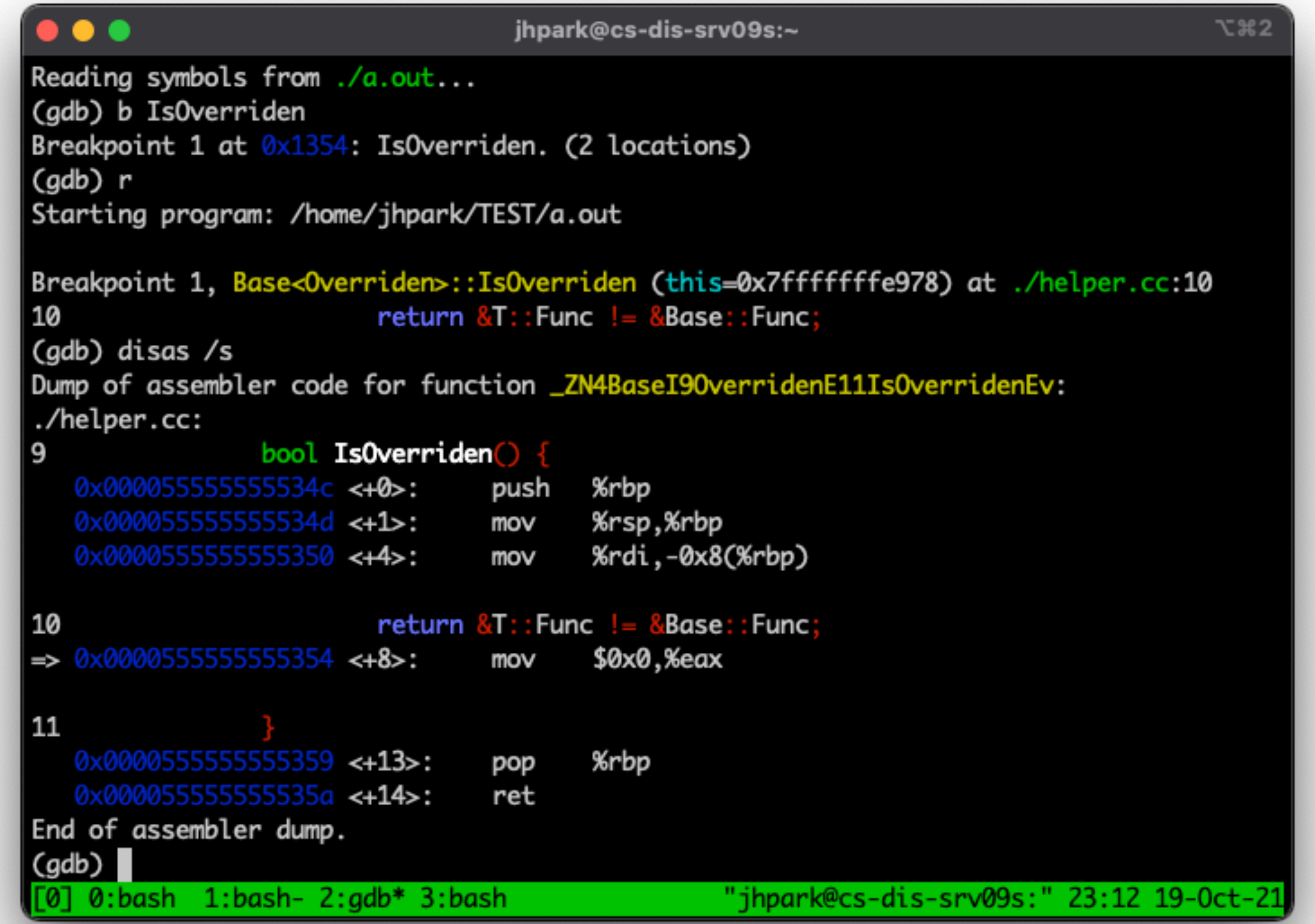
- Override Helper Class



```
vldb@NVDIMM:~/Tools$
vldb@NVDIMM:~/Tools$
vldb@NVDIMM:~/Tools$ ./a.out
NOT Override
NOT Override
vldb@NVDIMM:~/Tools$ 
```

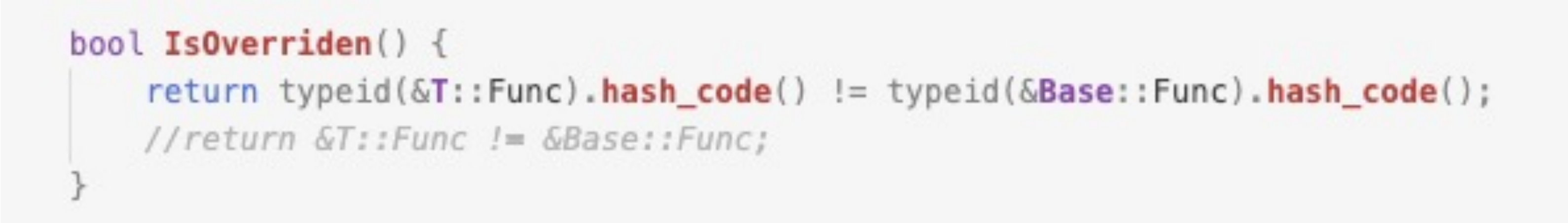
Overriding 확인용 Helper Class

- Override Helper Class



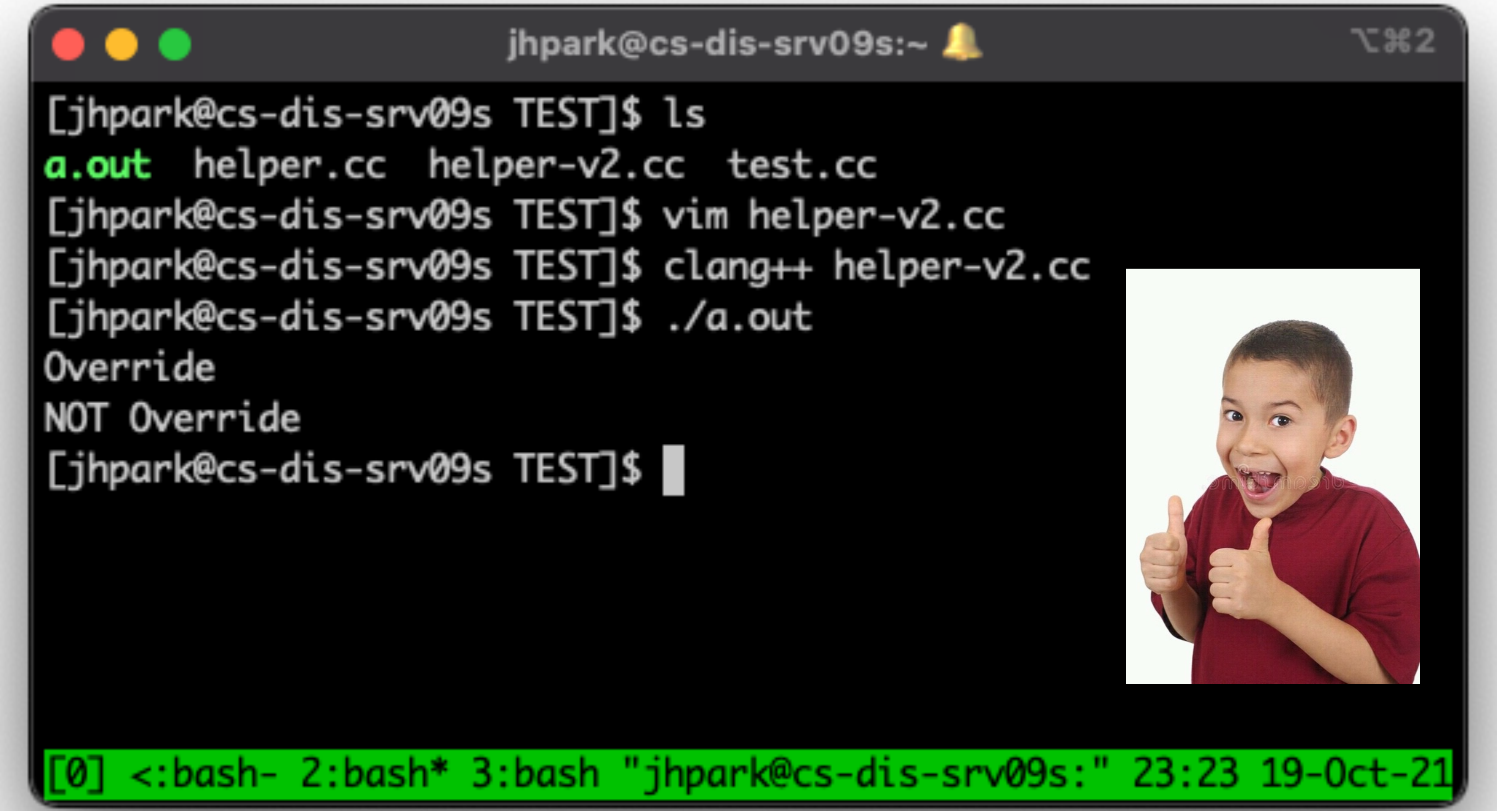
```
Reading symbols from ./a.out...
(gdb) b IsOverriden
Breakpoint 1 at 0x1354: IsOverriden. (2 locations)
(gdb) r
Starting program: /home/jhpark/TEST/a.out

Breakpoint 1, Base<Overriden>::IsOverriden (this=0x7fffffffe978) at ./helper.cc:10
10                      return &T::Func != &Base::Func;
(gdb) disas /s
Dump of assembler code for function _ZN4BaseI9OverridenE11IsOverridenEv:
./helper.cc:
9               bool IsOverriden() {
   0x000055555555534c <+0>:     push   %rbp
   0x000055555555534d <+1>:     mov    %rsp,%rbp
   0x0000555555555350 <+4>:     mov    %rdi,-0x8(%rbp)

10                      return &T::Func != &Base::Func;
=> 0x0000555555555354 <+8>:     mov    $0x0,%eax

11              }
   0x0000555555555359 <+13>:    pop    %rbp
   0x000055555555535a <+14>:    ret
End of assembler dump.
(gdb) 
[0] 0:bash  1:bash- 2:gdb* 3:bash               "jhpark@cs-dis-srv09s:" 23:12 19-Oct-21
```

Overriding 확인용 Helper Class



```
bool IsOverriden() {
    return typeid(&T::Func).hash_code() != typeid(&Base::Func).hash_code();
    //return &T::Func != &Base::Func;
}
```

Overriding 확인용 Helper Class



```
[jhpark@cs-dis-srv09s TEST]$ ls
a.out  helper.cc  helper-v2.cc  test.cc
[jhpark@cs-dis-srv09s TEST]$ vim helper-v2.cc
[jhpark@cs-dis-srv09s TEST]$ clang++ helper-v2.cc
[jhpark@cs-dis-srv09s TEST]$ ./a.out
Override
NOT Override
[jhpark@cs-dis-srv09s TEST]$ 



[0] <:bash- 2:bash* 3:bash "jhpark@cs-dis-srv09s:" 23:23 19-Oct-21
```

References

[1] Marc Gregoire, 2018, Professional C++, 4th edition, WILEY

[2] http://en.wikipedia.org/wiki/Curiously_recurring_template_pattern

[3] https://m.post.naver.com/viewer/postView.nhn?volumeNo=22567976&memberNo=559061