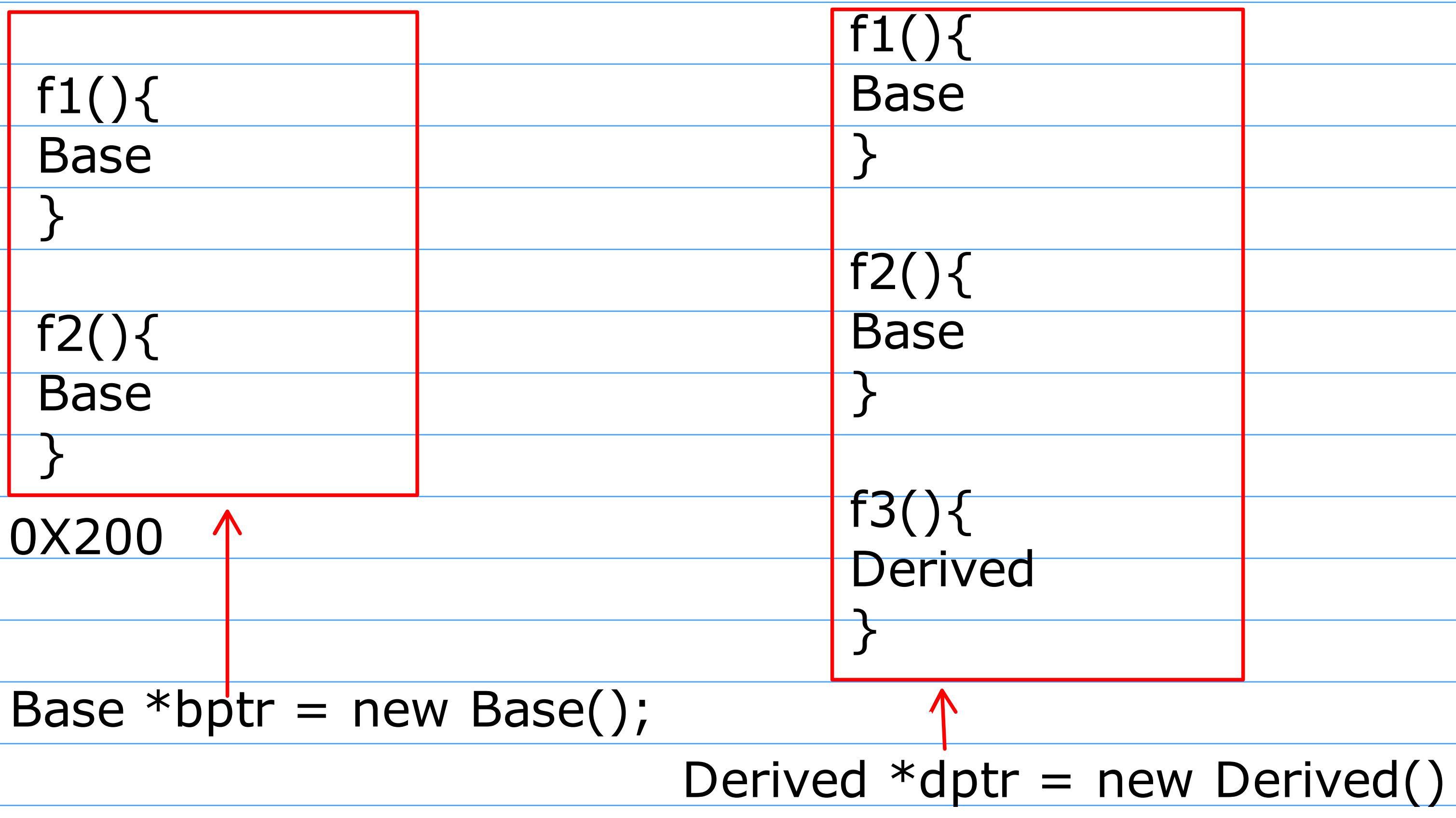
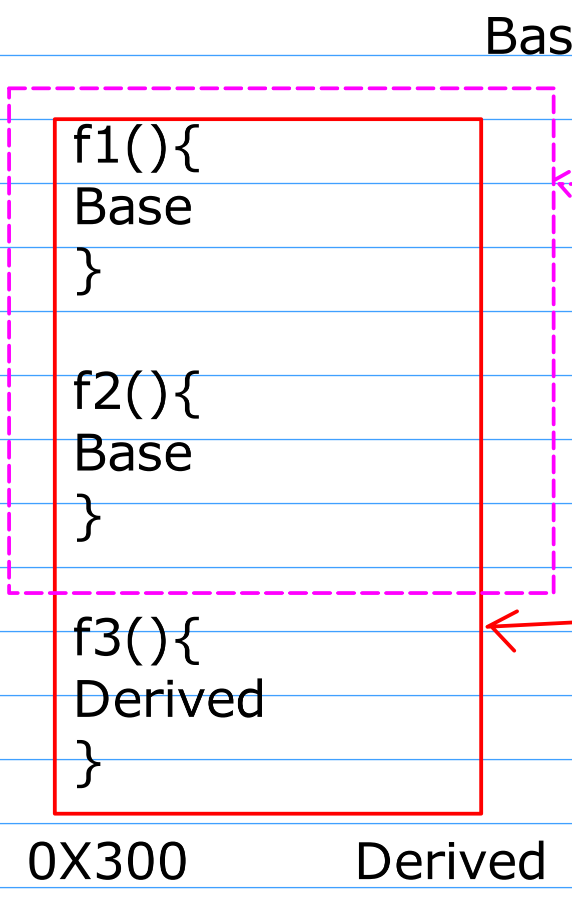
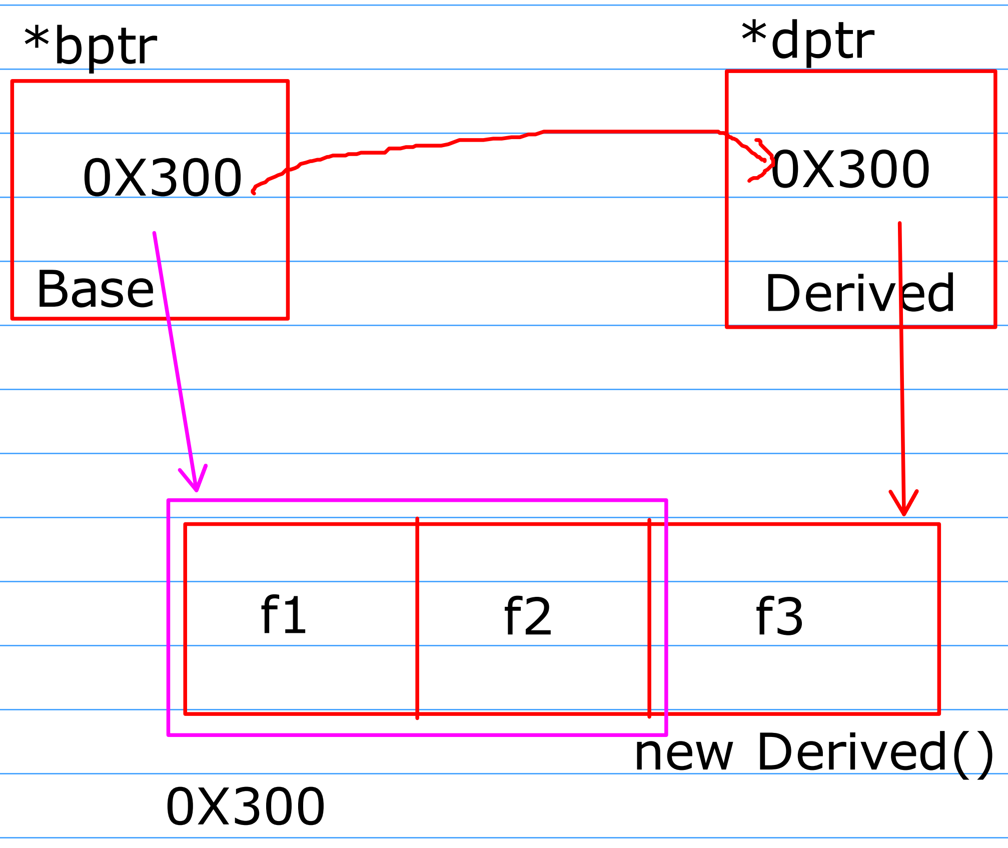
```
class Base{
public:
void f1(){
}
}

class Derived : public Base{

}
```

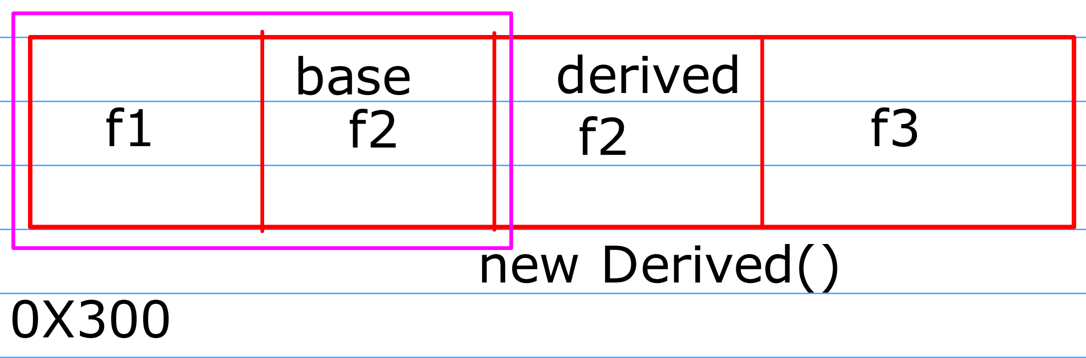


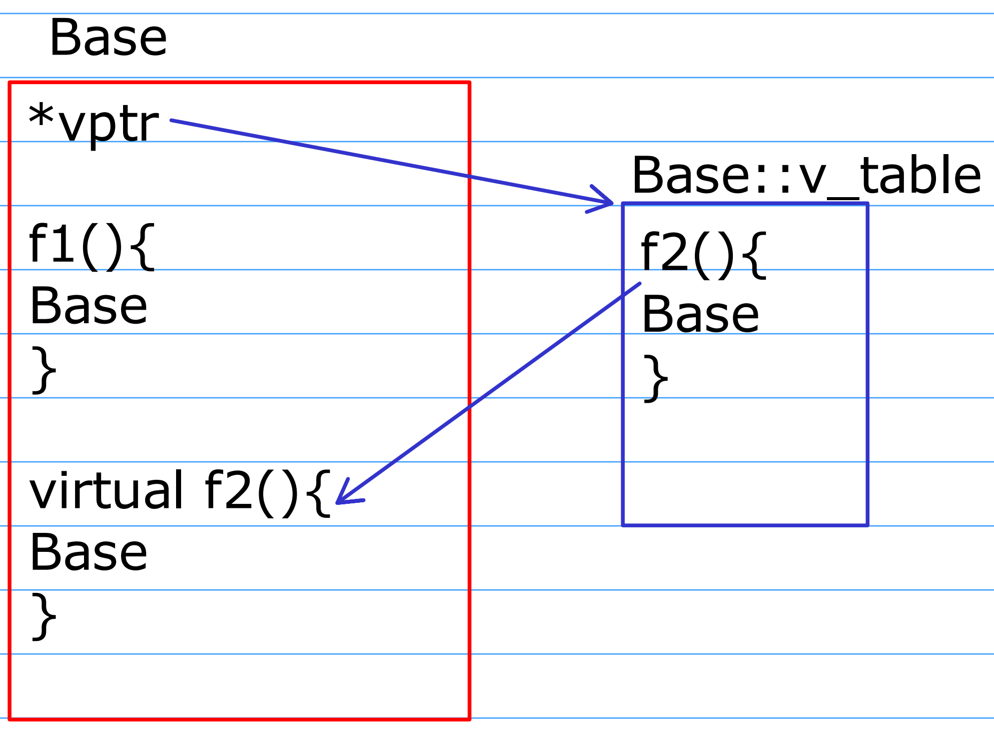


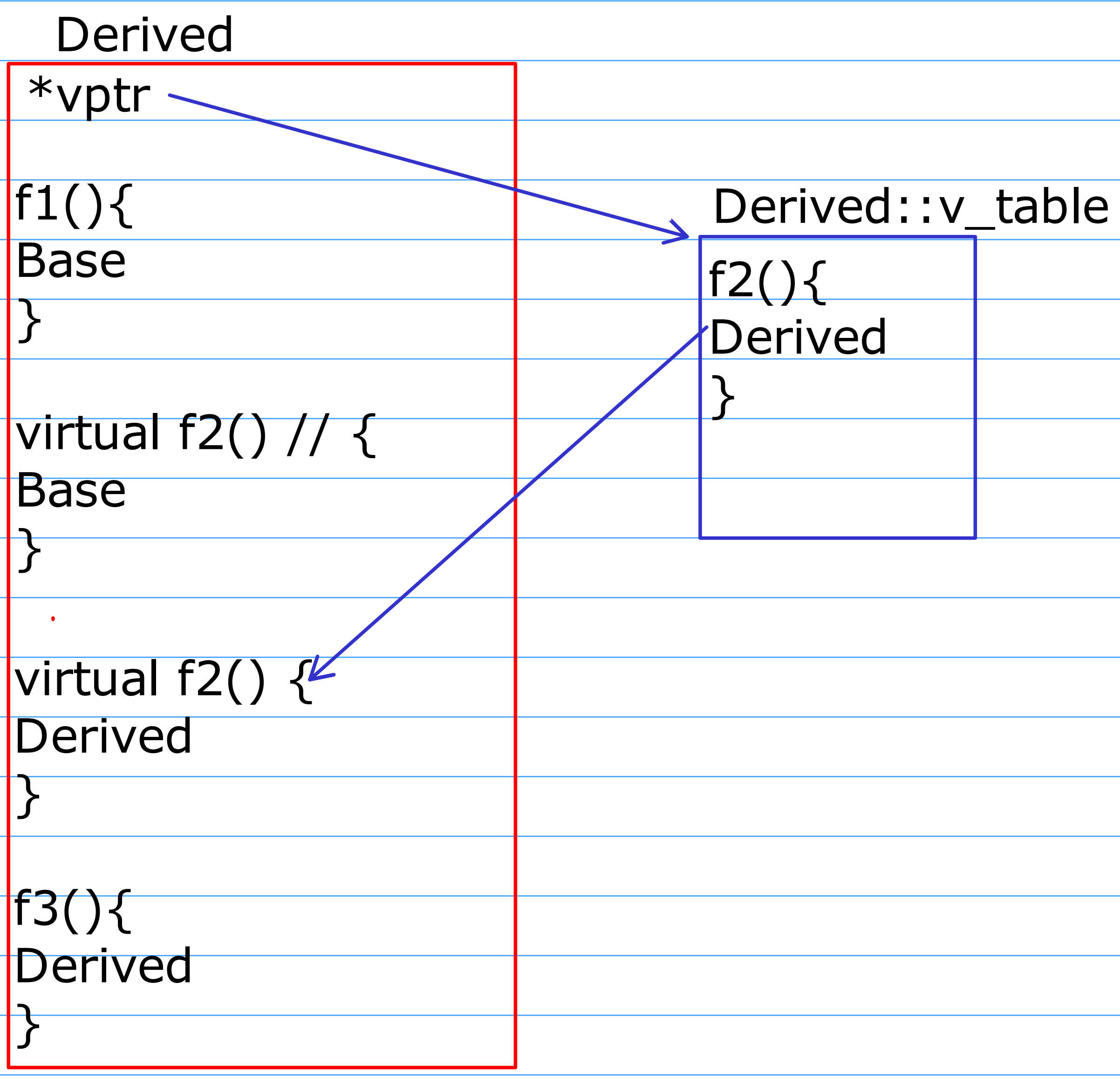
```
Base *bptr = new Derived(); // 0X300
// Upcasting
bptr->f1();// OK
bptr->f2();// OK
bptr->f3();// NOT OK -> Object Slicing

Derived *dptr =(Derived *) bptr; // 0X300
// Downcasting

dptr->f3();
```



Why to override the function?
- If the implementation of the base class function in 100 % incomplete
- If the implementation of the base class function is partail complete
- If you want the implementation totally different from the base class implementation







```
Base b;
b.f1(); b.f2();

Base *bptr = new Base();
bptr->f1();
bptr->f2(); // Base::f2()

Base *bptr = new Derived(); // upcasting
bptr->f1();
bptr->f2(); // Derived::f2()
```

```
Derived d;
d.f1();d.f2();d.f3();

Derived *dptr = new Derived();
dptr->f1();
dptr->f2();
dptr->f3();
```

Advanced Casting Operator
1. dynamic_cast
2. static_cast
3. reinterpret_cast
4. const_cast

Person * p[10];

Add Employee

Add Student

Display All Person

Display Only Employees (typeid())

Display Only Student (typeid())

findEmployee()

findStudent()

for()