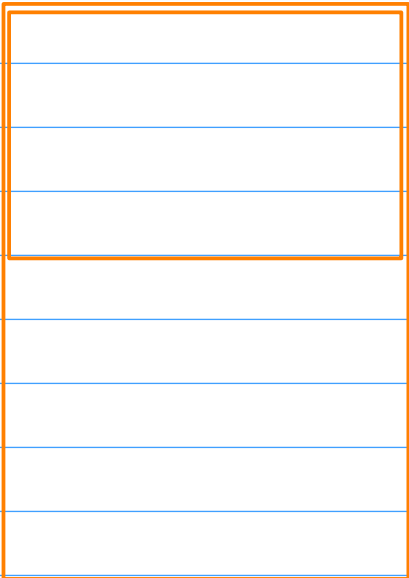


## Upcasting and Downcasting

```
Base *ptr = new Base();  
Derived *dptr = bptr;
```

```
*ptr = NULL  
ptr->accept(); // Program crash(CPP) // NullPointerException(Java)
```

```
Base *ptr = new Base(); // NO Upcasting  
Derived *dptr = bptr; // Downcasting // ClassCastException (Java)  
dptr->f3(); // Program crash(CPP)
```



Base b

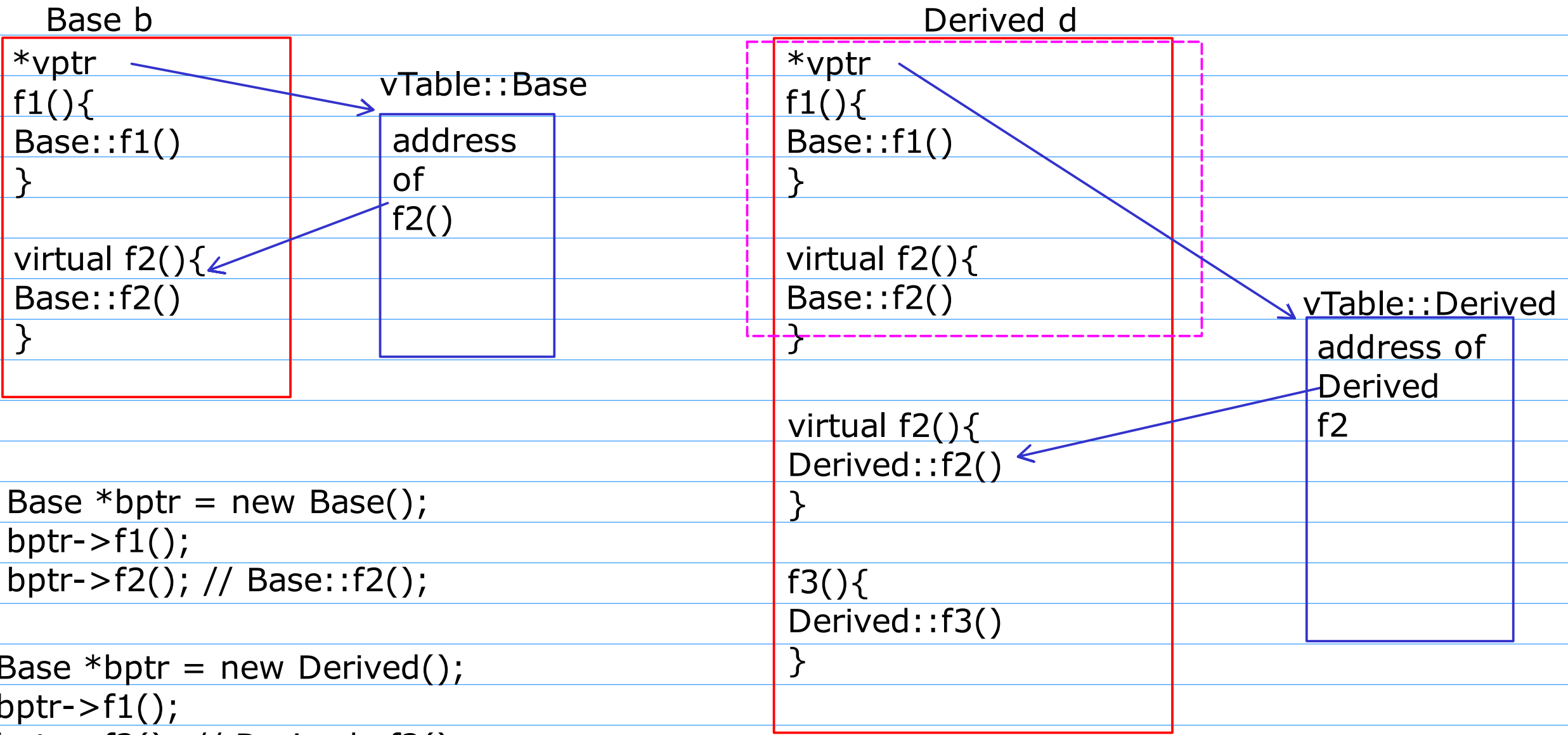
```
f1(){  
Base::f1()  
}  
  
f2(){  
Base::f2()  
}
```

Derived d

```
f1(){  
Base::f1()  
}  
  
f2(){  
Base::f2()  
}  
  
f2(){  
Derived::f2()  
}  
  
f3(){  
Derived::f3()  
}
```

## Function Overriding

- Redefining the function of the base class once again inside the derived class is called as function overriding.
- Why to override the function ?
  1. If the function inside the base class is 100 % incomplete
  2. If the function inside the base class is partial complete
  3. If the implemenattion of the function in derived class we want toatlly differnt than the base class funtion (Java ->toString(), equals())



Hybrid Inheritance  
Employee

Manager      Salesman

Salesmanager

Product      Product\*[3]

Book      Tape

const\_cast

- 1. Notes
- 2. Assignemnt Q1
- 3. Hybrid Inheritance