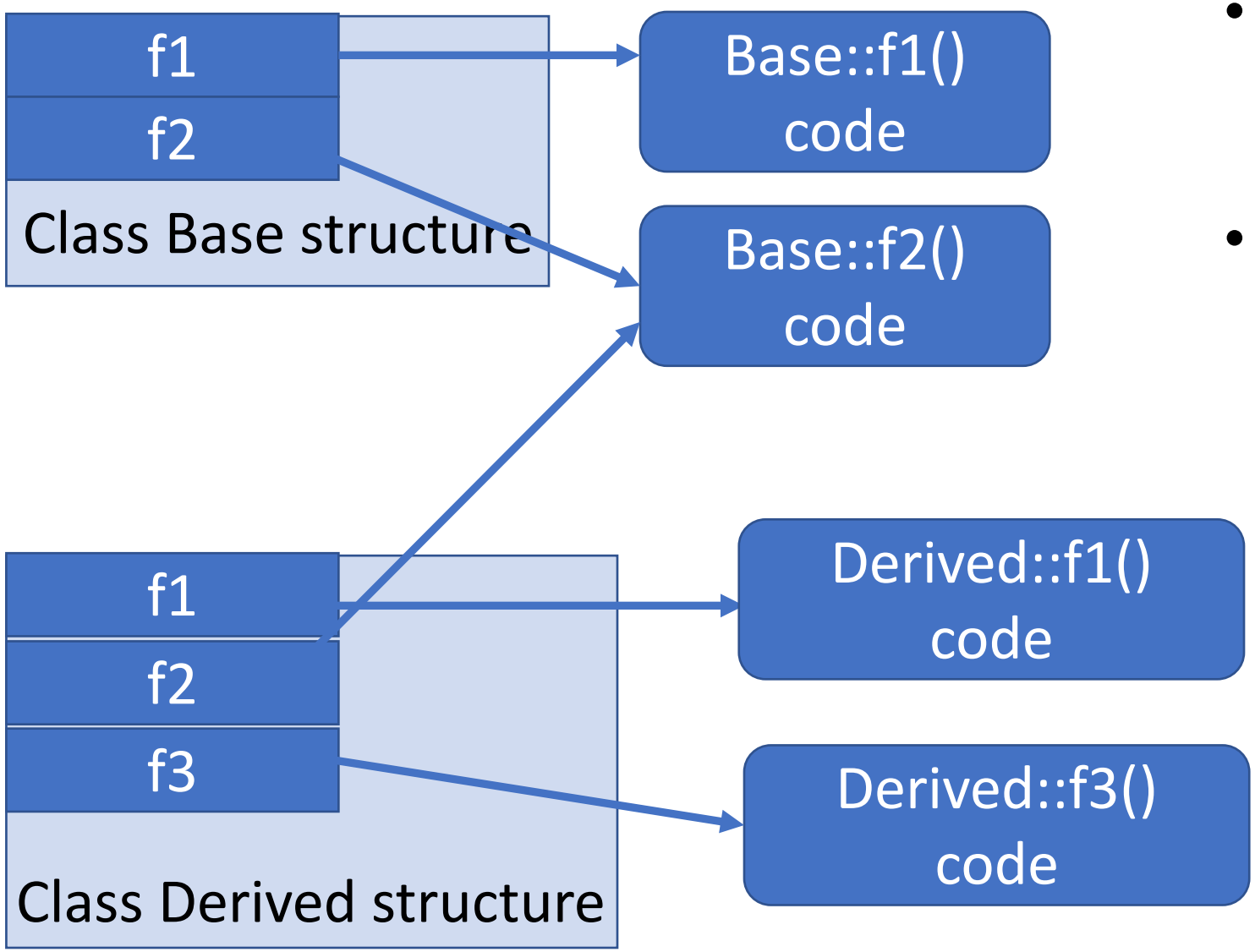
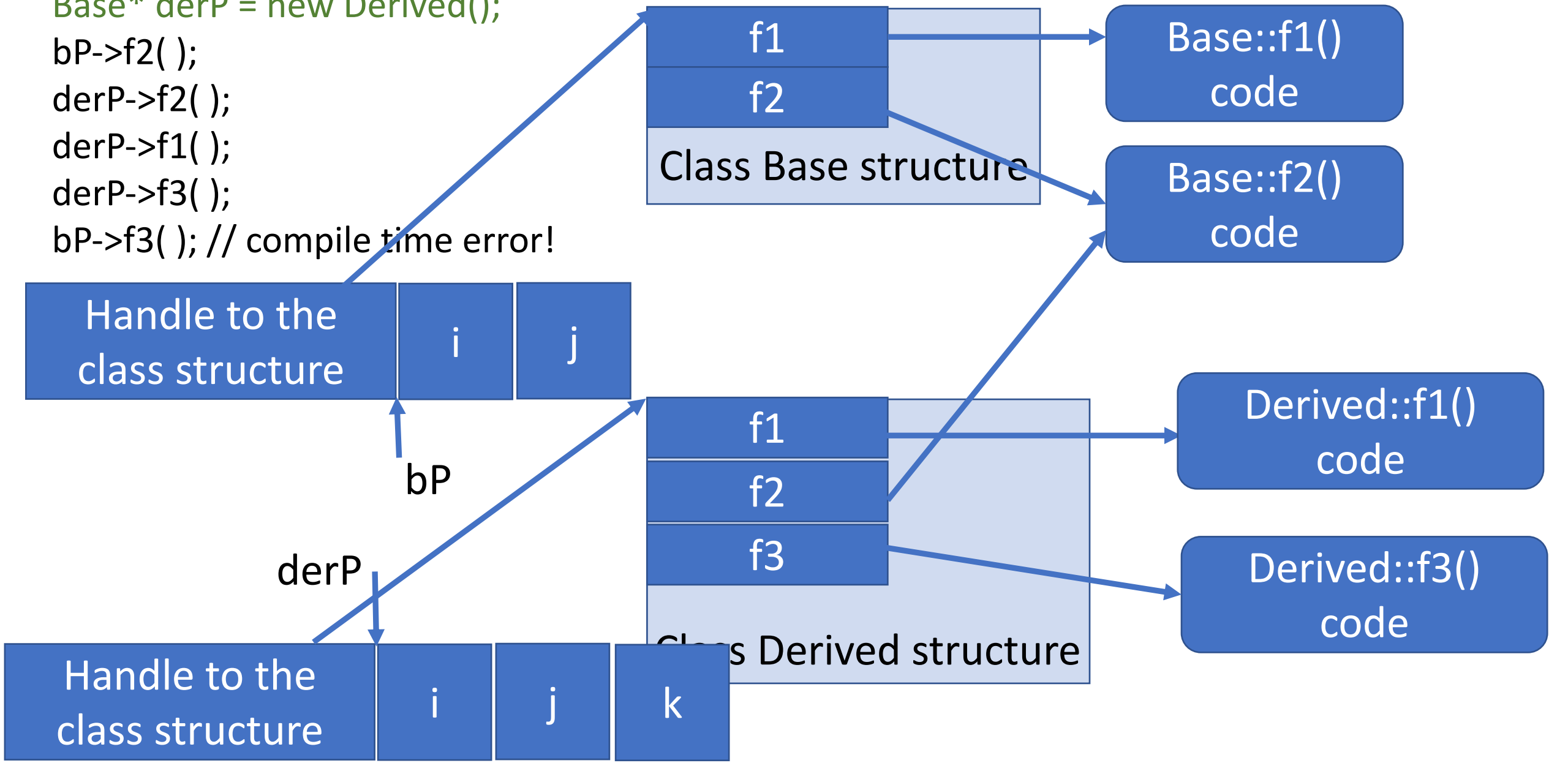
How polymorphism works

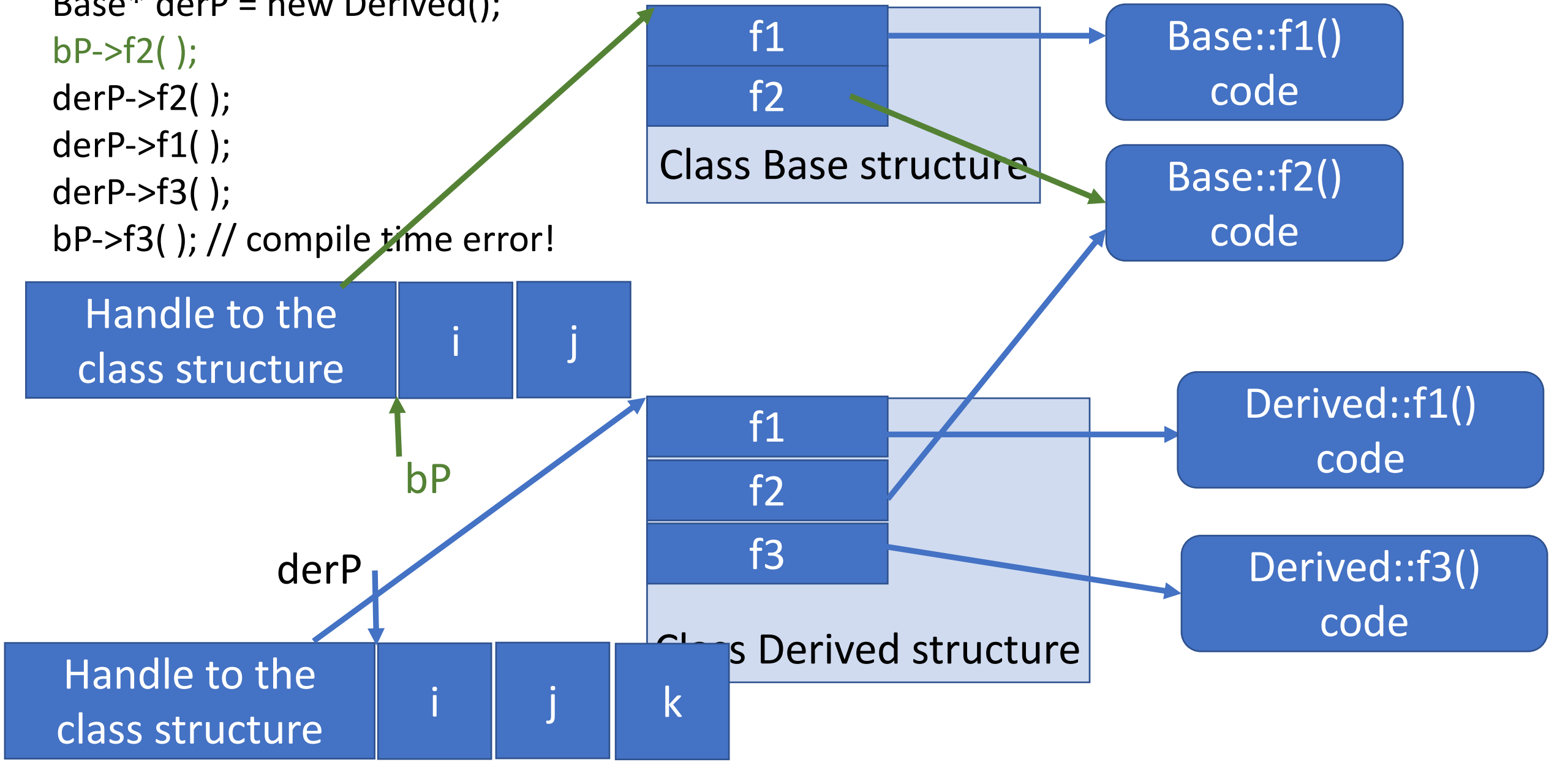


- Class Base defines fields i and j and virtual functions f1() and f2().
- Class Derived defines field k and virtual functions f1() and f3()

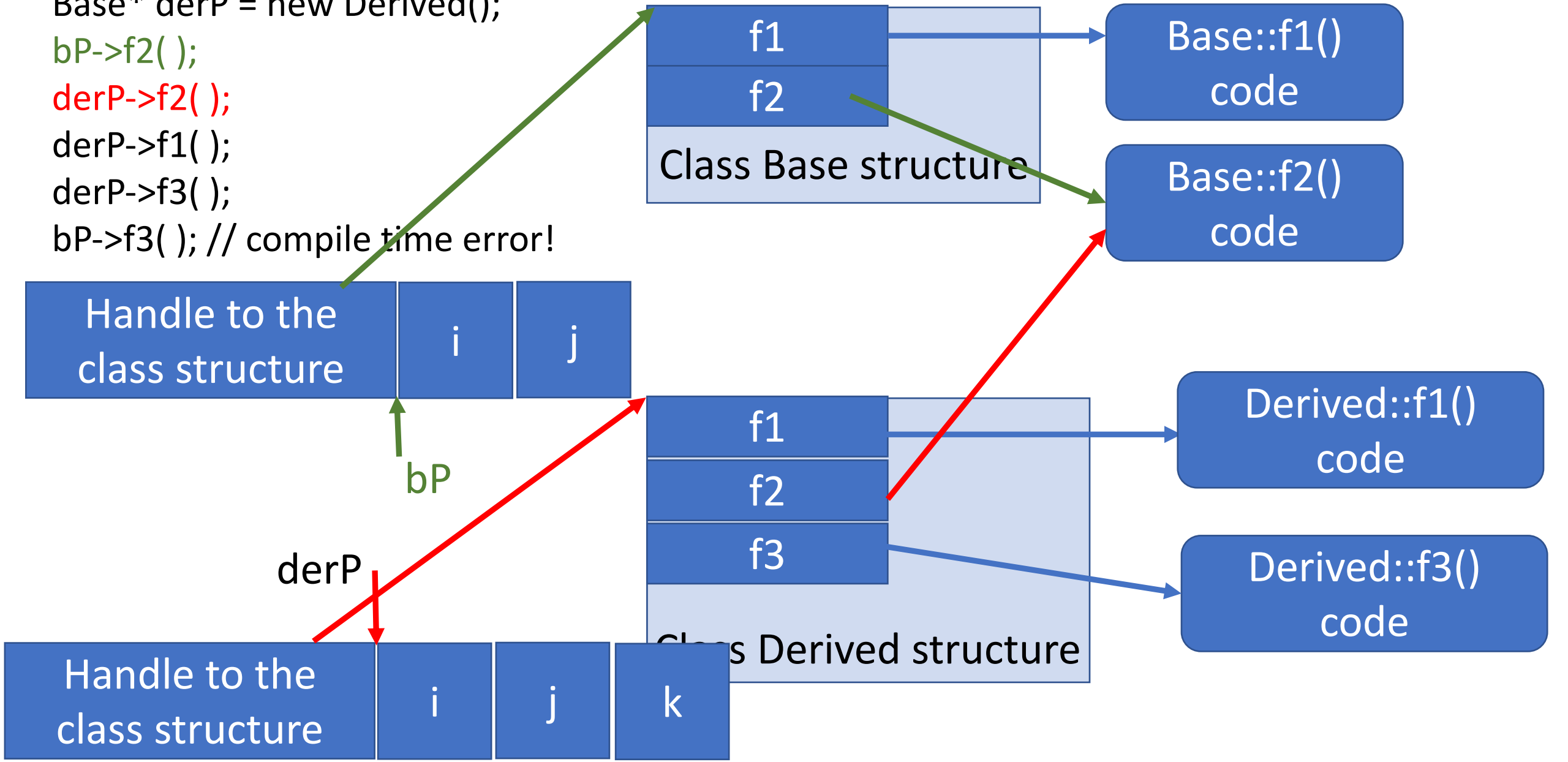
```
Base* bP = new Base( );
Base* derP = new Derived();
bP->f2( );
derP->f2( );
derP->f1( );
derP->f3( );
bP->f3( ); // compile time error!
```



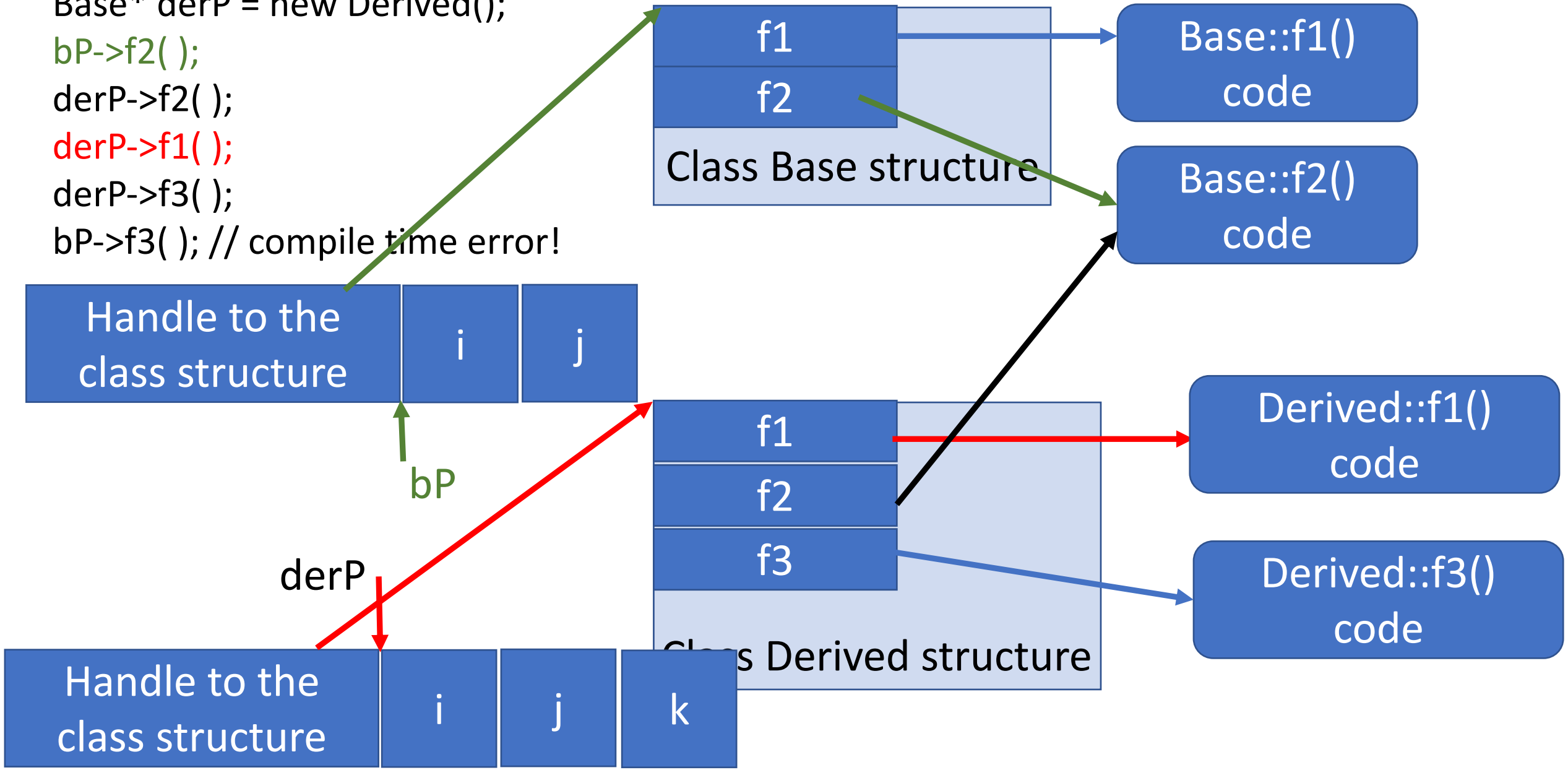
```
Base* bP = new Base( );
Base* derP = new Derived();
bP->f2( );
derP->f2( );
derP->f1( );
derP->f3( );
bP->f3( ); // compile time error!
```



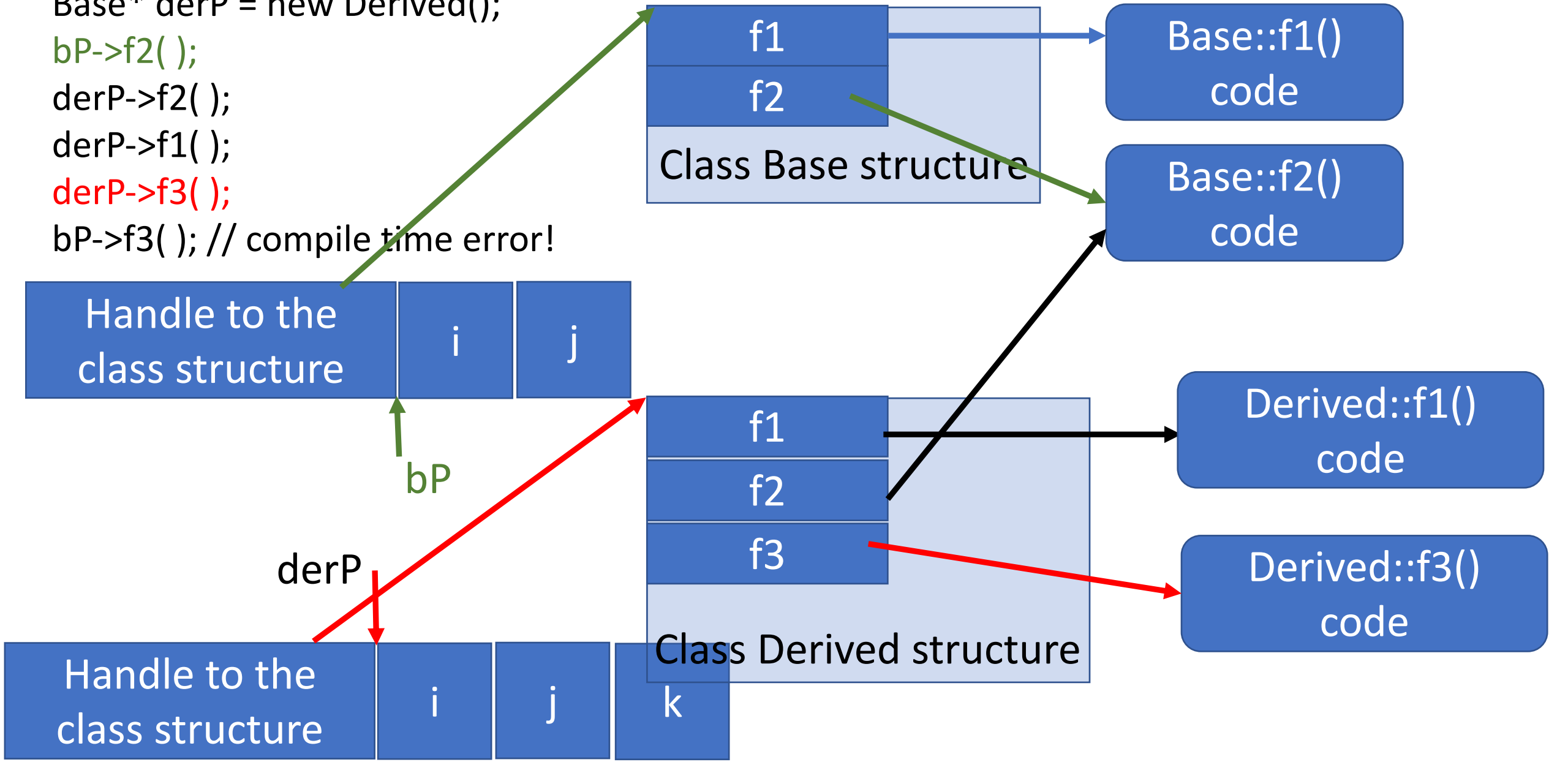
```
Base* bP = new Base( );
Base* derP = new Derived();
bP->f2( );
derP->f2( );
derP->f1( );
derP->f3( );
bP->f3( ); // compile time error!
```



```
Base* bP = new Base( );
Base* derP = new Derived();
bP->f2( );
derP->f2( );
derP->f1( );
derP->f3( );
bP->f3( ); // compile time error!
```



```
Base* bP = new Base( );
Base* derP = new Derived();
bP->f2( );
derP->f2( );
derP->f1( );
derP->f3( );
bP->f3( ); // compile time error!
```



```
Base* bP = new Base( );
Base* derP = new Derived();
bP->f2( );
derP->f2( );
derP->f1( );
derP->f3( );
bP->f3( ); // compile time error!
```