

```
public class SampleClass extends BaseClass1 {
    private int i;
    private String str;
}
```

```
public class BaseClass1 {
    protected char c;
}
```

Reference:

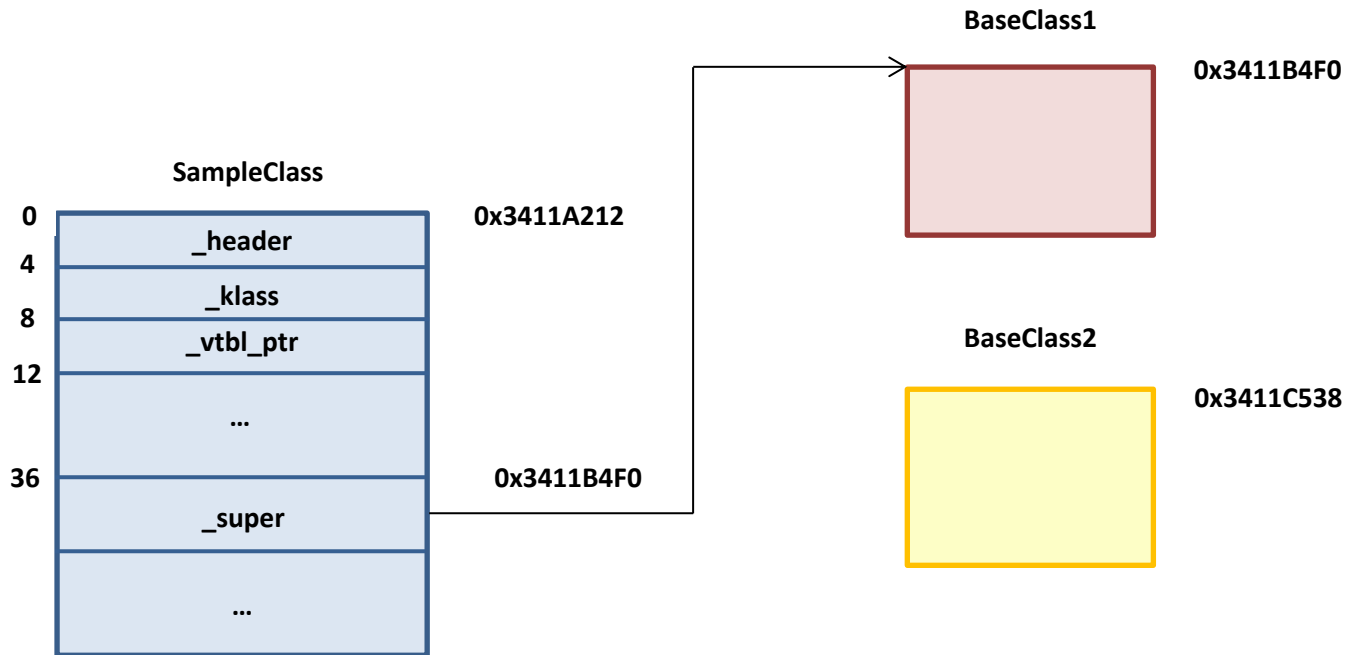
<http://hg.openjdk.java.net/jdk7/hotspot/hotspot/file/9b0ca45cd756/src/share/vm/oops/klass.hpp>

```
// Klass layout:
// [header          ] klassOop
// [klass pointer   ] klassOop
// [C++ vtbl ptr    ] (contained in Klass_vtbl)
// [layout_helper   ]
// [super_check_offset] for fast subtype checks
// [secondary_super_cache] for fast subtype checks
// [secondary_supers    ] array of 2ndary supertypes
// [primary_supers 0]
// [primary_supers 1]
// [primary_supers 2]
// ...
// [primary_supers 7]
// [java_mirror      ]
// [super            ]
// [name             ]
// [first subclass]
// [next_sibling    ] link to chain additional subclasses
// [modifier_flags]
// [access_flags    ]
// [verify_count    ] - not in product
// [alloc_count     ]
// [last_biased_lock_bulk_revocation_time] (64 bits)
// [prototype_header]
```

For 32 bit JVM

[header]	4 byte
[klass pointer]	4 byte
[C++ vtbl ptr]	4 byte
[layout_helper]	4 byte
[super_check_offset]	4 byte
[secondary_super_cache]		4 byte
[secondary_supers]	4 byte (because of pointer to array)
[primary_supers]	4 byte (because of pointer to array)
[java_mirror]	4 byte
[super]	4 byte

For 32 bit JVM, as you can see offset of “**super**” field is “**36**”.

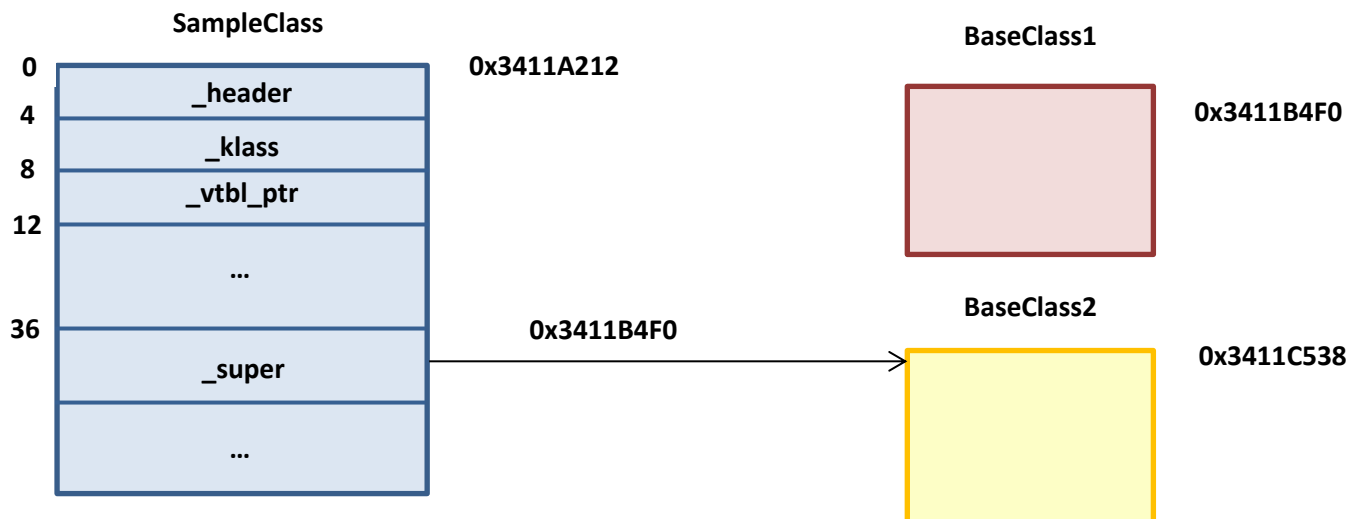


```
try {
    BaseClass2 baseObj1 = (BaseClass2)(Object)new SampleClass(); // ClassCastException
} catch (ClassCastException e) {
    e.printStackTrace();
}
long addressOfClass = JvmUtil.addressOfClass(SampleClass.class); // 0x3411A212
long addressOfBaseClass2 = JvmUtil.addressOfClass(BaseClass2.class); // 0x3411C538

// Set super class pointer of "SampleClass" to "BaseClass2" class address
unsafe.putLong(addressOfClass + 36, addressOfBaseClass2);

try {
    BaseClass2 baseObj2 = (BaseClass2)(Object)new SampleClass(); // No Exception
} catch (ClassCastException e) {
    e.printStackTrace();
}
```

After executing code above;



After executing code above;

Memory layout of “**SampleClass**” class (128 bytes from its starting address):

```
[0x0000]: 01 00 00 00 a8 08 97 38 2c 7a 87 65 18 00 00 00
[0x0010]: 28 00 00 00 40 9f 6e 04 00 00 00 00 00 00 f7 37
[0x0020]: 00 00 97 38 38 c5 11 34 a0 c6 11 34 00 00 00 00
[0x0030]: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[0x0040]: 40 87 5d 24 38 c5 11 34 00 00 00 00 00 00 00 00
[0x0050]: 09 00 00 00 21 00 20 20 00 00 00 00 00 00 00 00
[0x0060]: 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00
[0x0070]: 00 00 00 02 00 00 00 00 a0 c9 11 34 20 c3 11 34
```

As you can see, **highlighted field** is address of “**BaseClass2**” class.

Memory layout of “**BaseClass2**” class (128 bytes from its starting address):

```
[0x0000]: 01 00 00 00 a8 08 97 38 2c 7a 87 65 10 00 00 00
[0x0010]: 24 00 00 00 a8 b0 9e 00 00 00 00 00 00 00 f7 37
[0x0020]: 00 00 97 38 38 c5 11 34 00 00 00 00 00 00 00 00
[0x0030]: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[0x0040]: d0 86 5d 24 00 00 97 38 a0 c6 11 34 60 bd 11 34
[0x0050]: 09 00 00 00 21 00 20 20 00 00 00 00 00 00 00 00
[0x0060]: 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00
[0x0070]: 00 00 ff 01 00 00 00 00 58 c8 11 34 90 c4 11 34
```

As you can see, **highlighted field** is address of “**BaseClass2**” class (**BaseClass2** class has no super class, so its super class itself).

You can access all demo codes from <https://github.com/serkan-ozal/jillegal-classredefine-demo>

--

Serkan ÖZAL

<https://github.com/serkan-ozal>