

C++ Class Implementation

```
struct MyStruct
{
    int myInt;
    double myDouble;
    char *myCString
};
```

```
struct MyStruct
{
    int myInt;
    double myDouble;
    char *myCString
};
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

```
struct MyStruct
{
    int myInt;
    double myDouble;
    char *myCString
};
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

```
MyStruct* mc = new MyStruct;
mc->myCString = "This is a string!"
```

```
struct MyStruct
{
    int myInt;
    double myDouble;
    char *myCString
};
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

```
MyStruct* mc = new MyStruct;
mc->myCString = "This is a string!"
```

Implementation

1. Go to the location pointed at by mc.
2. Go forward 12 bytes.
3. Store a pointer to "This is a string!"

```
class MyClass
{
public:
    int myInt;
    double myDouble;
    char *myCString
};
```

```
class MyClass
{
public:
    int myInt;
    double myDouble;
    char *myCString
};
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

```
class MyClass
{
public:
    int myInt;
    double myDouble;
    char *myCString
};
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

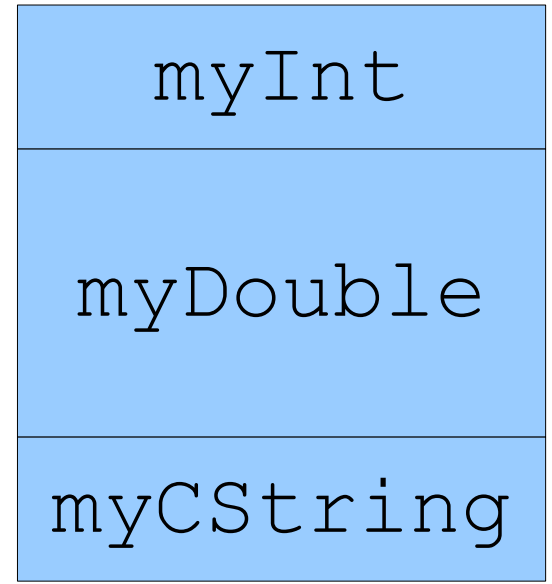
myCString

```
MyClass* mc = new MyClass;
mc->myCString = "This is a string!"
```



```
class MyClass
{
public:
    int myInt;
    double myDouble;
    char *myCString
};
```

Offset 0
Offset 4
Offset 8
Offset 12



```
MyClass* mc = new MyClass;
mc->myCString = "This is a string!"
```

Implementation

1. Go to the location pointed at by mc.
2. Go forward 12 bytes.
3. Store a pointer to "This is a string!"

```
class Base
{
public:
    int myInt;
    double myDouble;
    char *myCString
};

class Derived: public Base
{
public:
    int dInt;
    double dDouble;
};
```

```
class Base
{
public:
    int myInt;
    double myDouble;
    char *myCString
};

class Derived: public Base
{
public:
    int dInt;
    double dDouble;
};
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

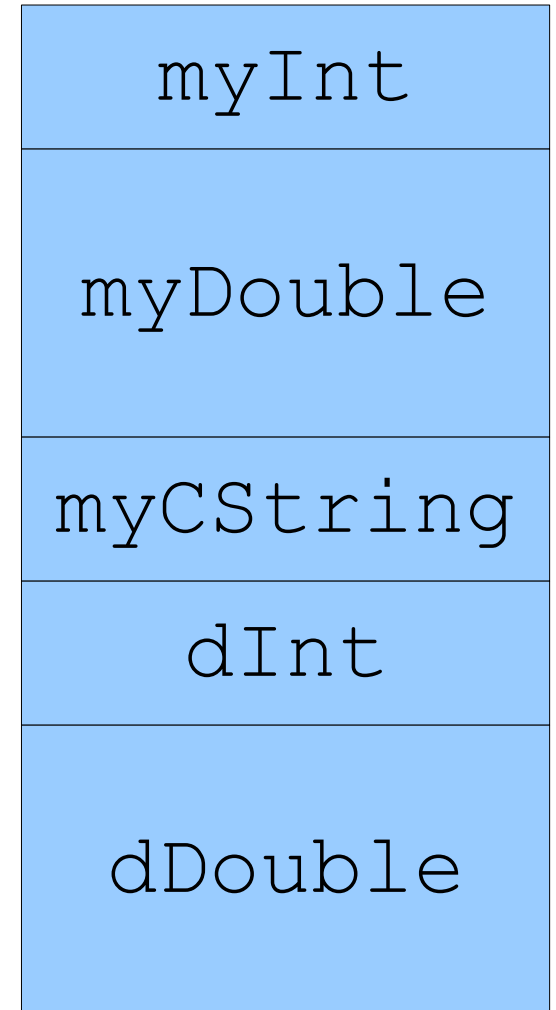
Offset 16

dInt

Offset 20

dDouble

Offset 24



```
class Base
{
public:
    int myInt;
    double myDouble;
    char *myCString
};

class Derived: public Base
{
public:
    int dInt;
    double dDouble;
};
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

Offset 16

dInt

Offset 20

dDouble

Offset 24

```
class Base
{
public:
    int myInt;
    double myDouble;
    char *myCString
};

class Derived: public Base
{
public:
    int dInt;
    double dDouble;
};
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

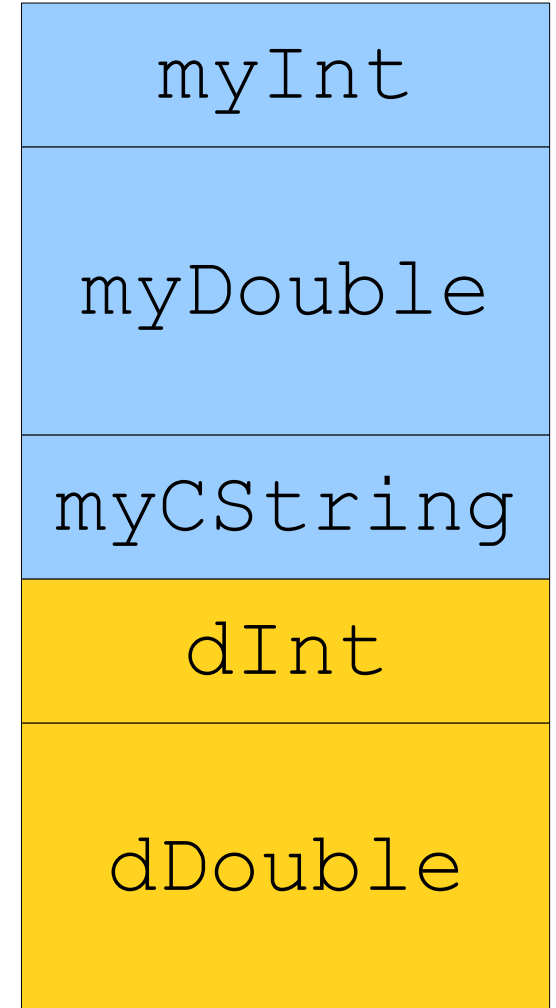
Offset 16

dInt

Offset 20

dDouble

Offset 24



```
class Base
{
public:
    int myInt;
    double myDouble;
    char *myCString
};

class Derived: public Base
{
public:
    int dInt;
    double dDouble;
};
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

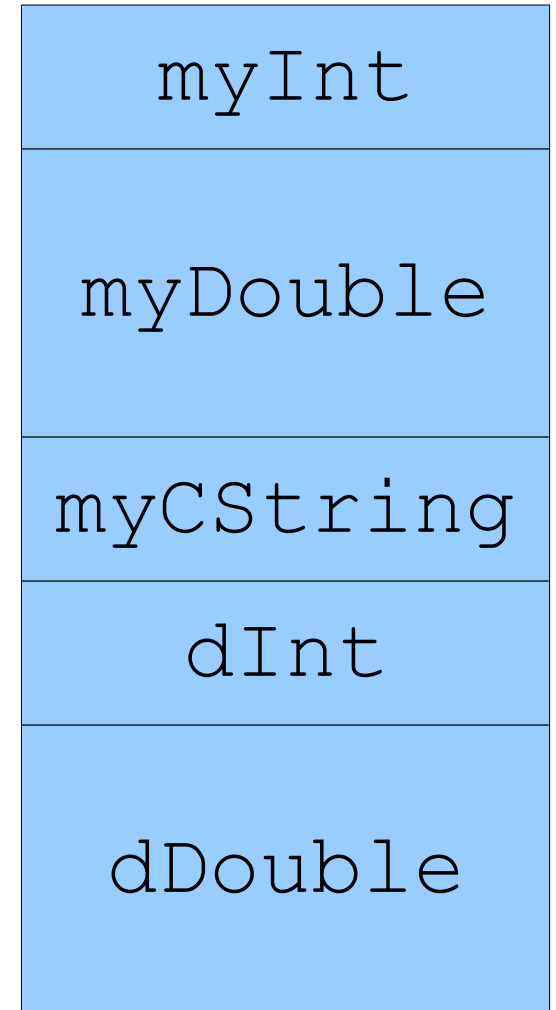
Offset 16

dInt

Offset 20

dDouble

Offset 24



```
class Base
{
public:
    int myInt;
    double myDouble;
    char *myCString
};

class Derived: public Base
{
public:
    int dInt;
    double dDouble;
};

Base* mc = new Derived;
mc->myCString = "This is a string!"
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

myCString

Offset 12

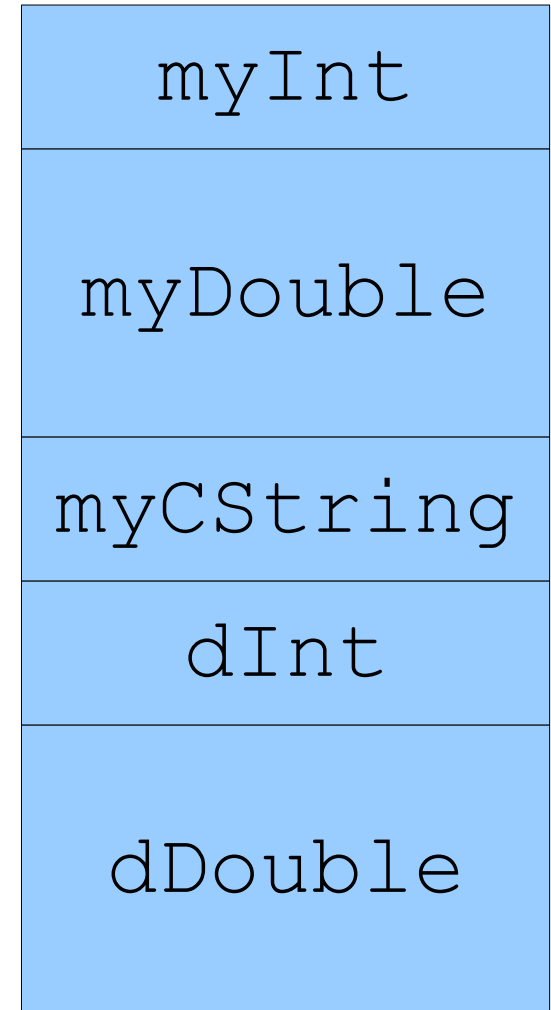
Offset 16

dInt

Offset 20

dDouble

Offset 24



```
class Base
{
public:
    int myInt;
    double myDouble;
    char *myCString
};
```

```
class Derived: public Base
{
public:
    int dInt;
    double dDouble;
};
```

```
Base* mc = new Derived;
mc->myCString = "This is a string!"
```

Offset 0

Offset 4

Offset 8

Offset 12

Offset 16

Offset 20

Offset 24

myInt

myDouble

myCString

dInt

dDouble

Implementation

1. Go to the location pointed at by mc.
2. Go forward 12 bytes.
3. Store a pointer to "This is a string!"


```
class MyClass
{
public:
    void myFn(int myInt);
private:
    int myInt;
    double myDouble;
    char *myCString
};
```

```
class MyClass
{
public:
    void myFn(int myInt);
private:
    int myInt;
    double myDouble;
    char *myCString
};
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

```
class MyClass
{
public:
    void myFn(int myInt);
private:
    int myInt;
    double myDouble;
    char *myCString
};
```

```
MyClass* mc = new MyClass;
mc->myFn(137);
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

```
class MyClass
{
public:
    void myFn(int myInt);
private:
    int myInt;
    double myDouble;
    char *myCString
};
```

```
MyClass* mc = new MyClass;
mc->myFn(137);
```

Implementation
myFn(mc, 137);

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

```
class MyClass
{
public:
    void myFn(int myInt);
private:
    int myInt;
    double myDouble;
    char *myCString
};
```

Offset 0

myInt

Offset 4

myDouble

Offset 8

Offset 12

myCString

```
MyClass* mc = new MyClass;
mc->myFn(137);
```

Implementation

```
myFn(mc, 137); // Just a regular function call!
```

```
class MyClass
{
public:
    void myFn(int myInt);
    virtual void vFn();
private:
    int myInt;
    double myDouble;
    char *myCString
};
```

```
class MyClass
{
public:
    void myFn(int myInt);
    virtual void vFn();
private:
    int myInt;
    double myDouble;
    char *myCString
};
```

Offset 0

Vtable*

Offset 4

myInt

Offset 8

myDouble

Offset 12

Offset 16

myCString

```
class MyClass
{
public:
    void myFn(int myInt);
    virtual void vFn();
private:
    int myInt;
    double myDouble;
    char *myCString
};
```

Offset 0

Vtable*

Offset 4

myInt

Offset 8

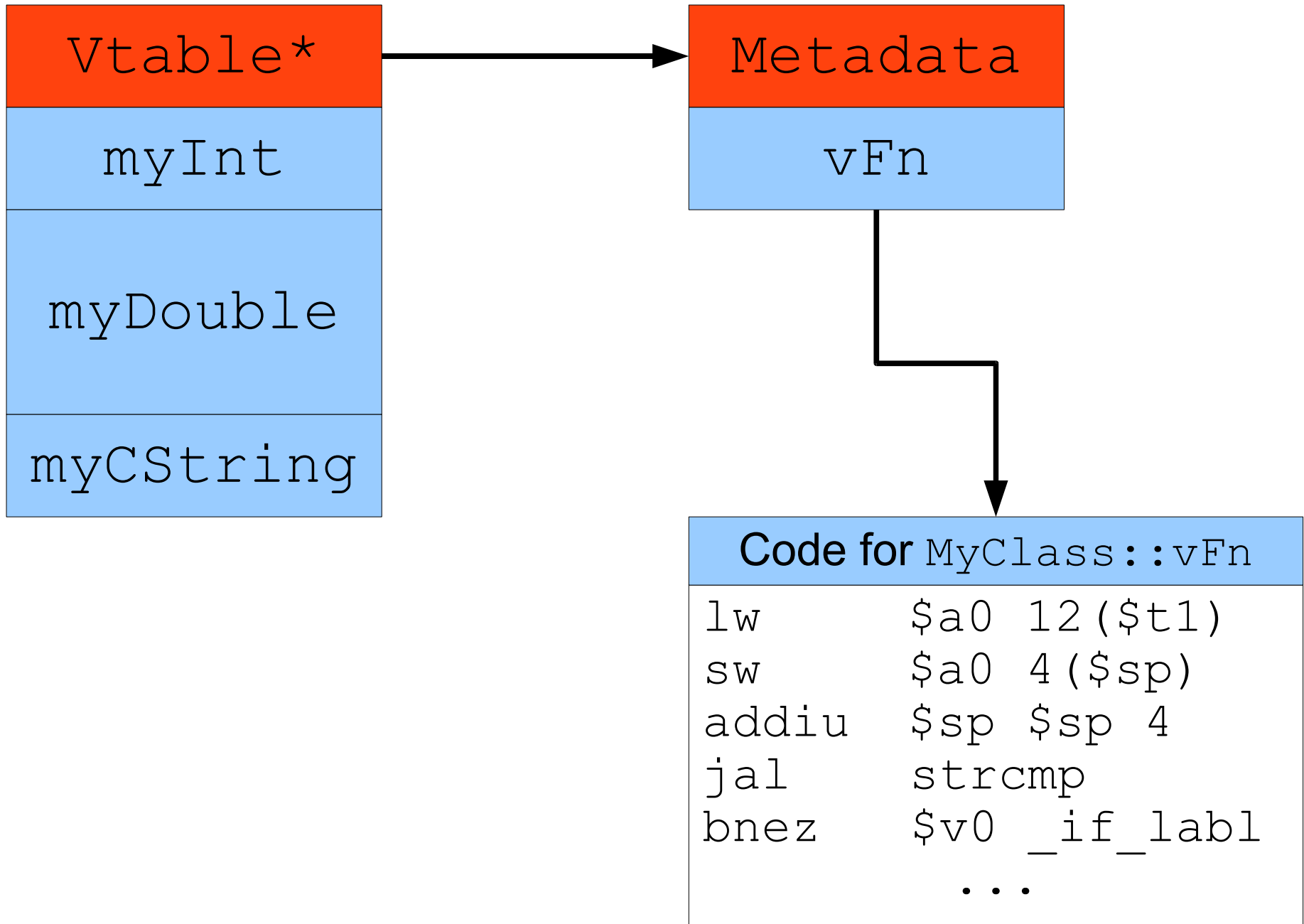
myDouble

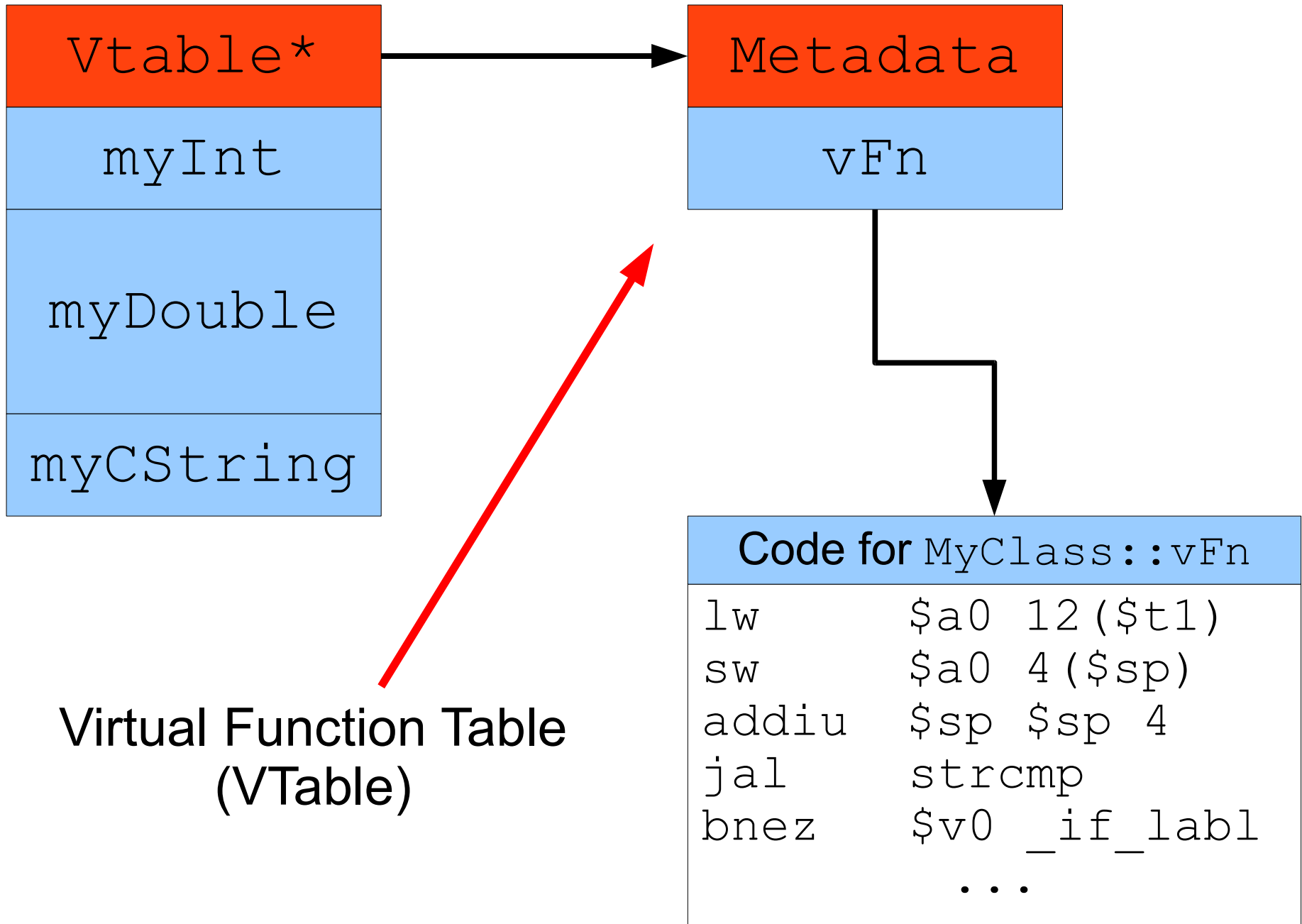
Offset 12

Offset 16

myCString

What's this Vtable* thing?





```
class MyClass
{
public:
    void myFn(int myInt);
    virtual void vFn();
private:
    int myInt;
    double myDouble;
    char *myCString
};
```

Offset 0

Vtable*

Offset 4

myInt

Offset 8

myDouble

Offset 12

Offset 16

myCString

```
class MyClass
{
public:
    void myFn(int myInt);
    virtual void vFn();
private:
    int myInt;
    double myDouble;
    char *myCString
};
```

```
MyClass* mc = new MyClass;
mc->vFn();
```

Offset 0

Vtable*

Offset 4

myInt

Offset 8

myDouble

Offset 12

Offset 16

myCString

```

class MyClass
{
public:
    void myFn(int myInt);
    virtual void vFn();
private:
    int myInt;
    double myDouble;
    char *myCString
};

```

Offset 0

Vtable*

Offset 4

myInt

Offset 8

myDouble

Offset 12

Offset 16

myCString

```

MyClass* mc = new MyClass;
mc->vFn();

```

Implementation

1. Go to the object pointed at by mc.
2. Go to the vtable.
3. Look up which version of **vFn** to call.
4. Call **vFn(mc)**