

コピーコンストラクタの注意点

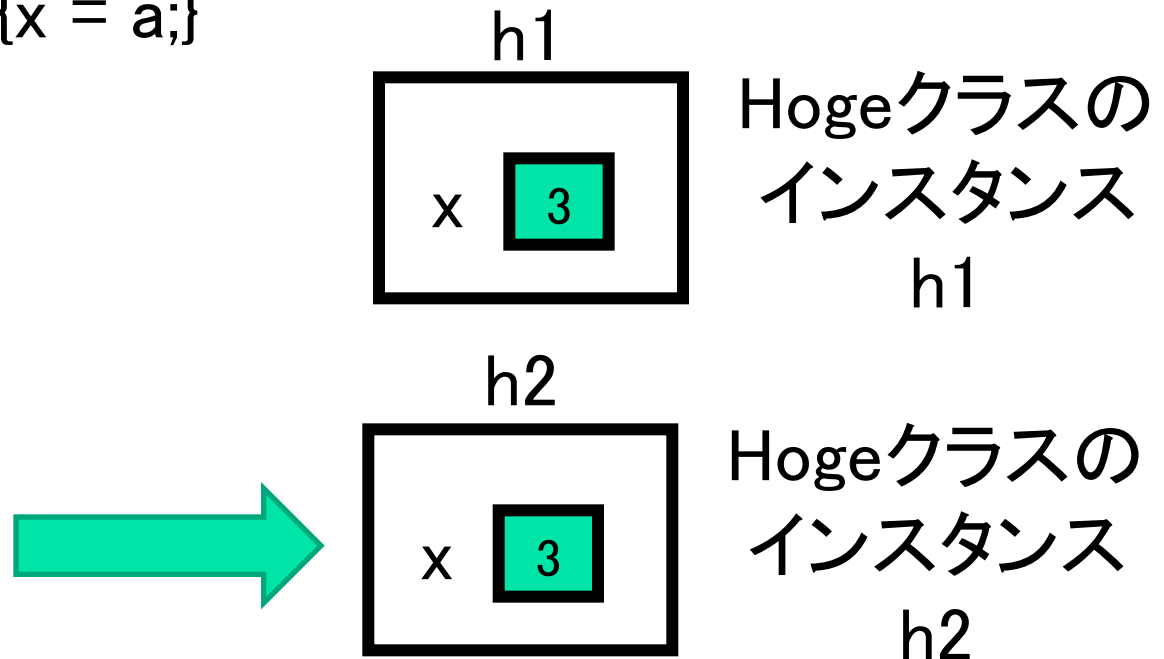
コピーコンストラクタ(OKの例)

```
class Hoge {  
    int x;  
public:
```

```
    ...  
    Hoge(const Hoge& h) {x = h.x;}  
    void set(int a) {x = a;}  
};
```

```
int main()  
{  
    Hoge h1;  
    h1.set(3);  
    Hoge h2(h1);  
}
```

コピーコンストラクタで
メンバ変数xの値をコピー

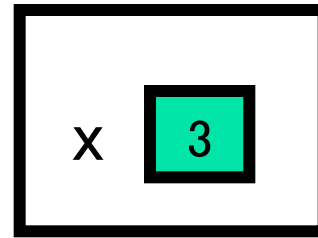


未定義でもOKの場合

```
class Hoge {  
    int x;  
public:  
    ...  
    void set(int a) {x = a;}  
};  
  
int main()  
{  
    Hoge h1;  
    h1.set(3);  
    Hoge h2(h1);  
}
```

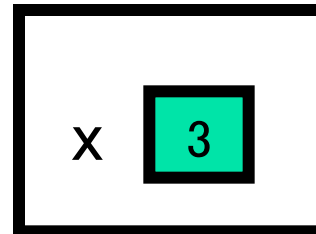
コピーコンストラクタ未定義
の場合、単純コピーを作成
(Hogeクラスでは問題ない)

h1



Hogeクラスの
インスタンス
h1

h2



Hogeクラスの
インスタンス
h2



未定義だとNGの場合

```
class MyString {
```

```
    char* s;
```

```
public:
```

```
    MyString(const char* n)
```

```
    {s = new char[strlen(n) + 1];
```

```
    strcpy(s, n);} 
```

```
    ~MyString{delete [] s;}
```

```
};
```

```
int main()
```

```
{
```

```
    MyString s1("hello");
```

```
    MyString s2 = s1;
```

```
}
```

ポインタをメンバに持ち、
動的確保と解放を行う場合

未定義だとNGの場合

```
class MyString {
```

```
    char* s;
```

```
public:
```

```
    MyString(const char* n)
```

```
    {s = new char[strlen(n) + 1];
```

```
     strcpy(s, n);}~MyString{delete [] s;
```

```
};
```

```
int main()
```

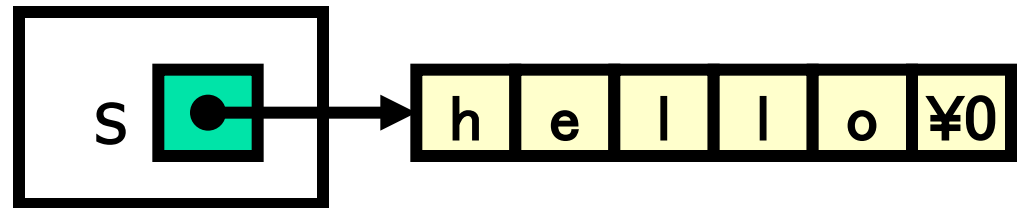
```
{
```

```
    MyString s1("hello");
```

```
    MyString s2 = s1;
```

```
}
```

“hello¥0”の領域確保
sで先頭アドレス保持



Stringクラスの
インスタンス

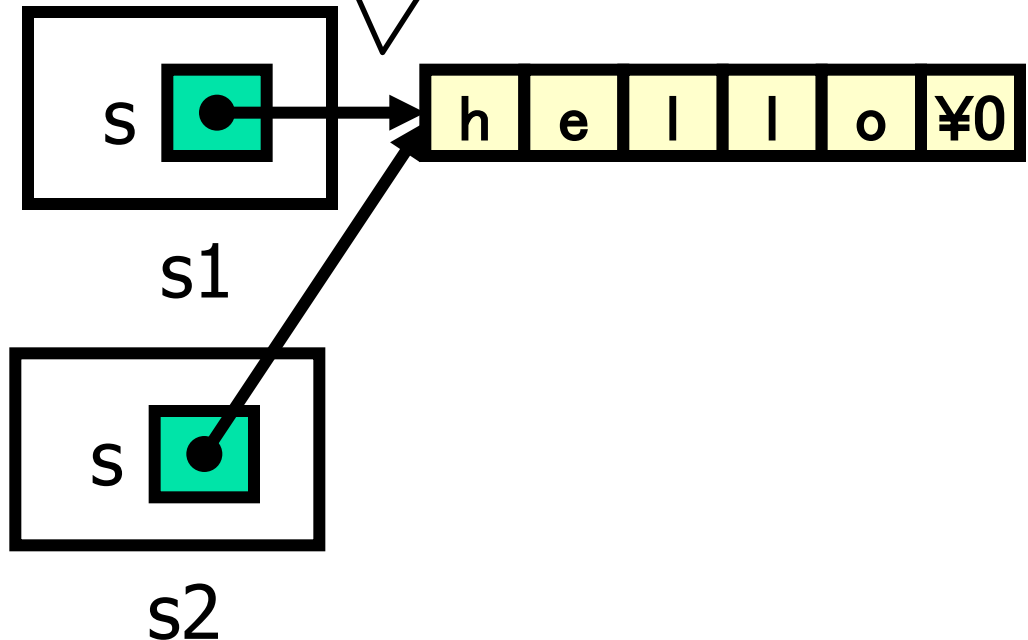
s1

未定義だとNGの場合

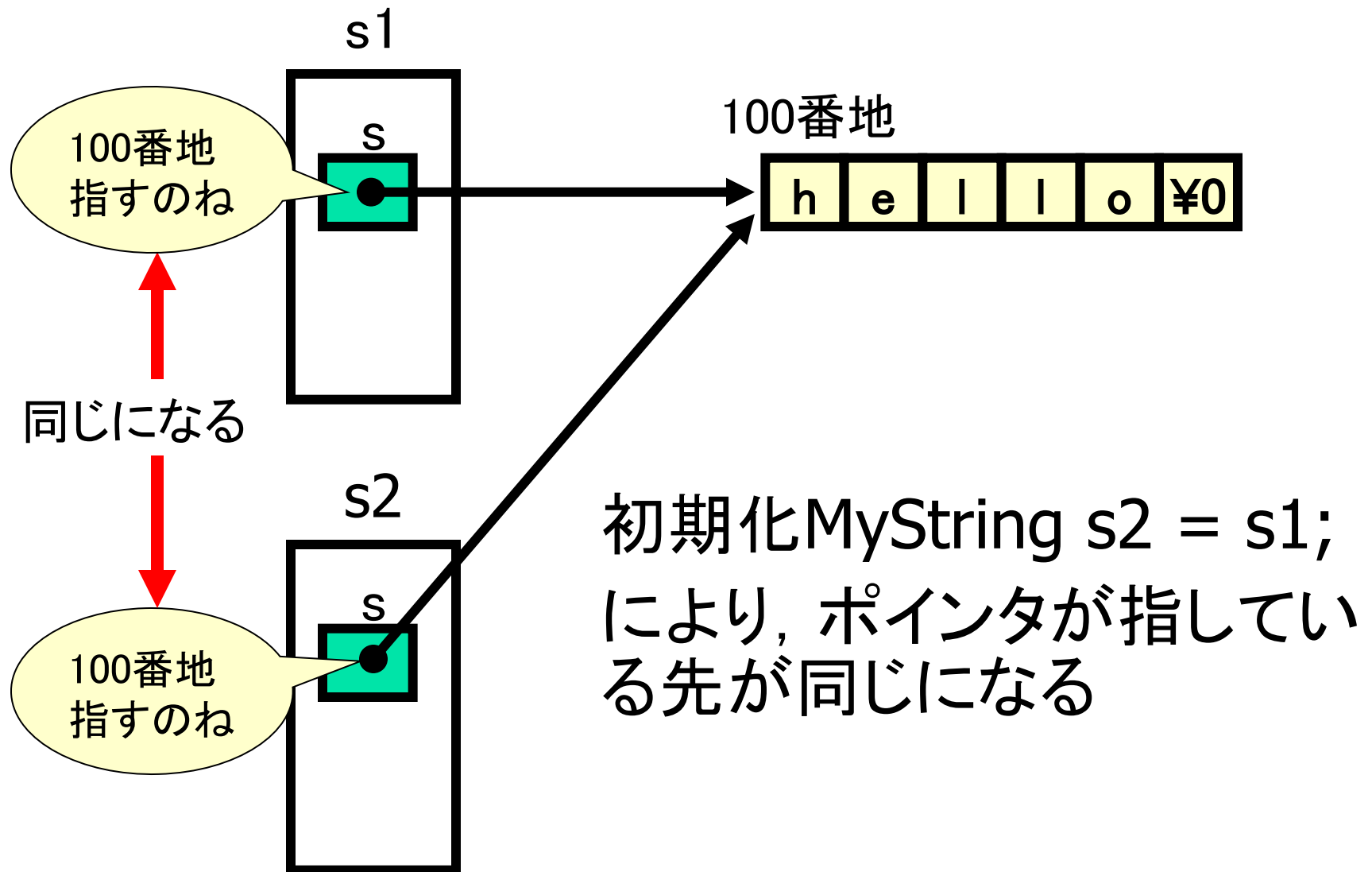
```
class MyString {  
    char* s;  
public:  
    MyString(const char* n)  
    {s = new char[strlen(n) + 1];  
     strcpy(s, n);}  
    ~MyString{delete [] s;}  
};
```

```
int main()  
{  
    MyString s1("hello");  
    MyString s2 = s1;  
}
```

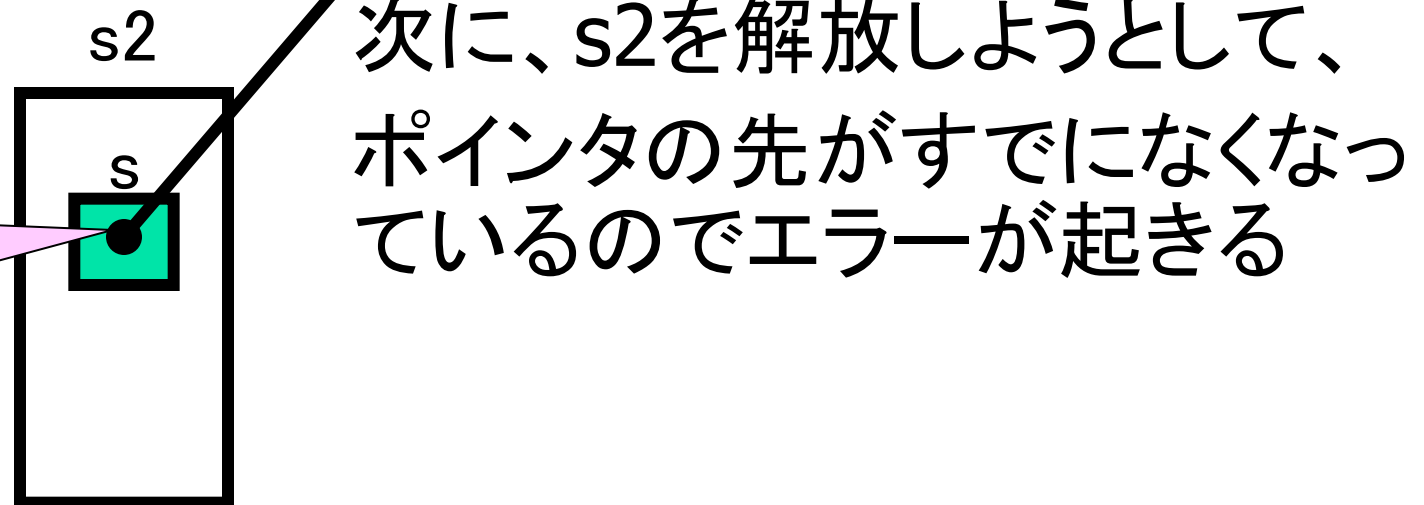
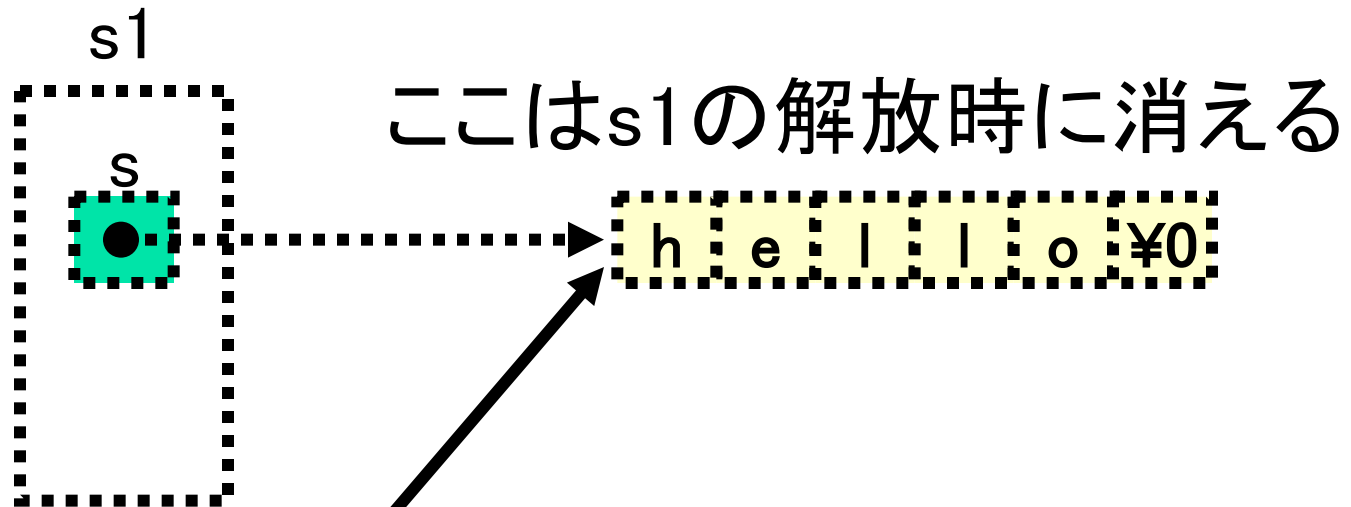
単純コピーだとポインタが
コピーされ、同じアドレスを
指してしまう



なにが起きているか



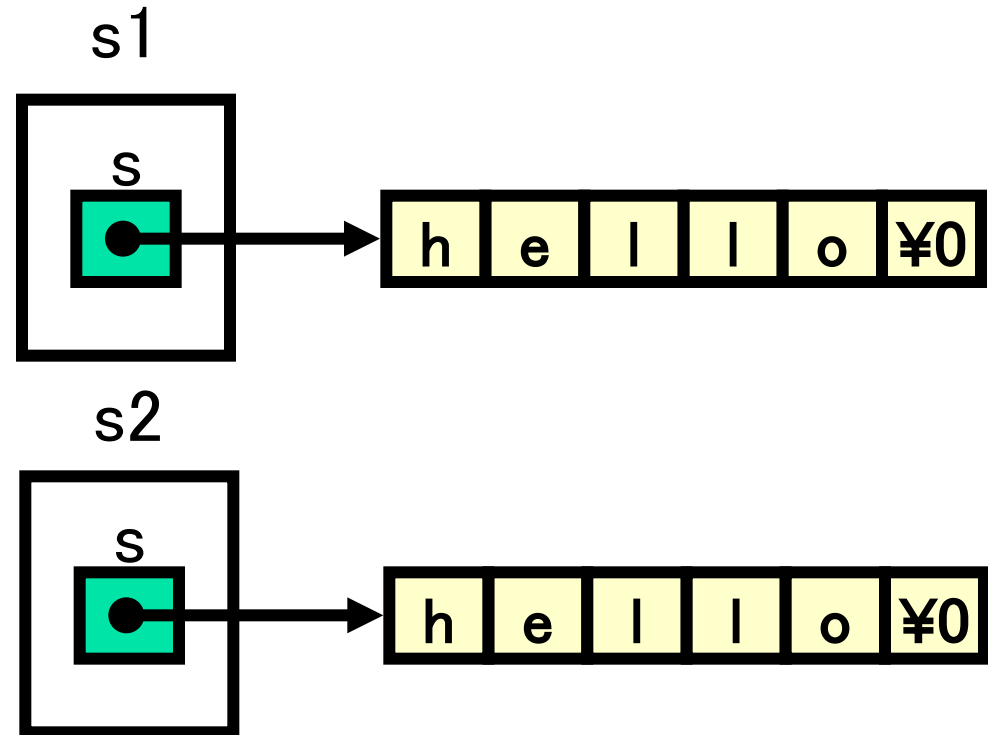
s1のデストラクタが呼ばれると



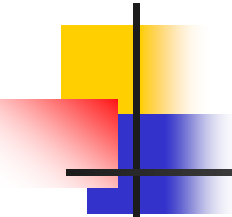
えっ?!
消えてる

コピーコンストラクタの注意点

```
int main()
{
    MyString s1("hello");
    MyString s2 = s1;
}
```



メモリ上の別の領域に、s1の文字列をコピーしたものが作成されるように、コピーコンストラクタを定義する必要がある



練習課題の注意点

```
MyString::MyString(const MyString &ref)
{
    // 定義部(メンバ変数のコピー方法を記述)
    // 実装は引数付きコンストラクタを参考にする
}

int main()
{
    // 引数付きコンストラクタが呼ばれる
    MyString s1("t123456A");
    // s1を実引数としてコピーコンストラクタが呼ばれる
    MyString s2 = s1;
}
```