# More on Basic C++ Programming

Hsuan-Tien Lin

Dept. of CSIE, NTU

February 26, 2013

# What We Have Done (Chapter 1)

- C++ (in class): I/O, class, access control, variable declaration, scope
- C++ (in homework): new/delete constructor, operator overloading,
- C++ (in reading): constant, typedef, namespace, expression, casting, control flow, functions, inline, C++ Programming
- TA hours (fabulous!): many more

# What We Will Do (Chapter 1)

- pointers and references
- template and STL

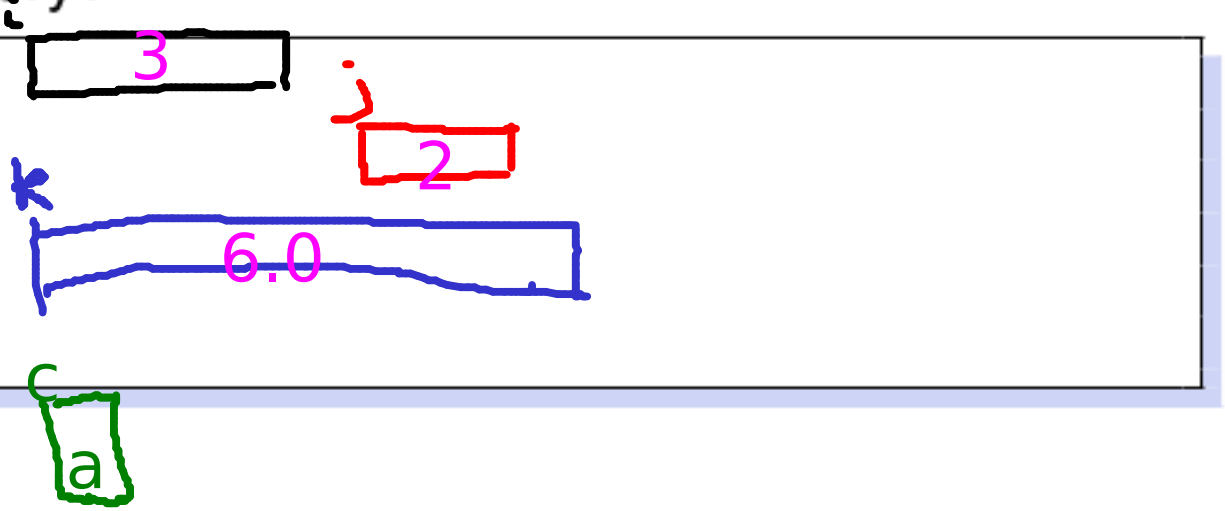
# What We Will Not Do (Chapter 2)

- Object Oriented Programming

# Fun Time (1)

What happens in memory?

```
1 int i;
2 short j;
3 double k;
4 char c = 'a';
5 i = 3; j = 2;
6 k = i * j;
```



# Life Cycle of a (Primitive) Variable

- declared and created

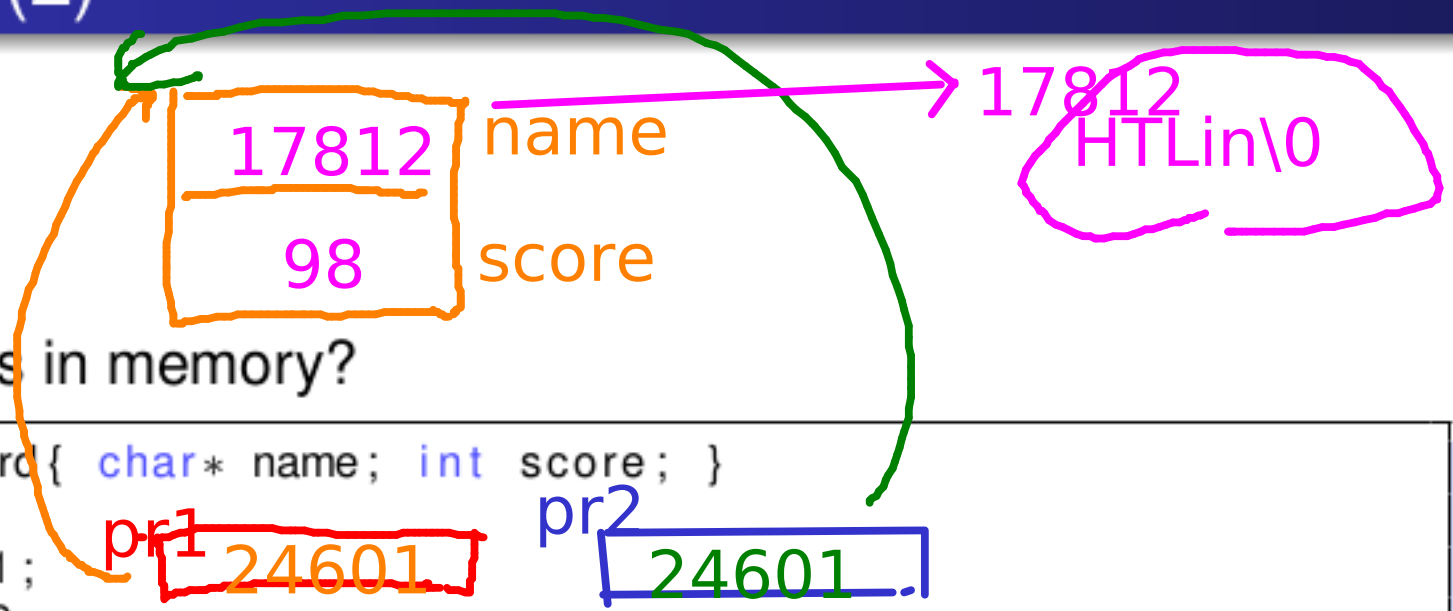
```
1  int count;
```

- used and modified

```
1  count += 1;
```

- destroyed
  –automatically (when out of scope)

# Fun Time (2)




What happens in memory?

```
1  class Record{ char* name; int score; }
2
3  Record* pr1;
4  Record* pr2;
5  pr1 = new Record();
6  pr2 = pr1;   //how many records are there?
7  pr1->name = "HTLin";
8  pr2->score = 98;
```

# Fun Time (3)


What happens in memory?

```
1  class Record{ char* name; int score; }
2
3  Record r1;
4  Record r2;
5  r2 = r1;  //how many records are there?
6  r1.name = "HTLin";
7  r2.score = 98;
```

# Life Cycle of an Object Instance (C++)

- pointer declared

```
1  Record* pr;
```

- instance created

```
1  pr = new Record();
```

- used and modified

```
1  cout << pr->name;
```

- destroyed

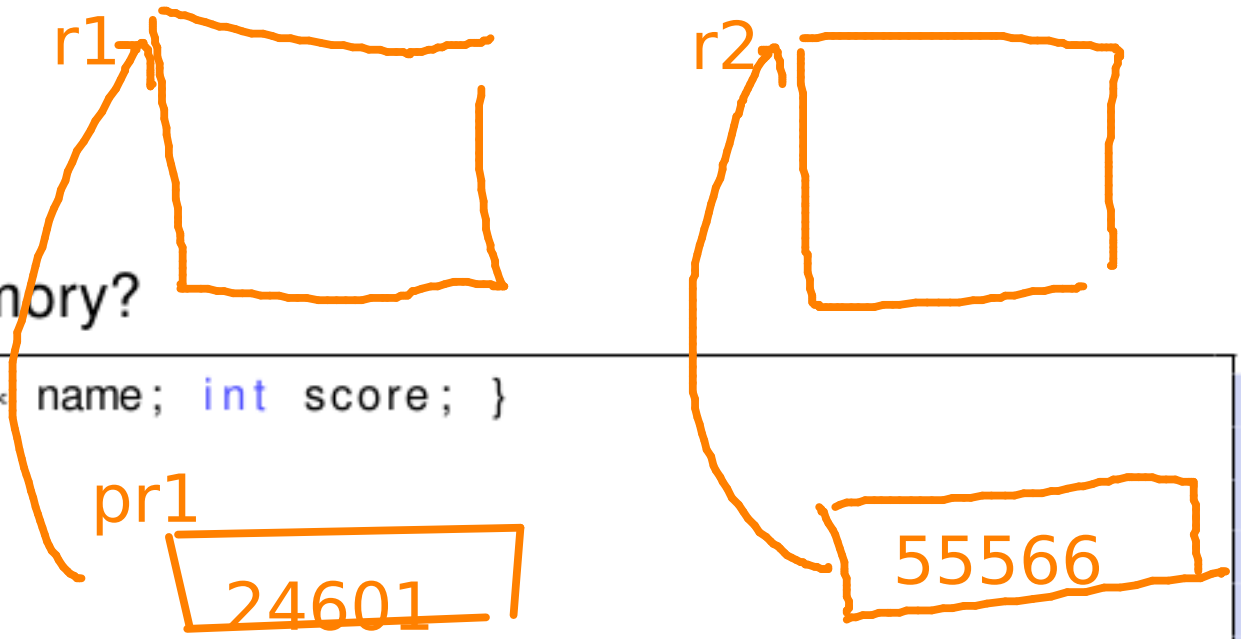
```
1  delete pr;
```

# Pointer: Key Point

- a instance occupies a space in the memory;
  老太太住在屏東一個房子裡面
- pointer: the “address” to the instance;
  用"海角七號"就可以找到老太太
- pointer variable: holds the pointer;
  一個"信封"，上面寫著海角七號
- any operation on the instance goes thru the pointer;
  要請老太太"回憶"時，拿個信封上寫"海角七號"，接著寫"回憶"，就會使命必達了

```
老人*  信封= new 老人(老太太資料);
信封->回憶();
```

# Fun Time (4)



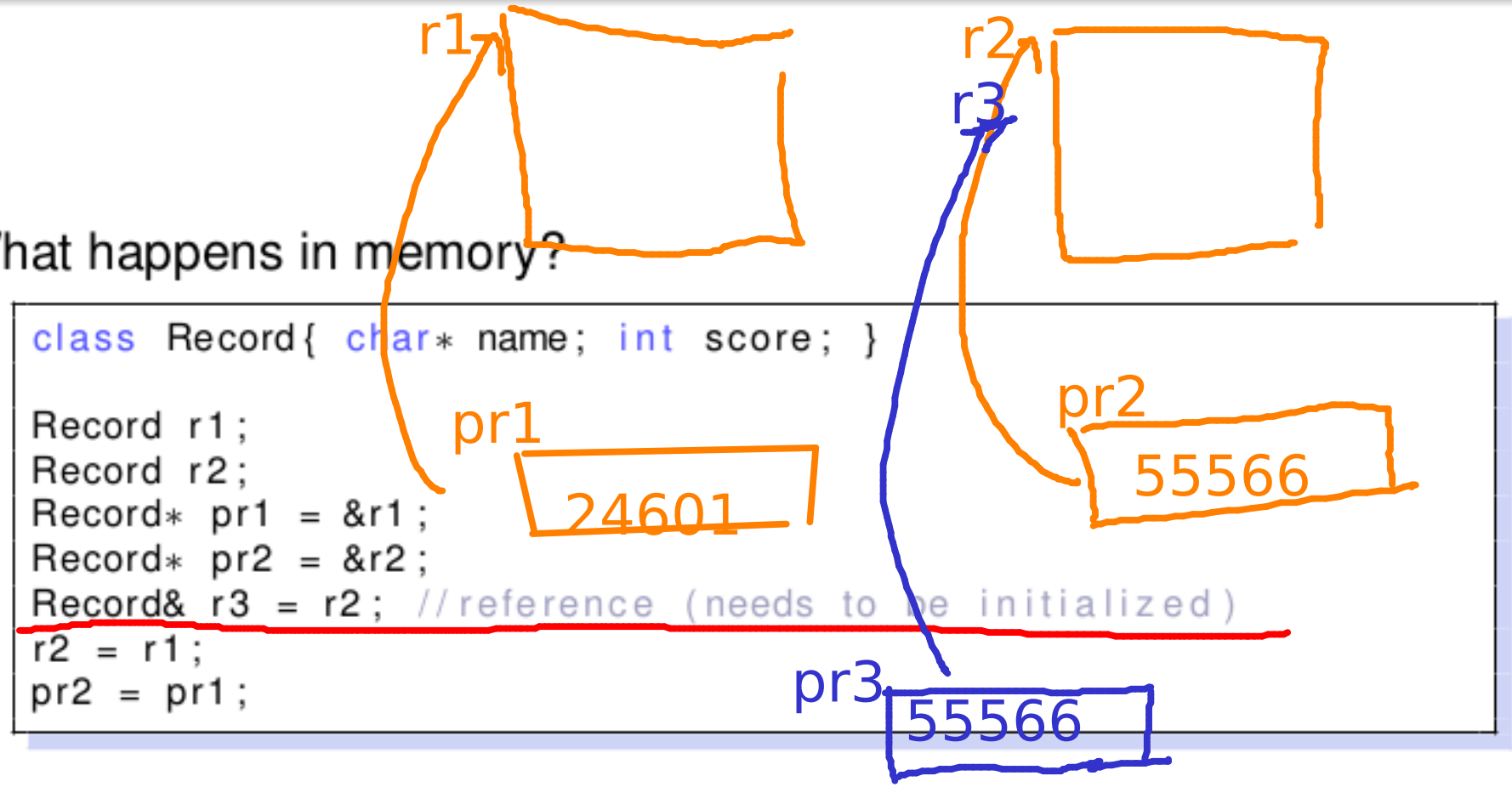

What happens in memory?

```
1  class Record{ char* name; int score; }
2
3  Record r1;
4  Record r2;
5  Record* pr1 = &r1;
6  Record* pr2 = &r2;
7  r2 = r1;
8  pr2 = pr1;
```

# Fun Time (5)

What happens in memory?

```
1  class Record{ char* name; int score; }
2
3  Record r1;
4  Record r2;
5  Record* pr1 = &r1;
6  Record* pr2 = &r2;
7  Record& r3 = r2; //reference (needs to be initialized)
8  r2 = r1;
9  pr2 = pr1;
```




r3.name

pr3->name

# Pointer/Reference: Key Point

- pointer: the “address” to the instance;
  用"海角七號"就可以找到老太太
- reference: the “other name” of the instance;
  用"小島友子"也可以稱呼老太太
- the “original” variable: holds the reference;
  一張"身份證"，上面寫著(住海角七號)
- pointer variable: holds the pointer;
  一個"信封"，上面寫著海角七號
- reference variable: holds the reference;
  一張"名片"，上面寫著(住海角七號)

```
老人  老太太;
老人*  信封= &老太太; //存放住址
老人& 小島友子= 老太太; //用別名
      老太太.回憶();
      信封->回憶();
      小島友子.回憶();
```