Introduction to C++ (Season 1)

Unit 3: More than C

第3单元:更上一层楼 – 超越C的语法

Section 3 : Reference & Dynamic Memory

第3节：引用与动态内存管理

Reference

引用

Reference (引用)

- A **reference** is an alias for another variable. (引用就是另一个变量的别名)
- Any changes made through the reference variable are actually performed on the original variable (通过引用所做的读写操作实际上是作用于原变量上).

- To declare a reference variable:

```
int x;
int&  rx = x;
```

- or

```
int x, &rx = x;
```

51. C++ pointers and references should have their reference symbol next to the type rather than to the name.
51. C++指针与引用符号应靠近其类型而非名字。

```
例如：   float* x; // NOT: float *x;
         int&   y; // NOT: int &y;
```

→TestReferenceVariable

Function Parameters: Pass By Reference (函数参数：引用传递)

- ❖ You can use a reference variable as a parameter in a function and pass a regular variable to invoke the function. (引用可做函数参数，但调用时只需传普通变量即可)
- ❖ When you change the value through the reference variable, the original value is actually changed. (在被调函数中改变引用变量的值，则改变的是实参的值)

→TestPassByReference

Comparision: 3 swap() functions

```
//pass by value
void swap(int x,  int y){
  int  t;
  t=x;  x=y;  y=t;
}
int main() {
  int a(5), b(10);
  cout << "Before: a=" << a <<
         " b=" << b << endl;
    swap( a,  b );
  cout << "After:  a=" << a <<
          "b=" << b << endl;
  return 0;
}
```

```
Before: a=5 b=10
After:  a=5 b=10
```

```
//pointer as formal params
void swap(int* x,  int* y){
  int  t;
  t=*x;  *x=*y;  *y=t;
}
int main() {
  int a(5), b(10);
  cout<< "Before: a=" << a <<
         " b=" << b << endl;
  swap( &a,  &b );
  cout<< "After: a=" << a <<
      <<"b="<<b<<endl;
    return 0;
}
```

```
Before: a=5 b=10
After:  a=10 b=5
```

```
//reference as formal params
void swap(int& x,  int& y){
  int  t;
  t=x;  x=y;  y=t;
}
int main() {
  int a(5), b(10);
  cout<< "Before: a=" << a <<
         " b=" << b << endl;
  swap( a,  b );
  cout << "After: a=" << a <<
          "b="<<b<<endl;
  return 0;
}
```

```
Before: a=5 b=10
After:  a=10 b=5
```

Dynamic Memory

动态内存管理

Dynamic memory: Allocate/Release

- C++中通过运算符new申请动态内存

```
new  <类型名> (初值) ;              //申请一个变量的空间
new  <类型名>[常量表达式] ; //申请数组
```

  - 如果申请成功，返回指定类型内存的地址；
  - 如果申请失败，返回空指针(整数0)。

- 动态内存使用完毕后，要用delete运算符来释放。

```
delete  <指针名>; //删除一个变量/对象
delete []<指针名>; //删除数组空间
```

70. "0" should be used instead of "NULL".
70. 用“0”代替“NULL” is part of the standard C library, but is made obsolete in C++.。
因为“NULL”是C语言标准库的内容，但是在C++中已经废止了。

Dynamic memory: Examples

	C	C++

Allocate
Release
	malloc();
	free();
		new
		delete

Example 1	char* s = (char*)malloc(1);
	free(s);
		char* s = new char(97);
		delete s;

Example 2	int* p = (int*) malloc(4*10);
	free(p);
		int* p = new int[10];
		delete [] p;

Example 3	int** q = (int**) malloc(4*10*3);
	free(q);
		~~int** q = new int[10][3];~~
		int (*q)[3] = new int[10][3];
		（暂不做要求）
		delete [] p;

int**   q = new int[10][3];
error C2440: “初始化” : 无法从 “int (*)[3]”转换为 “int **”、