C++ Note 1

C++ pass the direct parameter to the function, there are two ways, directly parameter and referred parameter.

1：

1. #include <iostream>
2. **using** **namespace** std;
4. **int** addition (**int** a, **int** b)
5. {
6. **int** r;
7. r=a+b;
8. **return** r;
9. }
11. **int** main ()
12. {
13. **int** z;
14. **int** x = 5, y = 3;
15. z = addition (x,y);
16. cout << "The result is " << z;

Output:8

This way would not change the value of x and y after executing the function

1. #include <iostream>
2. **using** **namespace** std;
4. **void** duplicate (**int** a, **int** b, **int** c)
5. {
6. a\*=2;
7. b\*=2;
8. c\*=2;
9. }
11. **int** main ()
12. {
13. **int** x=1, y=3, z=7;
14. duplicate (x, y, z);
15. cout << "x=" << x << ", y=" << y << ", z=" << z;
16. **return** 0;
17. }

If use the reference way to pass parameter.

1. #include <iostream>
2. **using** **namespace** std;
4. **void** duplicate (**int**& a, **int**& b, **int**& c)
5. {
6. a\*=2;
7. b\*=2;
8. c\*=2;
9. }
11. **int** main ()
12. {
13. **int** x=1, y=3, z=7;
14. duplicate (x, y, z);
15. cout << "x=" << x << ", y=" << y << ", z=" << z;
16. **return** 0;
17. }

Output:2，6，14。

This is a int value to pass, but if there is a mixed data type, the reference way would be more effiencient.

|  |  |
| --- | --- |
| 3  4 | string concatenate (string a, string b)  {  return a+b;  } |

1. string concatenate (string& a, string& b)
2. {
3. **return** a+b;
4. }

And to fix the value changed problem, The code:

1. string concatenate (**const** string& a, **const** string& b)
2. {
3. **return** a+b;
4. }