Cse332\_ Copy Control 2

1. Jing Lu, Seed(Peiyun) Zeng
2. Note that n1 or n2 is not r\_value, so we cannot create r\_value reference to n1 or n2; But l\_value and int can be assigned to each other.

int n1 = 1;

int n2 = 2;

int &ref\_n1 = n1;

int &ref\_n2 = n2;

//this won't work because n1 is an l\_value, we have to use move

int &&r\_ref\_n1 = n1;

int &&r\_ref\_n2 = n2;

3. after using move function, the two int variables could be printed out and could be initialized to other int variables.Although move() is supposed to remove the int variables from the memory but it seems that the compiler did not delete them yet.

int &&r\_ref\_n1 = std::move(n1);

int &&r\_ref\_n2 = std::move(n2);

cout << r\_ref\_n1 << endl;

cout << r\_ref\_n2 << endl;

int x = r\_ref\_n1;

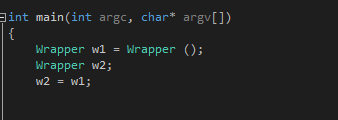
int y = r\_ref\_n2;

cout << x << endl;

cout << y << endl;

4.

Input:



Output:



We tried the following :

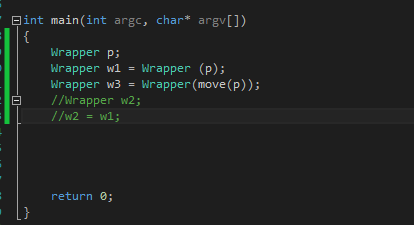
Wrapper w1 = new Wrapper(); //w1 will be numbered as 1

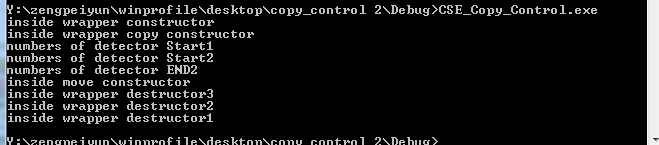
Wrapper w2; //w2 will be numbered as 2

w2 = w1;

When using w2=w1, it is using w2’s copy assignment operator on w1, which will create a temporary Wrapper w3 numbered as 3 using the copy constructor of w1. Then w1.detector==w3.detector. Then we swapped w2.detector and w3.detector. So detector no.2 will be destructed first due to the swap. Then detector no.3 is destructed and no.1 is the last one destructed.

5.

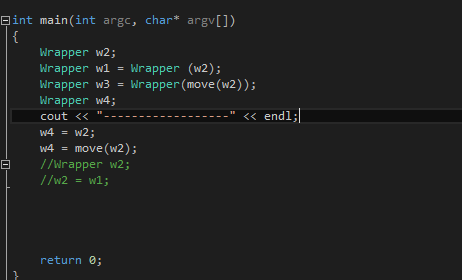


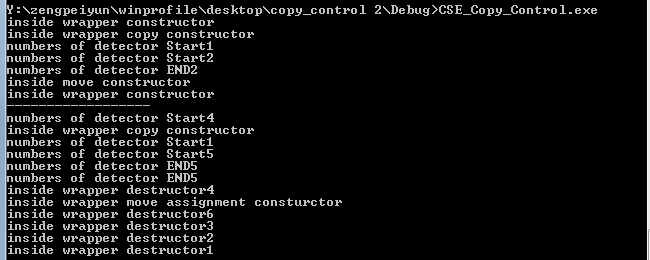


When constructing w1, we are using the copy constructor(as printed above)

When constructing w3, we are using the move constructor(as printed above). After constructing w3, p will be destroyed.

6.





same deal as the last question.