

X

swayam.gov.inhttps://swayam.gov.in/nc_details/NPTEL

navnathdeshmukh363@gmail.com ▾

[NPTEL \(https://swayam.gov.in/explorer?ncCode=NPTEL\)](https://swayam.gov.in/explorer?ncCode=NPTEL) » **Programming in Modern C++ (course)****Course outline****How does an NPTEL
online course work? ()****Week 0 ()****Week 1 ()****Week 2 ()****Week 3 ()**

W10_Programming_Qs-1

Due on 2023-04-06, 23:59 IST

Consider the following program (in C++11).

- Fill in the blank at LINE-1 with appropriate header for overloading copy assignment operator.
- Fill in the blank at LINE-2 with appropriate header and initialization list for move constructor.
- Fill in the blank at LINE-3 with appropriate header for overloading move assignment operator.

Your last recorded submission was on 2023-04-04, 20:39 ISTSelect the Language for this assignment. C++ ▾

```
1 #include <iostream>
2 #include <vector>
3
4 class item {
```

Week 4 ()**Week 5 ()****Week 6 ()****Week 7 ()****Week 8 ()****Week 9 ()****Week 10 ()**

- ☐ Lecture 46 : C++11 and beyond: General
Features: Part 1 (unit? unit=112&lesson=113)
- ☐ Lecture 47 : C++11 and beyond: General
Features: Part 2 (unit? unit=112&lesson=114)
- ☐ Lecture 48 : C++11 and beyond: General
Features: Part 3 (unit? unit=112&lesson=115)
- ☐ Lecture 49 : C++11 and beyond: General
Features: Part 4: Rvalue and Move/1 (unit? unit=112&lesson=116)
- ☐ Lecture 50 : C++11 and beyond: General

```

5     public:
6         item() : cp_(nullptr) { }
7         item(char c) : cp_(new char(c)) { }
8         item& operator=(const item& ob) {          // LINE-1: copy assignment
9             if (this != &ob) {
10                 delete cp_;
11                 cp_ = new char(*(ob.cp_) + 1);
12             }
13             return *this;
14         }
15         item(item&& ob):cp_(ob.cp_) {          // LINE-2: move constructor
16             ob.cp_ = nullptr; }
17         item& operator=(item&& ob) {          // LINE-3: move assignment
18             if (this != &ob) {
19                 cp_ = ob.cp_;
20                 ob.cp_ = nullptr;
21             }
22             return *this;
23     }
24     void print(){
25         if(cp_ == nullptr)
26             std::cout << "moved, ";
27         else
28             std::cout << *cp_ << ", ";
29     }
30     ~item() { delete cp_; }
31     private:
32         char* cp_ = nullptr;
33 };
34
35 int main(){
36     char a;
37     std::cin >> a;
38     item i1(a);
39     item i2;
40     i2 = i1;
41     i1.print();
42     i2.print();
43
44     item i3 = std::move(i1);
45     item i4;
46     i4 = std::move(i1);
47     i1.print();
48     i3.print();
49     i4.print();
50
51     return 0;
52 }

```

You may submit any number of times before the due date. The final submission will be considered for grading.

Features: Part 5: Rvalue
and Move/2 (unit?
unit=112&lesson=117)

☐ Tutorial 10 : How to
optimize C++11 programs
using Rvalue and Move
Semantics? (unit?
unit=112&lesson=118)

☒ Week 10 Lecture Material
(unit?
unit=112&lesson=119)

☒ Quiz: Week 10 :
Assignment 10
(assessment?name=208)

☒ **W10_Programming_Qs-
1**
(/noc23_cs50/progassignment?
name=209)

☐ W10_Programming_Qs-2
(/noc23_cs50/progassignment?
name=210)

☐ W10_Programming_Qs-3
(/noc23_cs50/progassignment?
name=211)

☐ Week 10 Feedback Form
(unit?
unit=112&lesson=120)

Week 11 ()

Download Videos ()

This assignment has Public Test cases. Please click on "Compile & Run" button to see the status of Public test cases. Assignment will be evaluated only after submitting using Submit button below. If you only save as or compile and run the Program , your assignment will not be graded and you will not see your score after the deadline.

Save as Draft

Compile & Run

Submit

Reset

Private Test cases used for Evaluation

Status

Test Case 1

Passed

Books ()**Transcripts ()****Problem Solving
Session ()**