

navnathdeshmukh363@gmail.com ~

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 ()

Week 4 ()

Week 5 ()

Week 6 ()

Week 7 ()

Week 8 ()

Week 9 ()

Lecture 41 : Input-Output: File Handling

W9_Programming_Qs-2

Due on 2023-03-30, 23:59 IST

Consider the following program (in C++11) that considers a number of purchases as input and find out the total expense. Fill in the blanks as per the instructions given below:

- Fill in the blank at LINE-1 with appropriate header to overload function operator.
- Fill in the blank at LINE-2 with an appropriate return statement to implement the overload function operator.
- Fill in the blank at LINE-3 with the appropriate parameter list for calling the function accumulate,

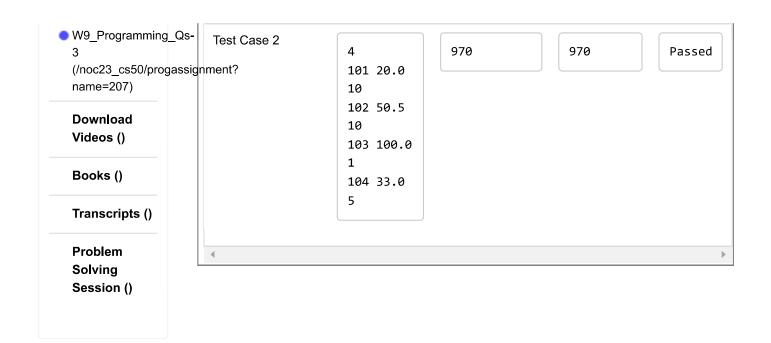
The program must satisfy the sample input and output.

Your last recorded submission was on 2023-03-25, 21:08 IST

Select the Language for this assignment. C++ ✓

```
1 #include <iostream>
   #include <list>
   #include <numeric>
 5
   class purchase{
 6
       public:
            purchase(int item_no, double price, int qty) : item_no_(item_no),
 8
                    price_(price), qty_(qty){}
 9
            double get_price(){ return price_; }
10
            int get_qty(){ return qty_; }
11
       private:
            int item_no_;
12
13
            double price_;
14
            int qty_;
15 | };
   struct total_expense{
   double operator() (double d, purchase& p){
16
                                                           //LINE-1
17
18
           return d + p.get_price()*p.get_qty();
                                                         //LINE-2
19
20
   };
21
   double get_total_expense(std::list<purchase> li, total_expense tc){
22
       double total = accumulate(li.begin(),li.end(),0.0,tc);
```

```
in C (unit?
                         24
                                  return total;
                         25 }
  unit=102&lesson=103)
Clecture 42:
  Input-Output:
  Streams in
  C++ (unit?
  unit=102&lesson=104)
Lecture 43 :
  C++ Standard
                            int main(){
   int n, a, b;
  Library: Part 1
  (Generic
                          1
                                  double c;
  Programming)
                          3
                                  std::list<purchase> li_pur;
  (unit?
                          4
                                  std::cin >> n;
                                  for(int i = 0; i < n; i++){</pre>
  unit=102&lesson=105)
                          5
6
7
8
                                       std::cin >> a >> c >> b;
                                                                      //read item_no, price and quantity
                                      purchase od(a, c, b);
Lecture 44 :
                                       li_pur.push_back(od);
  C++ Standard
                          9
  Library: Part 2
                         10
                                  total_expense tc;
                                  std::cout << get_total_expense(li_pur, tc);</pre>
                         11
  (STL) (unit?
                         12
                                  return 0;
  unit=102&lesson=106)
                         13
                       You may submit any number of times before the due date. The final submission will be
Lecture 45 :
                       considered for grading.
  C++ Standard
  Library: Part 3
                       This assignment has Public Test cases. Please click on "Compile & Run" button to see the
  (STL) (unit?
                       status of Public test cases. Assignment will be evaluated only after submitting using Submit
  unit=102&lesson=107) 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.
Tutorial 09 :
  How to design
                                      Save as Draft
                                                         Compile & Run
                                                                                 <u>S</u>ubmit
                                                                                                      <u>R</u>eset
  a UDT like
  built-in types?:
  Part 3:
  Updates and
                        Compilation: Passed
  Mixes of UDTs
                        Public Test Cases: 2 / 2 Passed
  (unit?
  unit=102&lesson=108)
                        Note: These tests may not be considered while scoring. Know more.
Week 9
  Lecture
                          Public Test Cases
                                                                 Expected Output
                                               Input
                                                                                     Actual Output
                                                                                                      Status
  Material (unit?
  unit=102&lesson=109)
                          Test Case 1
                                                                  702.5
                                                                                       702.5
                                                                                                       Passed
 Quiz: Week 9 :
                                                 101 50.0
  Assignment 9
                                                 4
  (assessment?
                                                 102 100.5
  name=202)
 W9_Programming_Qs-
  (/noc23_cs50/progassignment?
  name=204)
W9_Programming_Qs-
  (/noc23_cs50/progassignment?
  name=205)
```



→

