

↑ 6.8 Vector back() and pop\_back()



Students:

Section 6.9 is a part of 2 assignments: CSC108 CH06.1-6.9 C6A ▾

Includes: CA

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

Due: 04/29/2025, 11:59 PM EDT

## 6.9 Using a loop to modify, copy, or compare vectors

### Modifying vector elements

A program may need to modify elements while iterating through a vector. The program below uses a loop to convert any negative vector element value to 0.

Figure 6.9.1: Modifying a vector during iteration example: Converting negatives to 0.

```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    const int NUM_ELEMENTS = 5; // Number of elements to add
    vector<int> userVals; // Empty vector for user values
    unsigned int i; // Loop index
    int userInput; // Element value

    cout << "Enter " << NUM_ELEMENTS << " integer values..." << endl;
    for (i = 0; i < NUM_ELEMENTS; ++i) {
        cout << "Enter value: ";
        cin >> userInput;
        userVals.push_back(userInput);
    }

    for (i = 0; i < userVals.size(); ++i) {
        if (userVals.at(i) < 0) {
            userVals.at(i) = 0;
        }
    }

    cout << "Updated vector values:";
    for (i = 0; i < userVals.size(); ++i) {
        cout << " " << userVals.at(i);
    }
    cout << endl;
    return 0;
}
```

Feedback?

**PARTICIPATION ACTIVITY**

## 6.9.1: Modifying a vector in a loop.

What are the resulting vector elements, assuming each question starts with a vector of size 4 and elements -55, -1, 0, 9?

- 1) for (i = 0; i < 4; ++i) {  
    itemsList.at(i) = i;  
}  
  - 54, 0, 1, 10
  - 0, 1, 2, 3
  - 1, 2, 3, 4
- 2) for (i = 0; i < 4; ++i) {  
    if (itemsList.at(i) < 0) {  
        itemsList.at(i) = itemsList.at(i) \* -1;  
    }  
}  
  - 55, -1, 0, -9
  - 55, 1, 0, -9
  - 55, 1, 0, 9
- 3) for (i = 0; i < 4; ++i) {  
    itemsList.at(i) = itemsList.at(i+1);  
}  
  - 1, 0, 9, 0
  - 0, -55, -1, 0
  - Error (program aborts)
- 4) for (i = 0; i < 3; ++i) {  
    itemsList.at(i) = itemsList.at(i+1);  
}  
  - 1, 0, 9, 9
  - 1, 0, 9, 0
  - Error (program aborts)
- 5) for (i = 0; i < 3; ++i) {  
    itemsList.at(i+1) = itemsList.at(i);  
}  
  - 55, -55, -55
  - 55, -55, -1, 0
  - Error (program aborts)

Feedback?

zyDE 6.9.1: Modifying a vector during iteration: Doubling element values.

Complete the following program to double each number in the vector.

[Load default template...](#)

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     const int NUM_ELEMENTS = 5; // Number of elements to add
7     vector<int> userVals; // Empty vector for user values
8     unsigned int i; // Loop index
9     int userInput; // Element value
10
11    cout << "Enter " << NUM_ELEMENTS << " integer values..." << endl;
12    for (i = 0; i < NUM_ELEMENTS; ++i) {
13        cout << "Enter value: " << endl;
14        cin >> userInput;
15        userVals.push_back(userInput);
16    }
```

67-5-994 22

Run

Feedback?

**Element by element vector copy**

In C++, the = operator conveniently performs an element-by-element copy of a vector, called a **vector copy operation**. The operation vectorB = vectorA resizes vectorB to vectorA's size, appending or deleting elements as needed. vectorB commonly has a size of 0 before the operation.

Figure 6.9.2: Using = to copy a vector: Original and sale prices.

```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> origPrices; // Source vector
    vector<int> salePrices; // Target vector
    unsigned int i; // Loop index

    // Add prices to original prices vector
    origPrices.push_back(10);
    origPrices.push_back(20);
    origPrices.push_back(30);
    origPrices.push_back(40);

    // Copy original prices to sales prices
    salePrices = origPrices;

    // Update sale prices. This does not affect original prices.
    salePrices.at(2) = 27;
    salePrices.at(3) = 35;

    // Output original and sale prices
    cout << "Original prices: ";
    for (i = 0; i < origPrices.size(); ++i) {
        cout << " " << origPrices.at(i);
    }
    cout << endl;

    cout << "Sale prices: ";
    for (i = 0; i < salePrices.size(); ++i) {
        cout << " " << salePrices.at(i);
    }
    cout << endl;

    return 0;
}
```

Original prices: 10 20 30 40

Sale prices: 10 20 27 35

Feedback?

**PARTICIPATION ACTIVITY**

## 6.9.2: Vector copy operation.

Assume vectors have been declared as follows and have been initialized as indicated in the comments:

```
vector<int> userVals(4); // {44, 55, 66, 77}
vector<int> newVals; // No elements yet
```

- 1) What is newVals after: newVals = userVals;  
Type answer as: 10, 20, 30, 40  
If appropriate type: Error

[Check](#) [Show answer](#)

- 2) What is newVals after:  
newVals = userVals;  
userVals.at(0) = 33;  
Type answer as: 10, 20, 30, 40  
If appropriate type: Error

[Check](#) [Show answer](#)

- 3) Given: vector<int> otherVals(9).  
What size is newVals after:  
newVals = userVals;  
...  
newVals = otherVals;  
If appropriate type: Error

[Check](#) [Show answer](#)

Feedback?

**PARTICIPATION ACTIVITY**

## 6.9.3: Vector comparing.

Assume vectors have been declared as follows and have been initialized as indicated in the comments:

```
vector<int> vectorX(2); // {3, 4}
vector<int> vectorY(5); // {3, 4, 0, 7, 8}
vector<int> vectorZ(5); // {3, 4, 0, 6, 8}
```

- 1) (vectorX == vectorY) will evaluate to:

- True
- False

- 2) Given: vectorX = vectorY; (vectorX == vectorY) will evaluate to:

- True
- False

- 3) (vectorZ == vectorY) will evaluate to:

- True
- False

- 4) (vectorZ.size() == vectorY.size()) will evaluate to:

- True
- False

Feedback?

**CHALLENGE ACTIVITY**

## 6.9.1: Loop-modifying or copying/comparing vectors.

620890.5010016.0x32q7

[Start](#)

Integer numElements is read from input, representing the number of integers to be read next. Then, the remaining integers are read and stored into vector numbersList. For each element in numbersList that is divisible by 9:

- Output the element, followed by ":" corrected to a number not divisible by 9" and a newline.
- Assign the element with the element's current value minus 1.

[Click here for example](#) ▾

Note: (x % 9 == 0) returns true if x is divisible by 9.

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     int numElements;
7     vector<int> numbersList;
8     unsigned int i;
9     int inputVal;
10
11    cin >> numElements;
12
13    for (i = 0; i < numElements; ++i) {
14        cin >> inputVal;
15        numbersList.push_back(inputVal);
16    }
17
18    cout << "Original numbers: ";
```

1

2

3

[Check](#)[Next level](#)

Feedback?

How was this section?



Provide section feedback

**Activity summary for assignment: CSC108 CH06.1-6.9 C6A ▾**

Due: 04/29/2025, 11:59 PM EDT

0 / 40 points

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)**Completion details** ▾

↓ 6.10 Swapping two variables (general)