6.2: Array/vector concept (general)

Students:
Section 6.3 is a part of 2 assignments: CSC108 CH06.1-6.9 O6A
Please browse to this assignment through BlackboardLearn so zybooks knows where to send your activity. Learn more
Includes: CA
Due: 04/29/2025, 11:59 PM EDT

6.3 Vectors

Vector declaration and element access

A programmer commonly needs to maintain a list of items. Just as people often maintain lists of items like a grocery list or a course roster. A **vector** is an ordered list of items of a given data type. Each item in a vector is called an **element**. A programmer must include the statement `#include <vector>` at the top of the file to use vectors.

Construct 6.3.1: Declaring an empty vector.

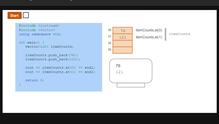
```
vector<datatype> vectorname;
```

Feedback?

The statement above declares an empty vector capable of storing an unspecified number of elements. The type of each vector element is specified within the angle brackets (<>).

push_back() appends a new element to the end of an existing vector, **at()** accesses a vector element at the index specified within the parenthesis.

PARTICIPATION ACTIVITY 6.3.1: Declaring vectors and adding elements.



Captions

Feedback?

Curly braces and angle brackets

Terminology note: {) are **braces**, < > are **angle brackets**, or **chevrons**.

PARTICIPATION ACTIVITY 6.3.2: Declaring and accessing vector elements.

Use the following code to answer the questions:

```
vector<int> yearsList;
yearsList.push_back(1999);
yearsList.push_back(2020);
yearsList.push_back(2025);
```

1) How many elements does the vector declaration create?
- 0
- 1
- 3

2) What value is stored at yearsList.at(1)?
- 0
- 1999
- 2020

3) What value does `currYear = yearsList.at(2)` assign currYear?
- 0
- 2025
- Invalid index

4) `biggerYear = yearsList.at(3)` is valid assignment?
- Yes, the third element is accessed.
- No, yearsList.at(3) doesn't exist.

5) What is the proper way to access the first element in yearsList?
- yearsList.at(0)
- yearsList.at(1)

6) What are the contents of the vector if the following statement is placed at the end of the above code: `yearsList.at(0) = yearsList.at(2)`?
- 1999, 2020, 1999
- 2020, 2020, 2025
- 2025, 2020, 2025

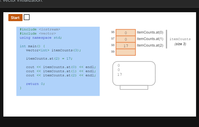
Feedback?

Vector initialization

A vector can be declared with an initial number of elements by specifying the number of elements within parentheses following the vector name. Ex: `vector<int> gameScores(6);` declares a vector gameScores with 4 integer elements.

A vector can also be initialized by specifying the initial values in the list, separated by commas. Ex: `vector<int> carSales = {5, 7, 11};` creates a vector of three integer elements initialized with values 5, 7, and 11. Such vector declaration and initialization does not require specifying the vector size, because the vector's size is automatically set to the number of elements within the braces.

PARTICIPATION ACTIVITY 6.3.3: Vector initialization.



Captions

Feedback?

PARTICIPATION ACTIVITY 6.3.4: Vector initialization.

1) Write a single statement to declare a vector of ints named salesGoals with 4 elements.

Check Show answer

2) Given the following, what is maxScores.at(3)?

```
vector<int> maxScores = {20, 20, 100, 50};
```

Check Show answer

Feedback?

Common error: Forgetting to include <vector>

A common error is to forget the #include <vector> at the top of the file when using vectors. Trying to then declare a vector variable may yield a strange compiler error message, such as:

```
(main)/a.out(main.o): unknown type name 'vector'
(main)/a.out(main.o): error: expected ';' before 'i'
```

The same error message may be seen if the vector library is included but the namespace std is not used.

Using an expression for a vector index

A powerful aspect of vectors is that the index is an expression. Ex: userNums.at(i) uses the value held in the int variable i as the index. As such, a vector is useful to easily lookup the Nth item in a list.

A vector's index must be an unsigned integer type. The vector index cannot be a floating point type, even if the value is 0.0, 1.0, etc.

The program below allows a user to print the age of the Nth oldest known person to have ever lived. The program quickly accesses the Nth oldest person's age using `oldestPeople.at(inthreshon - 1)`. Note that the index is `inthreshon - 1` rather than just `inthreshon` because a vector's indices start at 0, so the 1st age is at index 0, the 2nd at index 1, etc.

Figure 6.3.1: Vector's ith element can be directly accessed using .at() - 1): Oldest people program.

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

int main() {
   vector<int> oldestPeople(5); // Vector of 5 ints
   int inthreshon;                // User input, Nth oldest person

   oldestPeople.at(0) = 122; // Died 1997 in France
   oldestPeople.at(1) = 119; // Died 1934 in U.S.
   oldestPeople.at(2) = 115; // Died 1992 in U.S.
   oldestPeople.at(3) = 115; // Died 2004 in Japan
   oldestPeople.at(4) = 114; // Died 2005 in Brunei

   cout << "Enter N (1..5): ";
   cin >> inthreshon;

   if (inthreshon >= 1 && inthreshon <= 5) {
      cout << "The #" << inthreshon << " oldest person lived ";
      cout << oldestPeople.at(inthreshon - 1) << " years." << endl;
   }

   return 0;
}
```

```
Enter N (1..5): 1
The #1 oldest person lived 122 years.

...

Enter N (1..5): 4
The #4 oldest person lived 115 years.

...

Enter N (1..5): 5
The #5 oldest person lived 114 years.

...

Enter N (1..5): 0

...

Enter N (1..5): 3
The #3 oldest person lived 115 years.
```

Feedback?

PARTICIPATION ACTIVITY 6.3.5: Nth oldest person program.

1) In the program above, what is the purpose of this check:
```
if (inthreshon >= 1 && inthreshon <= 5) {
```

- To avoid overflow because nthPerson's data type can only store values from 1 to 5.
- To ensure only valid vector elements are accessed because the vector oldestPeople only has 5 elements.

Feedback?

PARTICIPATION ACTIVITY 6.3.6: Vector declaration and accesses.

1) Declare a vector named myVals that stores 10 items of type int.

Check Show answer

2) Assign x with the value stored at index 8 of vector myVals.

Check Show answer

3) Given myVals has 10 elements, assign the last element in myVals with the value 555.

Check Show answer

4) Assign the myVals element at the index held in currIndex with the value 777.

Check Show answer

5) Assign temp1st with the myVals element at the index one after the value held in variable i.

Check Show answer

Feedback?

Loops and vectors

A key advantage of vectors becomes evident when used in conjunction with loops. The program below uses a loop to allow a user to enter 8 integer values, storing those values in a vector, and then printing those 8 values.

A vector's **size()** function returns the number of vector elements. Ex: In the program below, userVals.size() is 8 because the vector was declared with 8 elements.

Figure 6.3.2: Vectors combined with loops are powerful together: User-entered numbers.

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

int main() {
   const int NUM_VALS = 8;      // Number of elements in vector
   vector<int> userVals(NUM_VALS);// Vector values
   unsigned int i;              // Loop index

   cout << "Enter " << NUM_VALS << " integer values..." << endl;
   for (i = 0; i < userVals.size(); ++i) {
      cout << "Value #" << (i + 1) << ": ";
      cin >> userVals.at(i);
   }

   cout << "You entered: ";
   for (i = 0; i < userVals.size(); ++i) {
      cout << userVals.at(i) << " ";
   }
   cout << endl;

   return 0;
}
```

```
Enter 8 integer values...
Value #1: 99
Value #2: 77
Value #3: 44
Value #4: 1
Value #5: 555555
Value #6: 0
Value #7: 2
You entered: 5 99 -1 -44 8 555555 0 2
```

Feedback?

PARTICIPATION ACTIVITY 6.3.7: Vector with loops.

Refer to the program above.

1) How many times does each for loop iterate?
- 8
- 9
- Unknown

2) Which one line of code can be changed to allow the user to enter 100 elements?
- `const int NUM_VALS = 9;`
- `for (i = 0; i < userVals.size(); ++i) {`

Feedback?

CHALLENGE ACTIVITY 6.3.1: Enter the output for the vector.

Lesson 6.3 Vectors (zyLab)

Start

Type the program's output

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

int main() {
   vector<int> vecThree;
   unsigned int i;

   vecThree.push_back(99);
   vecThree.push_back(88);
   vecThree.push_back(77);

   for (i = 0; i < vecThree.size(); ++i) {
      cout << vecThree.at(i) << endl;
   }

   return 12;
}
```

6
8
0

1 2 3 4 5

Check Next

Feedback?

CHALLENGE ACTIVITY 6.3.2: Vectors.

Lesson 6.3 Vectors (zyLab)

Start

Two integers are read from input as variables student1 and student2. Use push_back() to add the variables student1 and student2 to vector studentLogs with the variables student1 and student2 in the order the input integers are read.

Ex: If the input is `32 23`, then the output is:

```
32 20
```

```
#include "codetest.h"
#include <iostream>
#include <vector>
using namespace std;

int main() {
   int i;
   vector<int> studentLogs;
   int student1;
   int student2;
   unsigned int i;

   cin >> student1;
   cin >> student2;

   /* Your code goes here */

   for (i = 0; i < studentLogs.size(); ++i) {
      cout << studentLogs.at(i) << " ";
   }
```

1 2 3 4 5

Check Next level

Feedback?

How was this section? Provide section feedback

Activity summary for assignment: CSC108 CH06.1-6.9 O6A
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