5.6 Vector resize

Commonly, the size of a list of items is not known during a program's compile time. Thus, a vector's size need not be specified in the vector's declaration. Instead, a vector's size can be set or changed while a program executes using ***resize(N)***. Ex: `highScore.resize(10)` resizes the highScores vector to have 10 elements.

resize() can be called multiple times. If the new size is larger, resize() adds elements at the end. If smaller, resize() deletes elements from the end. If userScores has size 3 (elements 0, 1, 2), `userScores.resize(2);` would delete element 2, leaving elements 0 and 1. A subsequent access to userScores.at(2) would result in an error.

PARTICIPATION ACTIVITY 5.6.1: Vector resize.

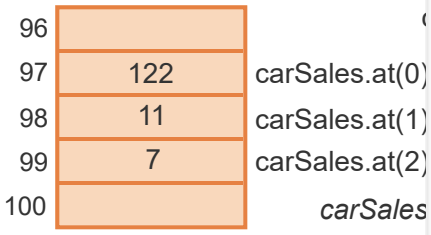
1 2 3 2x speed

```
vector<int> carSales;

// carSales.at(0) = 122; Would be out-of-range error

carSales.resize(3);

carSales.at(0) = 122;
carSales.at(1) = 11;
carSales.at(2) = 7;
```



carSales.resize(3) allocates 3 elements with indices 0, 1, and 2.
Elements can now be accessed using the .at() function.

Feedback?

The program below asks a user to indicate the number of values the user will enter, allocates that number of elements for a vector, assigns the vector's elements with user-entered values, and then displays the vector's elements.

Figure 5.6.1: Resizing a vector based on user input.

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

int main() {
   vector<int> userVals; // No elements yet
   int numVals;
   unsigned int i;

   cout << "Enter number of integer values: ";
   cin >> numVals;

   userVals.resize(numVals); // Allocate elements

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

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

   return 0;
}
```

```
Enter number of integer values:
7
Enter 7 integer values...
Value: -5
Value: -99
Value: 0
Value: 13
Value: 7
Value: -22
Value: 1
You entered: -5 -99 0 13 7 -22 1
```

Feedback?

PARTICIPATION ACTIVITY 5.6.2: Vector resize and size functions.

Given the vector declaration:

```
vector<int> agesVctr;
```

1) Immediately after the declaration, agesVctr has only 1 element.

   - True
   - False

Correct

The vector has no elements. agesVctr.at(0) would yield an error.

2) agesVctr.size(4) allocates 4 elements for agesVctr.

   - True
   - False

Correct

agesVctr.resize(4) allocates elements. agesVctr.size() returns the current size of vector agesVctr.

3) Given agesVctr has 3 elements,

Correct

agesVctr.resize(4) adds 4 more elements, totalling 7 elements.

- ○ True
- ◉ False

agesVctr.resize(4) adds just 1 more element, to bring the total to 4.

4) Given agesVctr has 3 elements with values 22, 18, and 19, agesVctr.resize(2) changes agesVctr to have 2 elements with values 22 and 18.

- ◉ True
- ○ False

Correct

When the new size is smaller than the existing size, resize removes elements from the end.

5) After agesVctr.resize(5) and agesVctr.at(0) = 99, agesVctr.size() evaluates to 1.

- ○ True
- ◉ False

Correct

agesVctr.size() evaluates to the number of allocated elements, regardless of whether those elements have been assigned with values yet. agesVctr.size() would return 5.

Feedback?

CHALLENGE ACTIVITY 5.6.1: Determining the size of a vector.

Assign currentSize with the size of the sensorReadings vector.

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

int main() {
   vector<int> sensorReadings(4);
   int currentSize;
   int input;
   cin >> input;

   sensorReadings.resize(input);

   /* Your solution goes here  */
   currentSize = sensorReadings.size();

   cout << "Number of elements: " << currentSize << endl;

```

```
   return 0;
}
```

Run ✓ All tests passed

✓ Testing with input: 10

Your output `Number of elements: 10`

✓ Testing with input: 2

Your output `Number of elements: 2`

Feedback?

CHALLENGE ACTIVITY 5.6.2: Resizing a vector.

Resize vector countDown to have newSize elements. Populate the vector with integers {newSize, newSize - 1, ..., 1}. Ex: If newSize = 3, then countDown = {3, 2, 1}, and the sample program outputs:

```
3 2 1 Go!
```

```
int main() {
   vector<int> countDown(0);
   int newSize;
   unsigned int i;

   cin >> newSize;

   /* Your solution goes here  */
   countDown.resize(newSize);
   for (i = 0; i < countDown.size(); ++i) {
      countDown.at(i) = newSize;
      newSize--;
   }

   for (i = 0; i < countDown.size(); ++i) {
      cout << countDown.at(i) << " ";
   }
   cout << "Go!" << endl;

   return 0;
}
```

Run ✓ All tests passed

✓ Testing size with input: 3

Your value `3`

✓ Testing elements with input: 3

Your output `3 2 1 Go!`

✓ Testing size with input: 5

Your value `5`

✓ Testing elements with input: 5

Your output `5 4 3 2 1 Go!`

✓ Testing size with input: 1

Your value `1`

✓ Testing elements with input: 1

Your output `1 Go!`

Feedback?