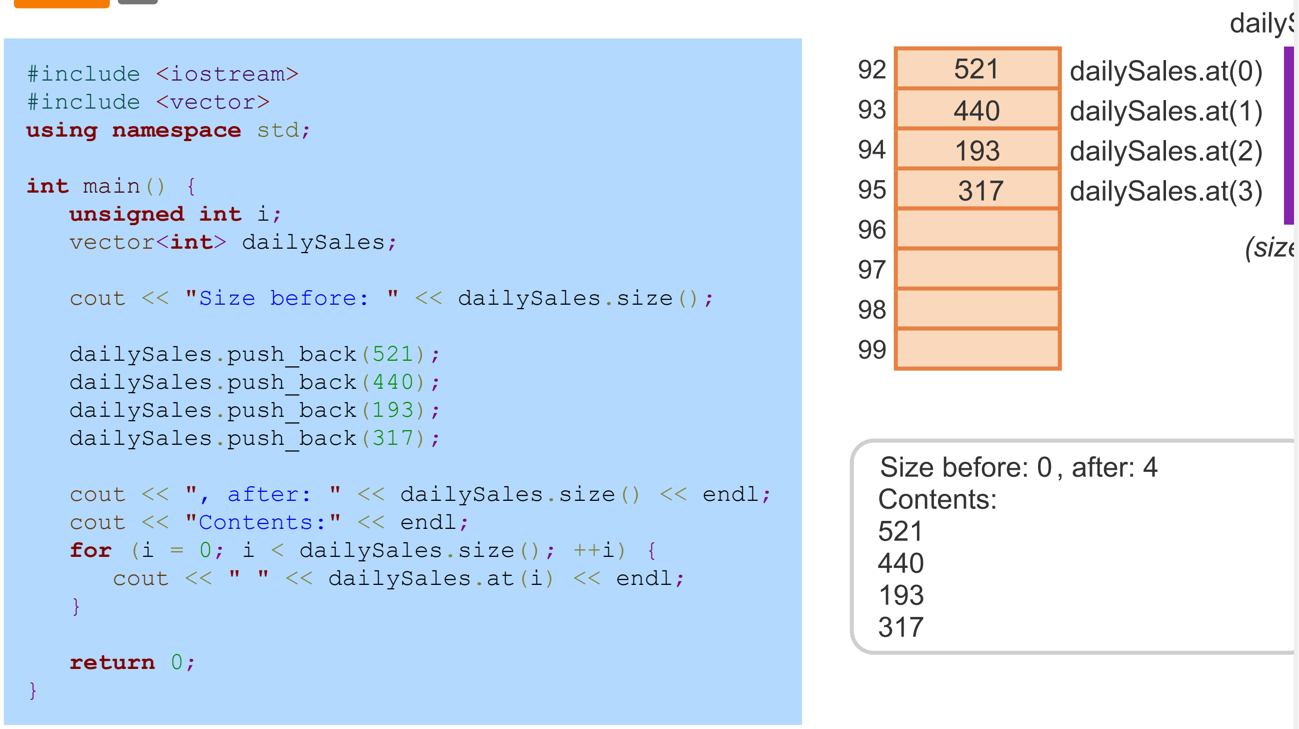
5.7 Vector push_back

Appending items to a vector

A programmer can append a new element to the end of an existing vector using a vector's ***push_back***. Ex: `dailySales.push_back(521)` creates a new element at the end of the vector dailySales and assigns that element with the value 521.

PARTICIPATION ACTIVITY | 5.7.1: The vector push_back() function.

Start 2x speed



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

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

   cout << "Size before: " << dailySales.size();

   dailySales.push_back(521);
   dailySales.push_back(440);
   dailySales.push_back(193);
   dailySales.push_back(317);

   cout << ", after: " << dailySales.size() << endl;
   cout << "Contents:" << endl;
   for (i = 0; i < dailySales.size(); ++i) {
      cout << " " << dailySales.at(i) << endl;
   }

   return 0;
}
```

```
Size before: 0 , after: 4
Contents:
521
440
193
317
```

Feedback?

PARTICIPATION ACTIVITY | 5.7.2: Vector push_back().

1) If vector `itemPrices` has two elements with values 45, 48, what

does itemPrices.size() return?

Check Show answer

2) If itemPrices has element values 45, 48, then after itemPrices.push_back(38), what are itemPrices' element values? Type answer as: 50, 60, 70

Check Show answer

Feedback?

Vector pop_back() and back()

The following table summarizes a few common functions dealing with the back (or last element) of a vector.

Table 5.7.1: Functions on the back of a vector.

push_back()	`void push_back(const int newVal);`
	
	Append new element having value newVal.
		`// playersList initially 55, 99, 44 (size is 3)`
		`playersList.push_back(77);`
		`// Appends new element 77`
		`// playersList is now 55, 99, 44, 77 (size is 4)`

back()	`int back();`
	
	Returns vector's last element. Vector is unchanged.
		`// playersList initially 55, 99, 44`
		`cout << playersList.back(); // Prints 44`
		`// playersList is still 55, 99, 44`

pop_back()	`void pop_back();` Removes the last element.	

		`// playersList is 55, 99, 44 (size 3)`
		`playersList.pop_back(); // Removes last element`
		`// playersList now 55, 99 (size 2)`
		
		`cout << playersList.back(); // Common combination of back()`
		`playersList.pop_back(); // followed by pop_back()`
		`// Prints 99. playersList becomes just 55`
		
		`cout << playersList.pop_back(); // Common error:`
		
		`// pop_back() returns void`

Shown for vector<int>, but applies to other types.

Feedback?

The program below declares a vector groceryList, which is initially empty. As the user enters grocery items one at a time, the program uses push_back() to append the items to the list. When done, the user can go shopping, and is presented one list item at a time (which the user presumably finds and places in a shopping cart). The program uses back() to get each item from the list and pop_back() to remove the item from the list. When the list is empty, shopping is finished.

Note that because the program removes items from the end of the list, the items are presented in reverse order.

Figure 5.7.1: Using push_back(), back(), and pop_back(): A grocery list example.

```
Enter grocery items or type
done.
Oranges
Apples
Bread
Juice
done

Enter any key for next item.
Juice   a
Bread   a
Apples   a
Oranges   a

Done shopping.
```

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

int main() {
   vector<string> groceryList; // Vector storing shopping
list
   string groceryItem;         // Individual grocery items
   string userCmd;             // User input

   // Prompt user to populate shopping list
   cout << "Enter grocery items or type done." << endl;
   cin >> groceryItem;
   while (groceryItem != "done") {
      groceryList.push_back(groceryItem);
      cin >> groceryItem;
   }

   // Display shopping list
   cout << endl << "Enter any key for next item." << endl;
   while (groceryList.size() > 0) {
      groceryItem = groceryList.back();
      groceryList.pop_back();
      cout << groceryItem << "   ";
      cin >> userCmd;
   }
   cout << endl << "Done shopping." << endl;

   return 0;
}
```

Feedback?

PARTICIPATION ACTIVITY 5.7.3: Vector back() and pop_back() functions.

1) If itemPrices has elements 45, 22, 38, what does price = itemPrices.pop_back() assign to price? Type Error if appropriate.

Check Show answer

2) If itemPrices has elements 45, 22, 38, what does price = itemPrices.back() assign to price? Type Error if appropriate.

Check Show answer

3) If itemPrices has elements 45, 22, 38, what is the vector after itemPrices.back() is called? Type answer as: 50, 60, 70

Check Show answer

4) If itemPrices has elements 45, 22, 38, then after itemPrices.pop_back(), what does itemPrices.at(2) return? Type Error if appropriate.

Check Show answer

Feedback?

CHALLENGE ACTIVITY 5.7.1: Appending a new element to a vector.

Append newValue to the end of vector tempReadings. Ex: If newValue = 67, then tempReadings = {53, 57, 60} becomes {53, 57, 60, 67}.

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

int main() {
   const int NUM_ELEMENTS = 3;
   vector<int> tempReadings(NUM_ELEMENTS);
   int newValue;
   unsigned int i;

   for (i = 0; i < tempReadings.size(); ++i) {
      cin >> tempReadings.at(i);
   }

   cin >> newValue;

   /* Your solution goes here  */
   tempReadings.push_back(newValue);

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

Run ✓ All tests passed

✓ Testing vector size with inputs: 53 57 60 67

Your value `4`

✓ Testing vector elements with inputs: 53 57 60 67

Your output

```
53 57 60 67
```

✓ Testing vector size with inputs: -43 -30 -35 -31

Your value `4`

✓ Testing vector elements with inputs: -43 -30 -35 -31

Your output

```
-43 -30 -35 -31
```

Feedback?

CHALLENGE ACTIVITY 5.7.2: Removing an element from the end of a vector.

Remove the last element from vector ticketList.

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

int main() {
   const int NUM_ELEMENTS = 3;
   vector<int> ticketList(NUM_ELEMENTS);
   unsigned int i;

   for (i = 0; i < ticketList.size(); ++i) {
      cin >> ticketList.at(i);
   }

   /* Your solution goes here  */
   ticketList.pop_back();

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

Run ✓ All tests passed

✓ Testing size with inputs: 5 100 12

Your value
```
2
```

✓ Testing elements with inputs: 5 100 12

Your output
```
5 100
```

✓ Testing size with inputs: 4 -4 3

Your value
```
2
```

✓ Testing elements with inputs: 4 -4 3

Your output
```
4 -4
```

Feedback?

CHALLENGE ACTIVITY 5.7.3: Reading the vector's last element.

Write a statement to print "Last mpg reading: " followed by the value of mpgTracker's last element. End with newline. Ex: If mpgTracker = {17, 19, 20}, print:

```
Last mpg reading: 20
```

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

int main() {
   const int NUM_ELEMENTS = 3;
   vector<int> mpgTracker(NUM_ELEMENTS);
   int i;

   for (i = 0; i < mpgTracker.size(); ++i) {
      cin >> mpgTracker.at(i);
   }

   /* Your solution goes here  */
```

```
    cout << "Last mpg reading: "<< mpgTracker.back()<< endl;

    return 0;
}
```

Run ✓ All tests passed

✓ Testing with inputs: 17 19 20

Your output
```
Last mpg reading: 20
```

✓ Testing with inputs: 30 20 40

Your output
```
Last mpg reading: 40
```

✓ Testing with inputs: 14 20 16

Your output
```
Last mpg reading: 16
```

Feedback?