Data Structures with C++ : CS189

Lecture 4-1: Vectors

Recap

- An ADT is a container for actual data, but is not data itself
  - Vector **of** ints. List **of** floats
- STL is a library that has perfect implementations of each one
- This is 189, so yes you will have to write parts of each
  - Even though in real life you won't
- Iterators and Functors are the hardest part because they are 100% new topics

Information Hiding

- The practice of letting people use your class without knowing how it works
- We will look inside, and cplusplus.com will tell you details, but in real life you won't entirely care
- Last week's survey of STL was super high level. This week's details will be low level

Vector<>

- A container that holds any number of things and lets you access each immediately
  - "Immediately" is tomorrow's "Big O" topic.
- To be called a vector (in case you make your own), you not only have to have certain methods, you have to make them a certain speed
  - "Speed" is tomorrow as well. Easier to talk about Big O if we have an example like vector from today

Vector Methods

- Real vectors have 30ish methods. Here are the most common that many adts share
  - push_back: add data to the end
  - pop_back: remove data from the end
  - size: get number of elements
  - resize: force number of elements
  - clear: remove all elements
- Some are vector-specific
  - at (or []): get specific element
  - reserve: allocate memory
  - capacity: how much memory do you have

My Vector.h

- For each adt, I'll be providing some form of assist in the Files section
  - Making you write every one from scratch is cruel
- You must always start your homework with my file if I give one
- The reason this class is completely in an h file is because of templates
  - Copy paste find replace
- If there were a vector.cpp, main wouldn't know about it and would get an empty vector header

Vector Internals

- Inside, a vector is an array
  - Every time the array fills up, a new bigger one is made and the data is copied over
  - The user never sees this
- Since it is an array, lookups are instant
  - Just index
- And remember the word "of" when thinking about how it is used
  - vector of ints
  - vector of Students
  - vector of vectors of maps of strings to ints
    - Templates!

Template's one Trick

- You can see the capital T in the vector file that is the placeholder for the find-replace
- You make a vector of something, and you assume everything worked since it compiled
- BUT. If there is a template method that nobody ever calls, it gets deleted during copy-paste
  - EVEN IF IT DOESN'T COMPILE

Reinforce the Template Trick

1. You do your homework
   a. It compiles
2. You don't test your code with a good main
   a. Say you never test "Size"
3. You turn it in
4. I add your homework to my tester
   a. Mine DOES test everything, including "Size"
5. If Size doesn't compile, you never noticed, but I will
6. Homework that doesn't compile is a zero
   a. It's not even code, it's text

Array Recap

- A normal array has to know its size because it is in the stack and the exe needs to know
  - int pants[10];
- A dynamically allocated array gets free memory in the heap. Only the pointer itself is a variable in the stack
  - int *shirts = new int[10];
  - delete[] shirts;
- Vector's inner array will change size all the time so please note that syntax

Reserve and Capacity

- Resize and Size are referring to data
- Reserve and Capacity are memory
- The code for Reserve is the hardest this week:
  - Start with an arbitrary capacity
  - When user adds data and we are "full" because size and capacity are equal
    - first we need to allocate a bigger array
    - and copy the old data over
    - then delete the old array
  - Either way, we can put the data in the next spot and increase size by one

Big 3/5/7

```
// Default
vector<int>X;

// Copy
vector<int> Y(X);

// Assign
X.push_back(1);
Y = X;
// Initialize has an =, but optimizes to copy
vector<int> Z = X;//Trick!
```

- We will concentrate on the Big 3 in 189
  - More recent versions of C++ have more than 3
    - We'll talk about C++ versions week 15ish
- If you ever write a class that uses "new", you MUST write these three:
  - Copy Constructor
  - Assignment operator
  - Destructor
- A straight copy would give another vector a pointer to our inner array
  - Many different crashes could result
- Not having destructor would leak our array

Go to Canvas Quiz

Don't run off until we make sure everyone is okay

End

Yes Vector is built in. Yes you still have to write it. Welcome to 189.