Lab 1: Vectors Part 1

CSCI 41

Objectives:

- Understand the vector ADT by implementing our very own version of it
- Practice working with the heap
- Practice writing tests
- Get used to using Makefiles

Overview

Ever wonder what goes on inside the std::vector class? Well wonder no longer! You'll be making your own Vec class that holds ints. If the heap is your enemy right now, hopefully by the end of this lab it will be your friend ❤️.

Part 1: Get the starter code

First, `cd` into the folder where you want to store your class files (e.g., `~/labs`). Copy the starter code there with the following line:

```
gimme lab01@csci41
```

This will make a lab01 folder in whatever directory you were in.

Part 2: Implement the Vec class

Open up vec.h to see the bigger picture, and then open up vec.cpp and start working on the method implementations.

For the constructor, the Vec should start out empty (numItems = 0) but able to hold one element (capacity = 1)—make sure you understand the subtle difference between those variables.

Part 3: Test your Vec class

Look at testVec.cpp. It includes vec.h (and `testing.h`) and tests that the `Vec` class was implemented correctly—these are the benchmarks that you will be graded against. Compile it with `make testVec`. Make sure you understand what's going on inside the `Makefile`.

In addition to **getting the current tests to pass**, your job is to **write 4 more tests** in the same style as the ones given.

Part 4: Submit your code

When you're satisfied with your solutions, you can submit them to the autograder for grading. **Do not submit and assume you got 100%—you may be unpleasantly surprised. Always check your grade.**

From your solution directory, submit your code to the autograder using the following command on the terminal:

```
turnin lab01@csci41 vec.cpp testVec.cpp
```

The autograder will grade your code within a minute or so. If it's not working, please yell at Lawton to fix it. Run `view-grades` to look at your grades on the terminal, or go to the class website to view them there. You may resubmit as much as you want before the due date—just follow this same process after you've updated your programs.

Rubric

Rubric Item	**Points (24 pts total)**
testVec.cpp runs and passes the tests that I've provided	16 pts (4 for each test)
testVec.cpp runs and passes the 4 additional tests that you wrote	8 pts (2 for each test)