

# CS 309

## Autonomous Intelligent Robotics (FRI I)

### HW 1: C++ Basics Due: February 12, 2019

You can find the latest version of this PDF at  
[http://justinhart.net/files/homework/2019\\_spring\\_cs309/HW1/hw1.pdf](http://justinhart.net/files/homework/2019_spring_cs309/HW1/hw1.pdf)

You can find an abstract class in  
[http://justinhart.net/files/homework/2019\\_spring\\_cs309/HW1/NumberList.h](http://justinhart.net/files/homework/2019_spring_cs309/HW1/NumberList.h)

Your job is to inherit from this class and implement its methods. Your class should be called `VectorNumberList`. Its prototype should be placed in `VectorNumberList.h` Its implementation should be placed in `VectorNumberList.cpp`. You should write a Makefile that compiles your code.

Your implementation should compile using `Make & g++` under Ubuntu 16.04.

Your implementation should use an STL vector for storing the ints provided through the methods in the class.

If you do the extra credit, it should use an array or arrays.

- `virtual void addNumberToList(int num) = 0;`
  - This should add one integer to the end of your list
- `virtual void removeNumberFromFront() = 0;`
  - This remove add one integer to the beginning of your list
- `virtual void removeNumberFromBack() = 0;`
  - This remove add one integer to the end of your list
- `virtual void getNumberAt(int index) = 0;`
  - This should return the number at the specified index.
- `virtual int getSizeOfList() = 0;`
  - This should return the length of the list.
- `virtual void clear() = 0;`
  - This should empty your list.
- `virtual void printNumbers() = 0;`
  - This print your list in order, one number per line of text on the terminal.

- virtual void computeNextFibonacciAndInsertAtEnd() = 0;
  - This should take the last 2 entries in your list, add them together, and insert the result at the end of the list.
- virtual void computeNFibonacci(int n) = 0;
  - This should do what computeNextFibonacciAndInsertAtEnd() does n times.
- virtual void copyListIntoMe(NumberList &numList) = 0;
  - This should replace the contents of the list from which it was called with those of a list passed to it.
- virtual void reverseList() = 0;
  - This should reverse the order of the entries in the list.

If you successfully complete this assignment, implementing `VectorNumberList`, you have the option of implementing `ArrayNumberList`. A correct implementation of `ArrayNumberList` would use an array of integers, rather than a vector. Doing this is worth 5 points of extra credit on this assignment, or a half a point point on your final grade.