

```
(Vector.h)

#pragma once

class Vector
{
public:
    Vector(void);
    Vector(int initialCap);
    ~Vector(void);
    void addElement(int element);
    int getElementAt(int index);

protected:
    void initialize(int capacity);
    void grow(void);

    int nbElements;
    int capacity;
    float factor;
    int *elements;
};
```

(vector.cpp)

```
#include "Vector.h"
#include <math.h>

Vector::Vector(void)
{
    this->initialize(10);
}

Vector::Vector(int initialCap)
{
    this->initialize(initialCap);
}

Vector::~Vector(void)
{
    delete this->elements;
}

void Vector::initialize(int capacity)
{
    this->capacity = capacity;
    this->elements = new int[capacity];
    this->nbElements = 0;
    this->factor = 2;
}

void Vector::addElement(int element)
{
    if (this->nbElements == this->capacity){
        this->grow();
    }
    this->elements[this->nbElements++] = element;
}

void Vector::grow(void)
{
    this->capacity = (int)floor(capacity * this->factor);
    int *newArray = new int[capacity];
    for (int i=0; i!=nbElements; i++){
        newArray[i] = this->elements[i];
    }
    delete this->elements;
    this->elements = newArray;
}

int Vector::getElementAt(int index)
{
    if (index >= this->nbElements) throw 1;
    return this->elements[index];
}
```