Lab11: classes

Dian-Lun Lin Yasin Zamani
Department of Electrical and Computer Engineering
University of Utah, Salt Lake City, UT

Class - Array

```
 1 #include <stdlib.h>
 2
 3 #include <iostream>
 4
 5 // Array
 6 int* construct_array(int n) {
 7   int* a = (int*)malloc(n * sizeof(int));
 8   return a;
 9 }
10
11 void destroy_array(int* a) { free(a); }
12
13 void read_array(int* a, int n) {
14   std::cout << "Enter " << n << " numbers: ";
15   for (int i = 0; i < n; ++i) std::cin >> a[i];
16 }
17
18 void print_array(int* a, int n) {
19   for (int i = 0; i < n; ++i) std::cout << a[i] << ' ';
20   std::cout << '\n';
21 }
22
23 int* add_array(int* a, int* b, int n) {
24   int* c = construct_array(n);
25   for (int i = 0; i < n; ++i) c[i] = a[i] + b[i];
26   return c;
27 }
```

Class - Array

```h
#pragma once
#include <iostream>

class Array {

  public:

    Array(size_t n);

    ~Array();

    void read();

    void print();

    //array = array + b
    void add(Array& b);

    bool operator == (Array& b);
};
```

```cpp
#include <iostream>
#include "array.h"
#include <climits>

Array::Array(size_t n): _n{n} {
  _data = (int*)malloc(n * sizeof(int));
}

Array::~Array() {
  free(_data);
}

void Array::read() {
  std::cout << "Enter " << _n << " numbers:\n";
  for(size_t i = 0; i < _n; ++i) {
    std::cin >> _data[i];
  }
}

void Array::print() {
  for(size_t i = 0; i < _n; ++i) {
    std::cout << _data[i] << ' ';
  }
  std::cout << '\n';
}

void Array::add(Array& b) {
  for(size_t i = 0; i < _n; ++i) {
    _data[i] += b._data[i];
  }
}

bool Array::operator == (Array& b) {
  for(size_t i = 0; i < _n; ++i) {
    if(_data[i] != b._data[i]) {
      return false;
    }
  }
  return true;
```

Class - Array

```
#include "array.h"
#include <iostream>

int main() {
  Array a(3);
  Array b(3);

  a.read();
  b.read();

  a.print();
  b.print();

  std::cout << "check equivalent: "<< (a == b) << '\n';

  a.add(b);

  a.print();
  b.print();
}
```

Lab assignment

- Implement operator !=
- Implement two member functions:
  - max()
  - sum()

```
std::cout << "Summation: " << a.sum() << '\n';
std::cout << "Maximum element: " << a.max() << '\n';
std::cout << "check inequivalent: "<< (a != b) << '\n';
```

```
Summation: 12
Maximum element: 6
check inequivalent: 1
```