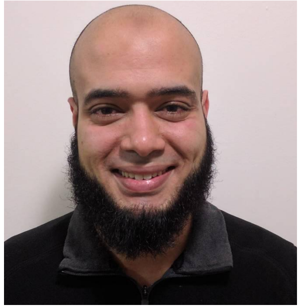
# C++ Programming

# Class Copy Constructor

# Homework

**Mostafa S. Ibrahim**
*Teaching, Training and Coaching since more than a decade!*

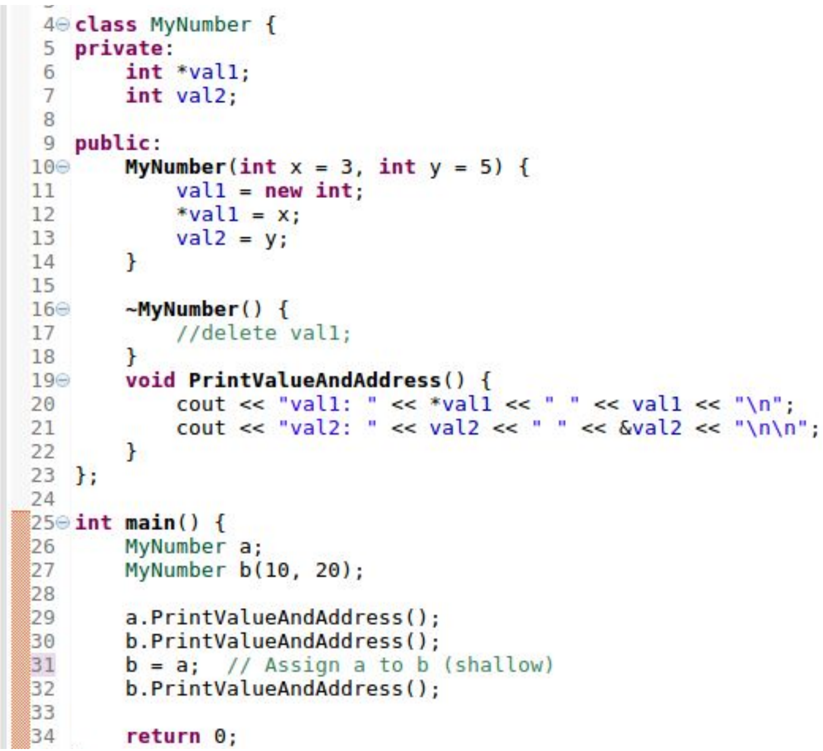
*Artificial Intelligence & Computer Vision Researcher*
*PhD* from Simon Fraser University - Canada
*Bachelor / Msc* from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



# Homework 1: copy constructor

- The format we used for copy constructor argument
  - const MyNumber &another
- 1) What happens if we moved a constructor to the **private** section?
  - Think first. Try second
- 2) What happens if we moved the copy constructor to the **private** section?
- 3) C++ won’t allow you to not use the &, Why?
  - What kind of behaviour will happen?
- 4) C++ don’t force you to use const. Why is it a good practice to use it?

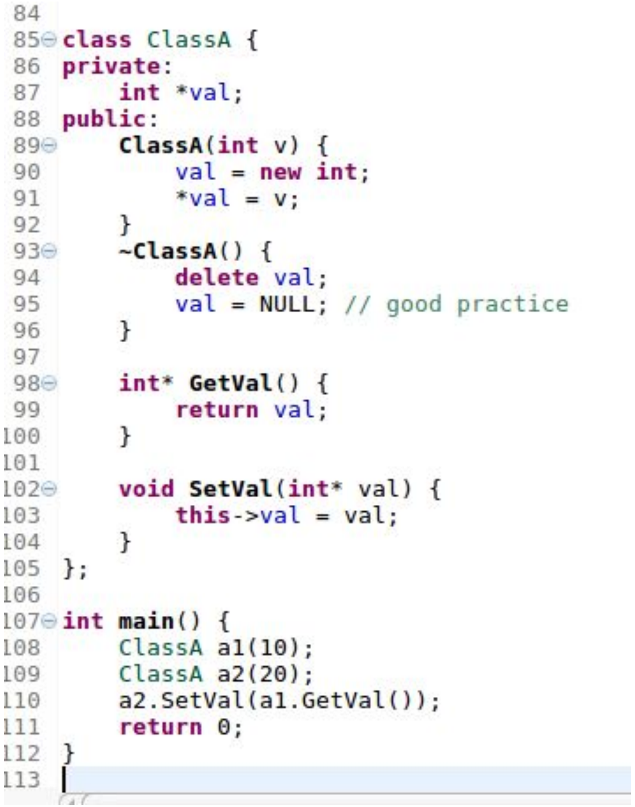
# Homework 2: MyNumber Comment!



```
 4 class MyNumber {
 5 private:
 6     int *val1;
 7     int val2;
 8 
 9 public:
10     MyNumber(int x = 3, int y = 5) {
11         val1 = new int;
12         *val1 = x;
13         val2 = y;
14     }
15 
16     ~MyNumber() {
17         //delete val1;
18     }
19     void PrintValueAndAddress() {
20         cout << "val1: " << *val1 << " " << val1 << "\n";
21         cout << "val2: " << val2 << " " << &val2 << "\n\n";
22     }
23 };
24 
25 int main() {
26     MyNumber a;
27     MyNumber b(10, 20);
28 
29     a.PrintValueAndAddress();
30     b.PrintValueAndAddress();
31     b = a;  // Assign a to b (shallow)
32     b.PrintValueAndAddress();
33 
34     return 0;
```

- We already studied this code for assigning objects
- Why did we commented line 17?
    - Think first. Try second

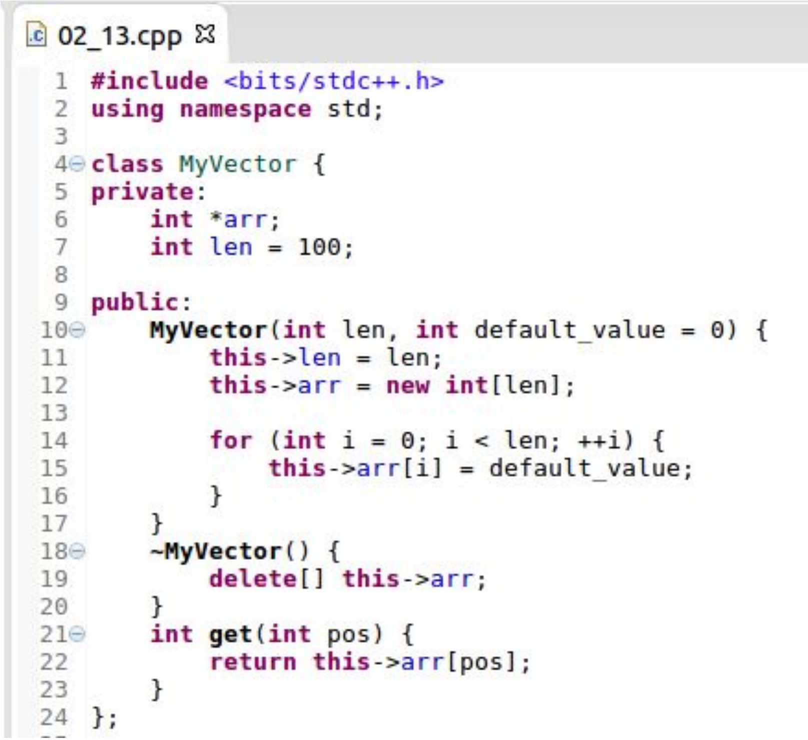
# Homework 3: Smart code!



```
class ClassA {
private:
    int *val;
public:
    ClassA(int v) {
        val = new int;
        *val = v;
    }
    ~ClassA() {
        delete val;
        val = NULL; // good practice
    }

    int* GetVal() {
        return val;
    }

    void SetVal(int* val) {
        this->val = val;
    }
};

int main() {
    ClassA a1(10);
    ClassA a2(20);
    a2.SetVal(a1.GetVal());
    return 0;
}
```

- What will happen when this code run?
- Why?
- How to change to prevent user from such wrong usage?
- Any tips for the coder who did so?

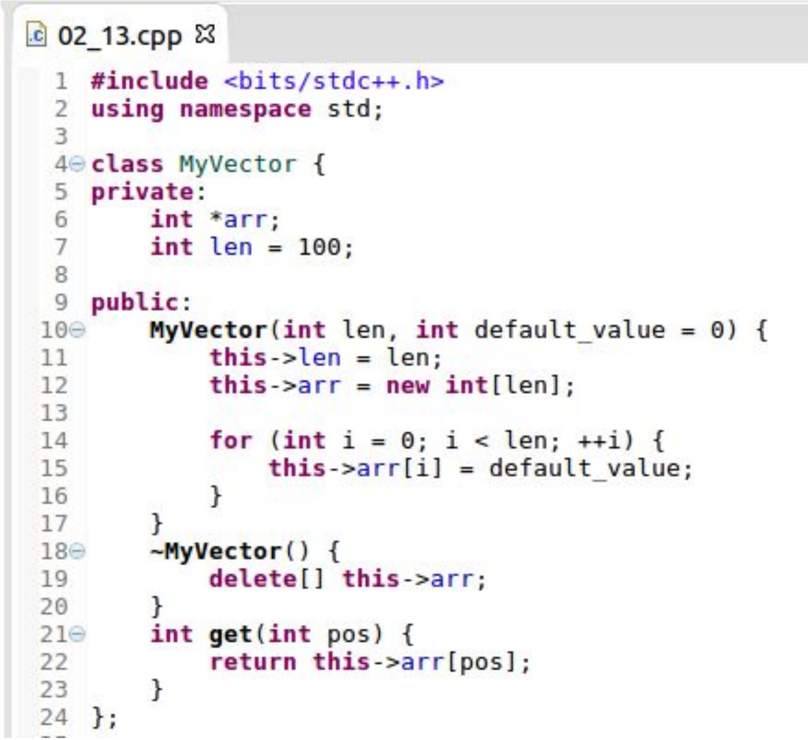
# Homework 4: MyVector



```cpp
#include <bits/stdc++.h>
using namespace std;

class MyVector {
private:
    int *arr;
    int len = 100;

public:
    MyVector(int len, int default_value = 0) {
        this->len = len;
        this->arr = new int[len];

        for (int i = 0; i < len; ++i) {
            this->arr[i] = default_value;
        }
    }
    ~MyVector() {
        delete[] this->arr;
    }
    int get(int pos) {
        return this->arr[pos];
    }
};
```

- We already coded this before
- Consider following enhancements
- 1) Provide Copy Constructor
- 2) In get method, make sure no out of index access
- 3) provide set method to change array content
- 4) method to get array length

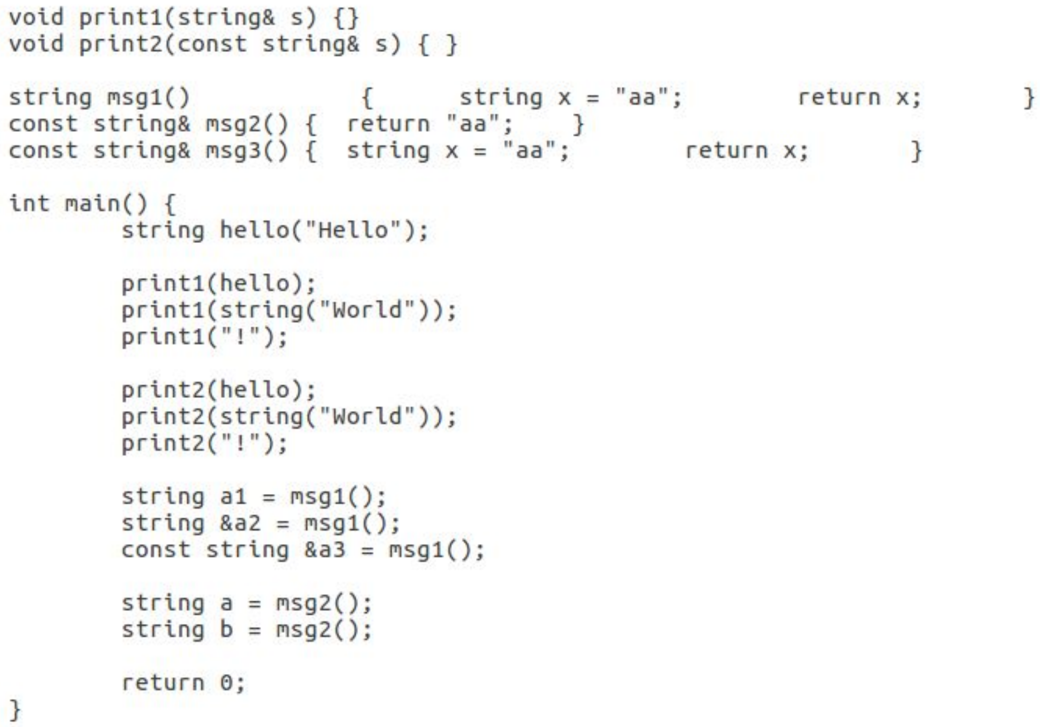
# Homework 5: MyVector &



```cpp
#include <bits/stdc++.h>
using namespace std;

class MyVector {
private:
	int *arr;
	int len = 100;

public:
	MyVector(int len, int default_value = 0) {
		this->len = len;
		this->arr = new int[len];

		for (int i = 0; i < len; ++i) {
			this->arr[i] = default_value;
		}
	}
	~MyVector() {
		delete[] this->arr;
	}
	int get(int pos) {
		return this->arr[pos];
	}
};
```

- One of developers wanted to add get length functionality. Coded as following:
- int& GetLen() {
-         return len;
- }
- Notice the **int&** (return by reference)
- Which OOP concept is violated?
- In main, show how a user can abuse our class with such mistake

# Homework 6: Be Careful!



```
void print1(string& s) {}
void print2(const string& s) { }

string msg1()            {      string x = "aa";        return x;       }
const string& msg2() {  return "aa";    }
const string& msg3() {  string x = "aa";        return x;      }

int main() {
        string hello("Hello");

        print1(hello);
        print1(string("World"));
        print1("!");

        print2(hello);
        print2(string("World"));
        print2("!");

        string a1 = msg1();
        string &a2 = msg1();
        const string &a3 = msg1();

        string a = msg2();
        string b = msg2();

        return 0;
}
```

- In this code, which lines of code don’t compile. Why?
- Which lines of code cause RTE?

# Homework 7: memory leak

- Let’s say you finished coding a project and run it.
- Think in one way to indicate this code probably has a memory leak, without checking the code itself.
    - You can only run it as much as you want
- Answer

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”