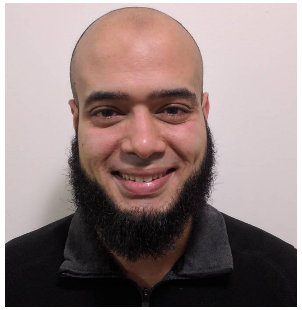
C++ Programming

Operator Overloading

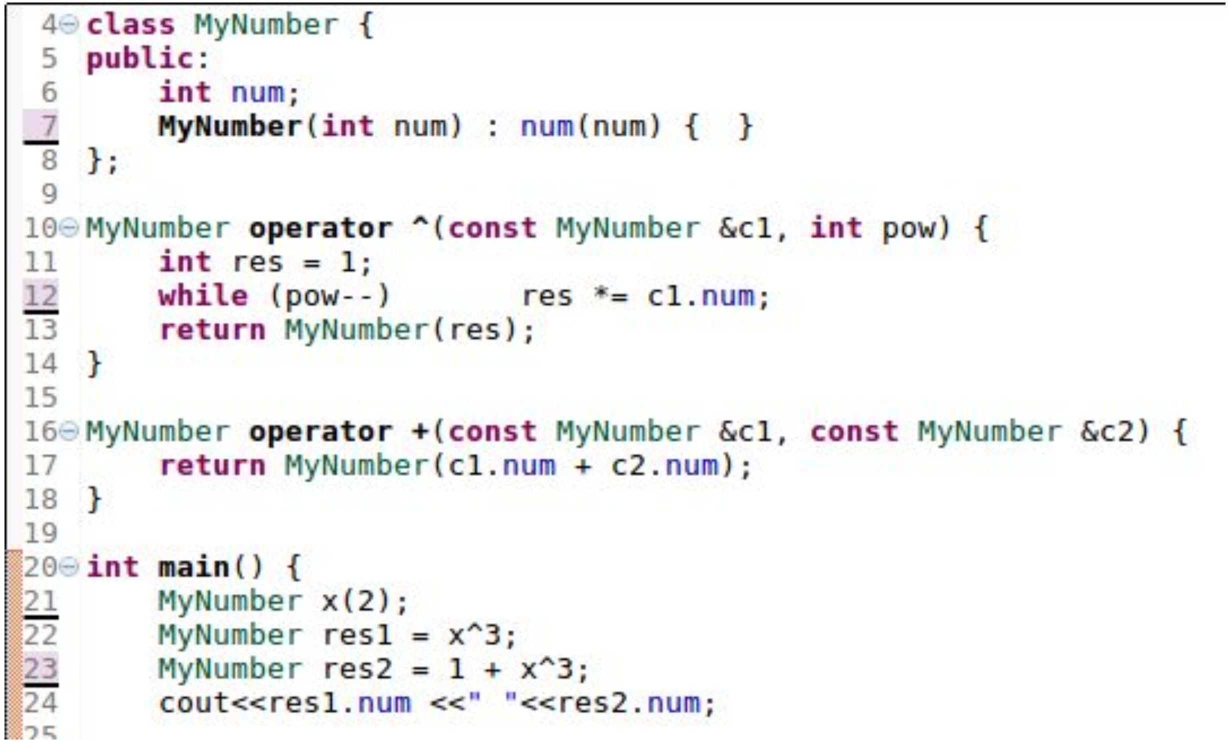
Homework 3

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)

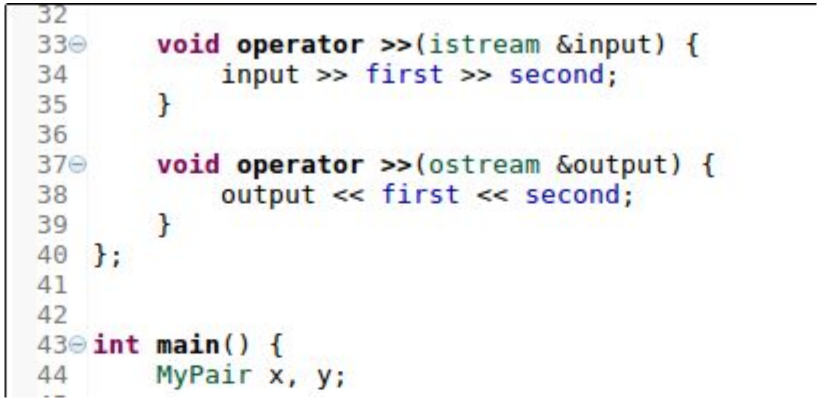


Homework 1: Guess the output



```
class MyNumber {
public:
    int num;
    MyNumber(int num) : num(num) {  }
};

MyNumber operator ^(const MyNumber &c1, int pow) {
    int res = 1;
    while (pow--)       res *= c1.num;
    return MyNumber(res);
}

MyNumber operator +(const MyNumber &c1, const MyNumber &c2) {
    return MyNumber(c1.num + c2.num);
}

int main() {
    MyNumber x(2);
    MyNumber res1 = x^3;
    MyNumber res2 = 1 + x^3;
    cout<<res1.num <<" "<<res2.num;
```

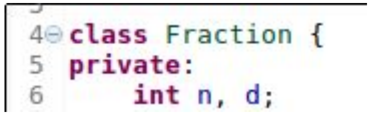
Homework 2: Reading and Writing



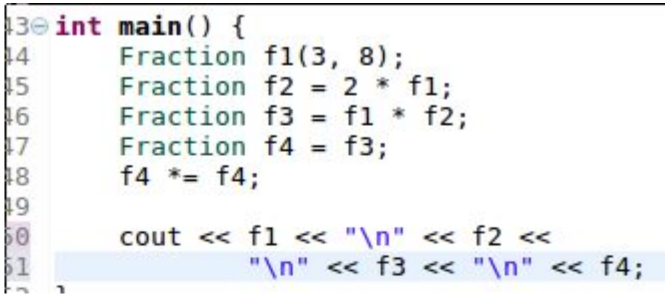
```
    void operator >>(istream &input) {
        input >> first >> second;
    }

    void operator >>(ostream &output) {
        output << first << second;
    }
};


int main() {
    MyPair x, y;
```

- In MyPair class:
- A fresh software engineer overloaded << operator this way
- How to use for reading in main?
    - Add needed lines to cin x, y
    - Add needed lines to cout x, y
- Provide tips

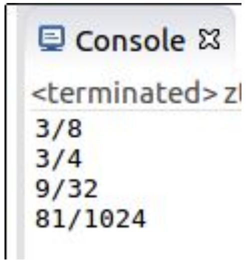
Homework 3: Fraction



```
class Fraction {
private:
    int n, d;
```



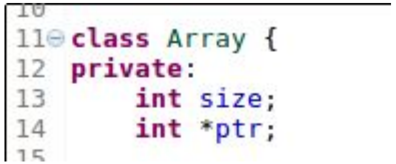
```
int main() {
    Fraction f1(3, 8);
    Fraction f2 = 2 * f1;
    Fraction f3 = f1 * f2;
    Fraction f4 = f3;
    f4 *= f4;

    cout << f1 << "\n" << f2 <<
            "\n" << f3 << "\n" << f4;
```



```
<terminated> z
3/8
3/4
9/32
81/1024
```

- Implement fraction class that support these operations
  - Use built in __gcd function
  - Or use
    - int gcd(int a, int b) {
      - return b == 0 ? a : gcd(b, a % b);
    - }
  - Feel free to skip fraction simplification
    - To simplify a, b
    - g = gcd(a, b)
      - a/g, b/g

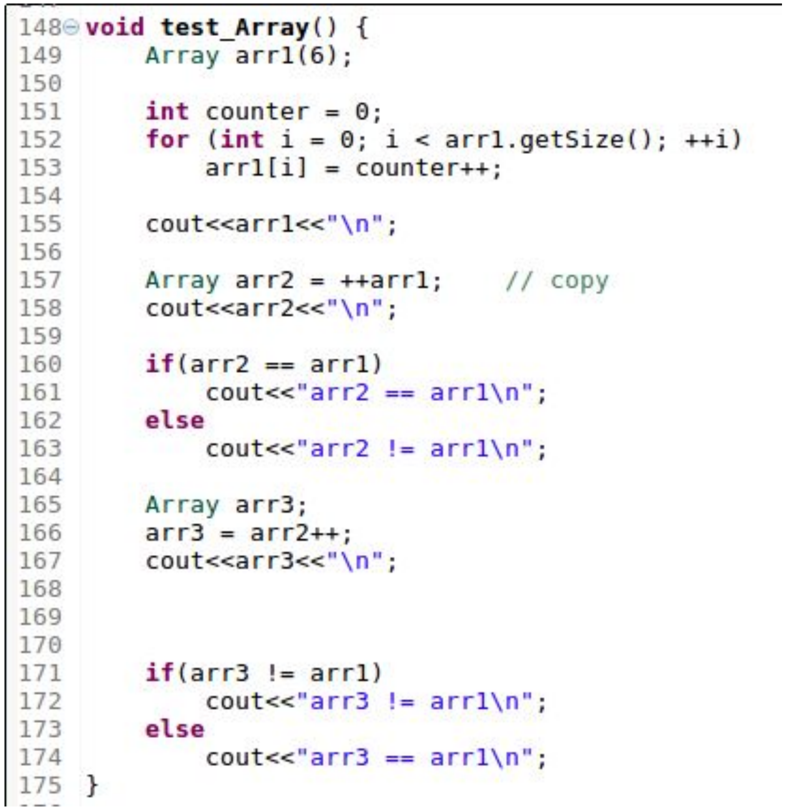
Homework 4: Array 1D



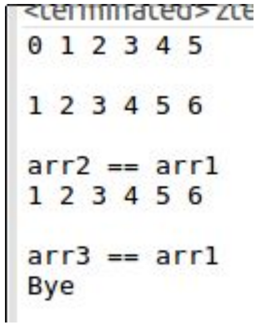
```
class Array {
private:
	int size;
	int *ptr;
```

- Implement Array class to have these members
- We need support for the following operators
  - Creating Array of N elements
  - Accessing using []
  - Cin and Cout
  - ++ prefix and postfix to increment every array cell with 1
  - Comparing: == and !=
  - **Assigning** array to another
  - See following main

Homework 4: Array 1D

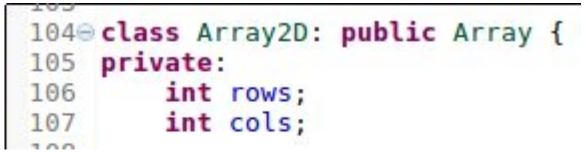


```cpp
void test_Array() {
    Array arr1(6);

    int counter = 0;
    for (int i = 0; i < arr1.getSize(); ++i)
        arr1[i] = counter++;

    cout<<arr1<<"\n";

    Array arr2 = ++arr1;    // copy
    cout<<arr2<<"\n";

    if(arr2 == arr1)
        cout<<"arr2 == arr1\n";
    else
        cout<<"arr2 != arr1\n";

    Array arr3;
    arr3 = arr2++;
    cout<<arr3<<"\n";



    if(arr3 != arr1)
        cout<<"arr3 != arr1\n";
    else
        cout<<"arr3 == arr1\n";
}
```



```
<terminated>
0 1 2 3 4 5

1 2 3 4 5 6

arr2 == arr1
1 2 3 4 5 6

arr3 == arr1
Bye
```

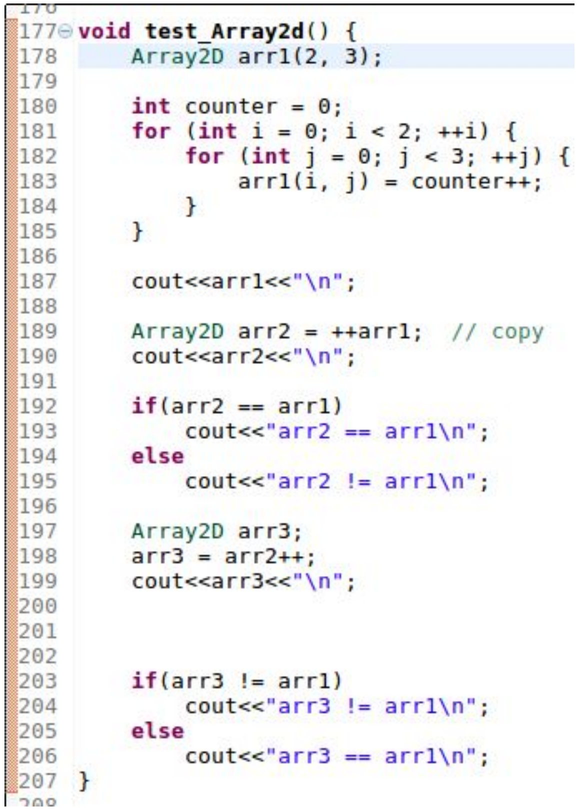
Homework 5: Array 2D



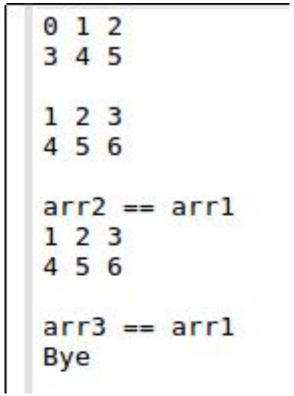
```
class Array2D: public Array {
private:
    int rows;
    int cols;
```

- We will extend the Array class we did and make use of it as much as possible
- Provide access as arr(i, j)

Homework 5: Array 2D



```cpp
void test_Array2d() {
    Array2D arr1(2, 3);

    int counter = 0;
    for (int i = 0; i < 2; ++i) {
        for (int j = 0; j < 3; ++j) {
            arr1(i, j) = counter++;
        }
    }

    cout<<arr1<<"\n";

    Array2D arr2 = ++arr1;  // copy
    cout<<arr2<<"\n";

    if(arr2 == arr1)
        cout<<"arr2 == arr1\n";
    else
        cout<<"arr2 != arr1\n";

    Array2D arr3;
    arr3 = arr2++;
    cout<<arr3<<"\n";



    if(arr3 != arr1)
        cout<<"arr3 != arr1\n";
    else
        cout<<"arr3 == arr1\n";
}
```



```
0 1 2
3 4 5

1 2 3
4 5 6

arr2 == arr1
1 2 3
4 5 6

arr3 == arr1
Bye
```

Homework 6: Operators in Company Payroll

- Recall the Another Company Payroll Homework in polymorphism
  - Properties: Comparable, Printable, Cloneable
- Change the code to support operator <<
  - E.g. cout<<some_payable
- Change Comparable to use operator < instead of the Compare Function

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."