Class- this Pointer

Introduction

In C++ each object gets its own copy of data members and all objects share a single copy of member functions.

Every object in C++ has access to its own address through an important pointer called this pointer.

The *this* pointer is an implicit parameter to all member functions. Therefore, inside a member function, this may be used to refer to the invoking object. It is used:

- When local variable's name is same as member's name
- To return reference to the calling object

Example

```
#include<iostream>
using namespace std;

class Test
{
private:
    int x;
    int y;
public:
    Test(int x = 0, int y = 0) { this->x = x; this->y = y; }

    Test &setX(int a) { x = a; return *this; }
    Test &setY(int b) { y = b; return *this; }
    void print() { cout << "x = " << x << " y = " << y << endl; }
};

int main()
{
    Test obj1(5, 5);
    // Chained function calls. All calls modify the same object
    // as the same object is returned by reference
    obj1.setX(10).setY(20);
    obj1.print();
    return 0;
}
```

Exercise

The goal of this exercise is to declare and define a class *ArrayComplex* which will store an array of complex numbers. For this you will need, first, to define a class named *Complex* (with a Cartesian representation of a complex number).

The class *ArrayComplex* will contain 2 attributes:

- The *size* of this array.
- A pointer *ptrArrayComplex* on a buffer dynamically allocated which will contain all the complex numbers.

But also, several methods:

3 constructors (a default constructor, a copy constructor and a constructor with 1 parameter _*size* which will explicitly allocate the array).

- An *Alloc* method which will explicitly allocate the array.
- A *Free* method which will explicitly free the memory associated to the array.
- A *ValueAt* method which will give access in read only to the value at the position index within the array.
- A *ValueAt* method which will give access in read and write to the value at the position index within the array.

Create (instantiate) in a main function a local and a dynamically allocated object corresponding to this class and test all the methods.