```cpp
#include <iostream>
#include <string>
using namespace std;
class IntStack
{
private:
    int* stackArray;
    int stackSize;
    int numElements;
public:
    IntStack(int);
    ~IntStack();

    void push(int);
    void pop(int&);
    bool isFull() const;
    bool isEmpty() const;
    void display() const;
    // Overloaded = operator
    const IntStack operator=(const IntStack& right)
    {
        delete[] this->stackArray;
        this->stackSize = right.stackSize;
        this->numElements = right.numElements;
        this->stackArray = new int[stackSize];
        for (int i = 0; i < numElements; i++)
            this->stackArray[i] = right.stackArray[i];
        return *this;
    }

    // Copy constructor
    IntStack(const IntStack& obj)
    {
        this->stackSize = obj.stackSize;
        this->numElements = obj.numElements;
        this->stackArray = new int[stackSize];
        for (int i = 0; i < stackSize; i++)
            this->stackArray[i] = obj.stackArray[i];
    }
};
IntStack::IntStack(int Size)
{
    stackArray = new int[Size];
    stackSize = Size;
    numElements = 0;
}
IntStack::~IntStack()
{
    delete[] stackArray;
```

```
}
void IntStack::push(int num)
{
    if (isFull())
        cout << "The stack is full.\n";
    else
    {
        stackArray[numElements] = num;
        numElements++;
    }
}
void IntStack::pop(int &num)
{
    if (isEmpty())
        cout << "The stack is empty.\n";
    else
    {
        numElements--;
        num = stackArray[numElements];
    }
}
bool IntStack::isFull() const
{
    return numElements == stackSize;
}
bool IntStack::isEmpty() const
{
    return numElements == 0;
}
void IntStack::display() const
{
    if (isEmpty())
        cout << "The stack is empty.\n";
    else
        for (int i = 0; i < numElements; i++)
            cout << stackArray[i] << "\t";
    cout << endl;
}

int main()
{
    IntStack stack1(6);
    stack1.push(5);
    stack1.push(10);
    stack1.push(15);

    IntStack stack2(6);
    stack2 = stack1;
    stack2.push(20);
```

```cpp

    IntStack stack3 = stack1;
    stack3.push(25);

    stack1.display();
    stack2.display();
    stack3.display();
    return 0;
}



```