

(Operator Overloading)

Given the following main function, write the:

- 1) Default & copy constructors.
- 2) Destructor.
- 3) Conditional operators (==) and (!=).
- 4) Arithmetic unary (++) and binary (+) operators.
- 5) Assignment operators (=) and (+=).
- 6) Const & non-const object subscript operators ([]).
- 7) Insertion (<<) and extraction (>>) operators.

```
# include <iostream>
# include "Array.h"
using namespace std;
int main ( )
{
    Array a ( 7 ), b;                // default size = 10

    cout << "enter the elements of Array a:" << endl;
    cin >> a;

    b = a;

    if ( a == b )
        cout << "these arrays are equal" << endl;

    a ++;

    if ( a != b )
        cout << "now, these arrays are not equal" << endl;

    Array c ( a );

    c += a;

    cout << c << endl;

    c [ 3 ] ++;

    cout << "value of c [ 3 ] = " << c [ 3 ] << endl;

    return 0;
}
```

ديناران هدية عند التنبيه على كل خطأ بمذكرات الموقع برسالة SMS أو بالبريد الإلكتروني

مواد عامة: Physics I/II, Circuits, English 123, Numerical, Dynamics, Strength, Statics

مواد كمبيوتر: C++, Java, MATLAB, Data Structures, Algorithms, Discrete Math, Digital Logic, Concepts

مواد تصميم: Mechanical Design I/II, Structural Analysis I/II, Concrete I/II, Soil, Fluid Mechanics, System Dynamics

(Operator Overloading)

```

#include <iostream>
using namespace std;

class Array
{
    friend void operator<< ( ostream &, const Array & );
    friend void operator>> ( istream &, Array & );

private:
    int *x;
    int size;

public:
    Array (          );
    Array (          );
    ~Array ( );

    bool operator== (          );
    bool operator!= (          );
    {
    }

    const Array & operator= (          );

    void operator++ ( );

    Array operator+ (          );

    void operator+= (          );
    {
    }

    int operator[] (          );
    int & operator[] (          );
};

```

ديناران هدية عند التنبيه على كل خطأ بمذكرات الموقع برسالة SMS أو بالبريد الإلكتروني

مواد عامة: Physics I/II, Circuits, English 123, Numerical, Dynamics, Strength, Statics

مواد تصميم: C++, Java, MATLAB, Data Structures, Algorithms, Discrete Math, Digital Logic, Concepts

مواد تصميم: Mechanical Design I/II, Structural Analysis I/II, Concrete I/II, Soil, Fluid Mechanics, System Dynamics

(Operator Overloading)

```
Array::Array (           )           // default constructor
{
```

```
}
```

```
Array::Array (           )           // copy constructor
{
```

```
}
```

```
Array::~~Array (         )           // destructor
{
```

```
}
```

ديتاران هدية عند التنبيه على كل خطأ بمذكرات الموقع برسالة SMS أو بالبريد الإلكتروني

مواد عامة: [Physics I/II](#), [Circuits](#), [English 123](#), [Numerical](#), [Dynamics](#), [Strength](#), [Statics](#)

مواد تصميم: [C++](#), [Java](#), [MATLAB](#), [Data Structures](#), [Algorithms](#), [Discrete Math](#), [Digital Logic](#), [Concepts](#)

مواد تصميم: [Mechanical Design I/II](#), [Structural Analysis I/II](#), [Concrete I/II](#), [Soil](#), [Fluid Mechanics](#), [System Dynamics](#)

(Operator Overloading)

```

Array::operator==(
{
}

Array Array::operator=(
{
}
}

```

ديتاران هدية عند التنبيه على كل خطأ بمذكرات الموقع برسالة SMS أو بالبريد الإلكتروني

مواد عامة: [Physics I/II](#), [Circuits](#), [English 123](#), [Numerical](#), [Dynamics](#), [Strength](#), [Statics](#)

مواد تصميم: [C++](#), [Java](#), [MATLAB](#), [Data Structures](#), [Algorithms](#), [Discrete Math](#), [Digital Logic](#), [Concepts](#)

مواد تصميم: [Mechanical Design I/II](#), [Structural Analysis I/II](#), [Concrete I/II](#), [Soil](#), [Fluid Mechanics](#), [System Dynamics](#)

(Operator Overloading)

```
Array::operator++ ( )
{
```

```
}
```

```
Array Array::operator+ ( )
{
```

```
}
```

ديتاران هدية عند التنبيه على كل خطأ بمذكرات الموقع برسالة SMS أو بالبريد الإلكتروني

مواد عامة: [Physics I/II](#), [Circuits](#), [English 123](#), [Numerical](#), [Dynamics](#), [Strength](#), [Statics](#)

مواد تصميم: [C++](#), [Java](#), [MATLAB](#), [Data Structures](#), [Algorithms](#), [Discrete Math](#), [Digital Logic](#), [Concepts](#)

مواد تصميم: [Mechanical Design I/II](#), [Structural Analysis I/II](#), [Concrete I/II](#), [Soil](#), [Fluid Mechanics](#), [System Dynamics](#)

(Operator Overloading)

```
void operator<< ( ostream & output, const Array &a )  
{
```

```
}
```

```
void operator>> ( istream &input, Array &a )  
{
```

```
}
```

ديتاران هدية عند التنبيه على كل خطأ بمذكرات الموقع برسالة SMS أو بالبريد الإلكتروني

مواد عامة: [Physics I/II](#), [Circuits](#), [English 123](#), [Numerical](#), [Dynamics](#), [Strength](#), [Statics](#)

مواد تصميم: [C++](#), [Java](#), [MATLAB](#), [Data Structures](#), [Algorithms](#), [Discrete Math](#), [Digital Logic](#), [Concepts](#)

مواد تصميم: [Mechanical Design I/II](#), [Structural Analysis I/II](#), [Concrete I/II](#), [Soil](#), [Fluid Mechanics](#), [System Dynamics](#)

```
int Array::operator[ ] (          )
{
```

```
}
```

```
int  Array::operator[ ] (          )
{
```

```
}
```

ديتاران هدية عند التنبيه على كل خطأ بمذكرات الموقع برسالة SMS أو بالبريد الإلكتروني

مواد عامة: [Physics I/II](#), [Circuits](#), [English 123](#), [Numerical](#), [Dynamics](#), [Strength](#), [Statics](#)

مواد تصميم: [C++](#), [Java](#), [MATLAB](#), [Data Structures](#), [Algorithms](#), [Discrete Math](#), [Digital Logic](#), [Concepts](#)

مواد تصميم: [Mechanical Design I/II](#), [Structural Analysis I/II](#), [Concrete I/II](#), [Soil](#), [Fluid Mechanics](#), [System Dynamics](#)