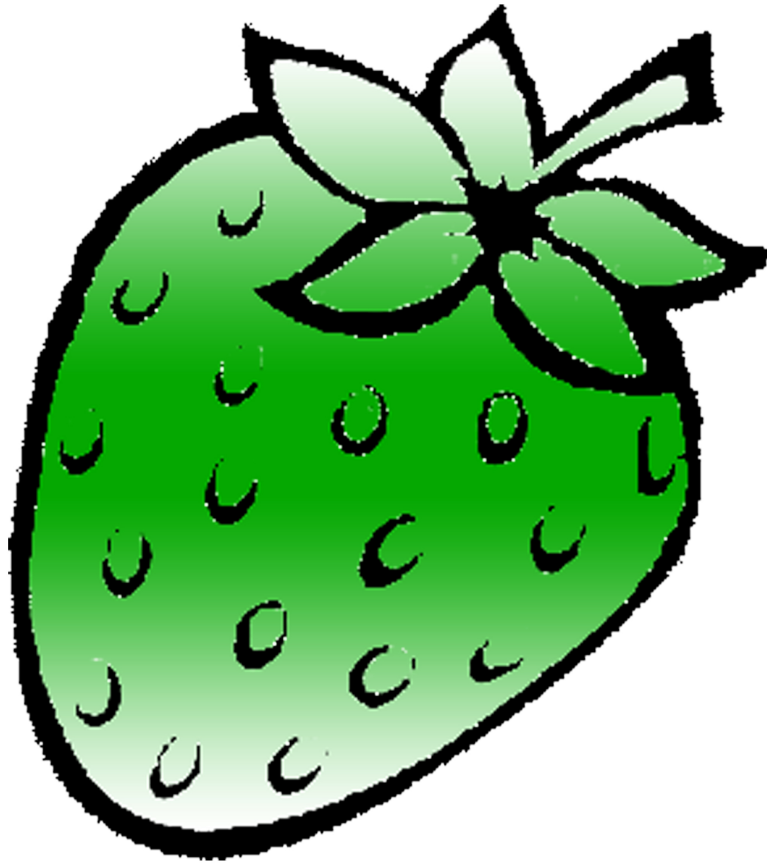


# STRAWBERRY



*/strawberrydevelopers*



*/strawberry\_app*

*For more visit:*

*Strawberrydevelopers.weebly.com*

## Experiment No. 5

### PART A

(PART A: TO BE REFERRED BY STUDENTS)

**Aim: To understand the below concept of Operator overloading.**

1. To overload unary operator.
2. To overload binary operator.

**P1:** Write a class *number* with one integer data member. Write a program to overload unary operator ++ to increment data member of object of class number.

**P2:** Create a *Matrix* class of size m \*n. Overload the '+' and '-' operator to add and subtract two *Matrix* objects respectively. Write a main function to implement it.

### A.2 Prerequisites

Sr.No	Concepts	Program No
1.	Overloading unary operator	P1
2.	Overloading Binary operator	P2

### A.3 Outcome:

After successful completion of this experiment students will be able to learn and implement the concept of operator overloading

### A.4 Theory:

Operator overloading means assigning additional meaning to existing C++ operator. We use special function known as 'operator' function to define an additional task to an operator. General form of operator function is:

```
Return type classname:: operator op(argument_list)
{
    Function Body// task defined
}
```

Developed By Strawberry

Here, `op` is the operator being overloaded. It is preceded by keyword `operator`.

**Operator `op` is a function name.**

Return type is a type of value returned by the specified operation.

Operator functions must be either member function or friend function. A member function has no arguments for unary operators and only one for binary operators. Friend function will have only one argument for unary operators and two for binary operators.

We can not overload following C++ operators:

- Class member access operator (`.`, `.*`)
- Scope Resolution Operator (`::`)
- Size operator (`sizeof`)
- Conditional operator (`?:`)

The process of overloading involves following steps:

- Create a class that defines the data type that is to be used in the overloading operation
- Declare the operator function **operator** `op()` in the public part of the class. It may be either a member function or friend function
- Define the operator function to implement the required operations.

To invoke overloaded unary operator we write: **`op x`** or **`x op`**

To invoke overloaded binary operator we write: **`x op y`**

**Example:**

```
#include <iostream.h>
#include<conio.h>
class Distance
{
private:
    int feet;           // 0 to infinite
    int inches;        // 0 to 12
public:
    // required constructors
    Distance() {
```

Developed By Strawberry

```

    feet = 0;

    inches = 0;

}

Distance(int f, int i){

    feet = f;

    inches = i;

}

// method to display distance

void displayDistance()

{

cout << "F: " << feet << " I:" << inches <<endl;

}

// overloaded minus (-) operator

Distance operator- ()

{

    feet = -feet;

    inches = -inches;

    return Distance(feet, inches);

}

};

int main()

{

    Distance D1(11, 10), D2(-5, 11);

    -D1;                // apply negation

    D1.displayDistance(); // display D1

    -D2;                // apply negation

    D2.displayDistance(); // display D2

    return 0;

}

```

Developed By Strawberry

## PART B

### (PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll No. N008	Name: AKSHAY BANDA
Program: MBA TECh COMPUTER SCIENCE	Division: C
Semester: 2	Batch : C1
Date of Experiment: 18/2/2015	Date of Submission: 18/2/2015
Grade :	

### **B.1. Software Code written by student:**

*(Paste your C++ code completed during the 2 hours of practical in the lab here)*

1.

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

class number
{
    int n;
public:
    void input()
    {
        cout<<"\nEnter number : ";
        cin>>n;
    }
    number operator++ ()
    {
        n++;
    }
    void display()
    {
        cout<<"\nNumber is : "<<n;
    }
};
```

Developed By Strawberry

```

int main()
{
    number n;
    n.input();
    ++n;
    n.display();
    return 0;
}

```

2.

```

#include<iostream>
using namespace std;

class matrix
{
    int **p,m,n,i,j;
public:
    matrix(int a, int b)
    {
        m=a;
        n=b;
        p=new int *[m];
        for(i=0;i<m;i++)
        {
            p[i]=new int[n];
        }
    }
    void input()
    {
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                cout<<"\nEnter number : ";
                cin>>p[i][j];
            }
        }
    }
    matrix operator+(matrix a)
    {

```

Developed By Strawberry

```

matrix c(m,n);
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        c.p[i][j]=p[i][j]+a.p[i][j];
    }
}
return c;
}
matrix operator-(matrix a)
{
    matrix c(m,n);
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            c.p[i][j]=p[i][j]-a.p[i][j];
        }
    }
    return c;
}
void display()
{
    cout<<"\nMatrix is\n ";
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            cout<<" "<<p[i][j];
        }
        cout<<"\n";
    }
}
};

int main()
{
    int m,n;
    cout<<"\nEnter rows : ";
    cin>>m;
    cout<<"\nEnter columns : ";

```

Developed By Strawberry

```

cin>>n;
matrix a(m,n),b(m,n),c(m,n),d(m,n);
a.input();
b.input();
cout<<"\nAdditon is : ";
c=a+b;
c.display();
cout<<"\nSubtraction is : ";
d=a-b;
d.display();
return 0;
}
int main()
{
int m,n;
cout<<"\nEnter rows : ";
cin>>m;
cout<<"\nEnter columns : ";
cin>>n;
matrix a(m,n),b(m,n),c(m,n),d(m,n);
a.input();
b.input();
cout<<"\nAdditon is : ";
c=a+b;
c.display();
cout<<"\nSubtraction is : ";
d=a-b;
d.display();
return 0;
}

```

## **B.2. Input and Output:**

*(Paste your program input and output in following format. If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)*

1.

Enter number : 7

Number is : 8

Developed By Strawberry



2.

Enter rows : 2

Enter columns : 2

Enter number : 2

Enter number : 2

Enter number : 2

Enter number : 2

Enter number : 2

Enter number : 2

Enter number : 2

Enter number : 2

Additon is :

Matrix is

4 4

4 4

Subtraction is :

Matrix is

0 0

0 0

Process returned 0 (0x0) execution time : 6.078 s

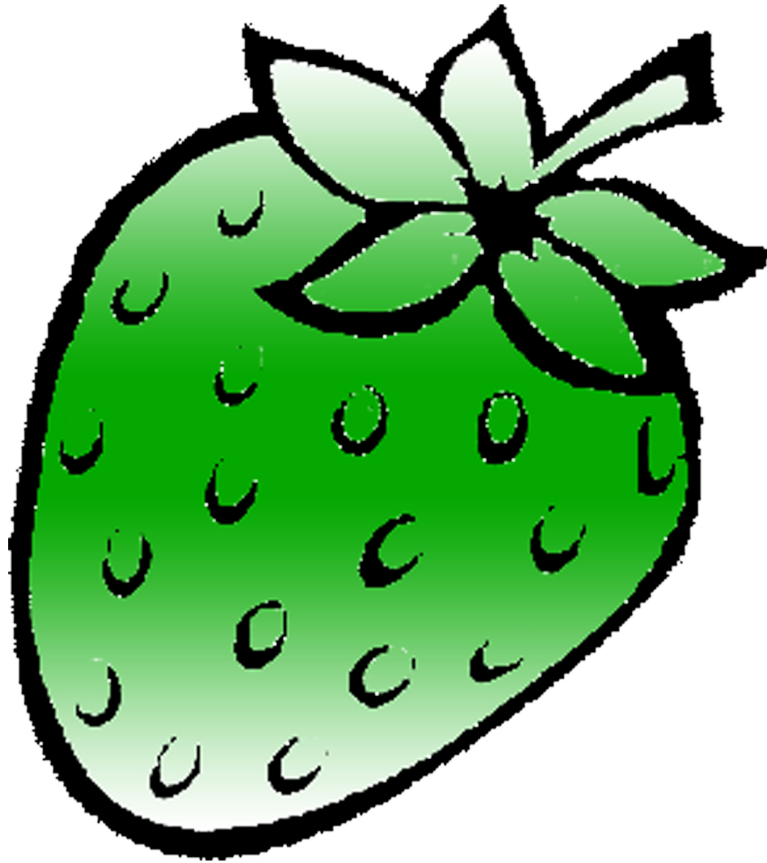
Press any key to continue.

### **B.3. Conclusion:**

*(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.1)*

Developed By Strawberry

# STRAWBERRY



*/strawberrydevelopers*



*/strawberry\_app*

*For more visit:*

*Strawberrydevelopers.weebly.com*