

OOP Assignment 3

Submitted By : Deepjyoti Deka – 190103014 - 4th Semester (CSE)

1. Write a Program to design a class having static member function named showcount which has the property of displaying the number of objects created of the class.

Ans:

Code:

```
#include <iostream>
using namespace std;
class newObject
{
    int code;
    static int count;
public:
    void setcode(void)
    {
        code = ++count;
    }
    void showcode(void)
    {
        cout << "Object Creation Number:" << code << endl;
    }
    static void showcount(void)
    {
        cout << "count:" << count << "\n";
    }
};
int newObject ::count;
int main() {
    newObject o1, o2 , o3;
    o1.setcode();
    o2.setcode();
    o3.setcode();
    newObject ::showcount();
    newObject o4 ;
    o4.setcode();
    newObject ::showcount();
}
```

```

    o1.showcode();
    o2.showcode();
    o4.showcode();
    return 0;
}

```

Input/Output:

```

03
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ g++ 1_.cpp
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out
count:3
count:4
Object Creation Number:1
Object Creation Number:2
Object Creation Number:4
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ █

```

2. Write a Program using class to process Shopping List for a Departmental Store. The list include details such as the Code No and Price of each item and perform the operations like Adding, Deleting Items to the list and Printing the Total value of a Order.

Ans:

Code:

```

#include <iostream>
using namespace std;
const int n = 100;
class ShopingList
{
    int itemCode[n];
    float itemPrice[n];
    int count;
public:
    void CNT(void) { count = 0; }
    void getitem(void);
    void displaySum(void);
    void remove(void);
}

```

```

    void displayItems(void);
};
void ShopingList ::getitem(void)
{
    cout << "Enter item code"<<endl;
    cin >> itemCode[count];
    cout << "Enter Item cost"<<endl;
    cin >> itemPrice[count];
    count++;
}
void ShopingList ::displaySum(void)
{
    float sum = 0;
    for (int i = 0; i < count; i++)
        sum = sum + itemPrice[i];
    cout << "\n Total Value:" << sum << "\n";
}
void ShopingList ::remove(void)
{
    int a;
    cout << "Enter Item Code"<<endl;
    cin >> a;
    for (int i = 0; i < count; i++)
        if (itemCode[i] == a)
            itemPrice[i] = 0;
}
void ShopingList ::displayItems(void)
{
    cout << "\nCode Price\n ";
    for (int i = 0; i < count; i++)
    {
        cout << "\n" << itemCode[i];
        cout << "          " << itemPrice[i];
    }
    cout << "\n";
}
int main()
{
    {
        ShopingList order;
        order.CNT();
        int x;
        do
        {

```

```
    cout << "\n Inputs";
    cout << "\n1 : Add";
    cout << "\n2 : Display Total";
    cout << "\n3 : Delete Item";
    cout << "\n4 : Display all ShopingList";
    cout << "\n5 : Exit"<<endl;
    cin >> x;
    switch (x)
    {
    case 1:
        order.getitem();
        break;
    case 2:
        order.displaySum();
        break;
    case 3:
        order.remove();
        break;
    case 4:
        order.displayItems();
        break;
    default:
        cout << "Error in input";
    }
    } while (x != 5);
    return 0;
}
}
```

Input/Output:

```
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ g++ 2.cpp
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out

Inputs
1 : Add
2 : Display Total
3 : Delete Item
4 : Display all ShopingList
5 : Exit
1
Enter item code
20
Enter Item cost
2

Inputs
1 : Add
2 : Display Total
3 : Delete Item
4 : Display all ShopingList
5 : Exit
2

Total Value:2

Inputs
1 : Add
2 : Display Total
3 : Delete Item
4 : Display all ShopingList
5 : Exit
4

Code Price
```

3. Write a Program which creates & uses array of object of a class.(for eg. implementing the list of Managers of a Company having details such as Name, Age, etc..).

Ans:

Code:

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

```

{
    char name[30];
    int age;
public:
    void getdata(void);
    void insertData(void);
};

void Manager ::getdata(void)
{
    cout << "Enter Manager Name " << endl;
    cin >> name;
    cout << "Enter his/her Age " <<endl;
    cin >> age;
}

void Manager ::insertData(void)
{
    cout << "Name: " << name << "\n";
    cout << "Age: " <<age<<"\n ";
}

const int size = 3;
int main()
{
    Manager manager[size];
    for (int i = 0; i < size; i++)
    {
        cout << "Enter the details Serially for " << i + 1 << "\n";
        manager[i].getdata();
    }
    cout << "\n";
    for (int i = 0; i < size; i++)
    {
        cout << "\n Manager" << i + 1 << "\n";
        manager[i].insertData();
    }
    return 0;
}

```

Input/Output:

```
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ g++ 3.cpp
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out
Enter the details Serially for 1
Enter Manager Name
Ram
Enter his/her Age
18
Enter the details Serially for 2
Enter Manager Name
Sham
Enter his/her Age
20
Enter the details Serially for 3
Enter Manager Name
█
```

4. Write a Program to find Maximum out of Two Numbers using friend function.
Note: Here one number is a member of one class and the other number is member of some other class.

Ans:

Code:

```
#include <iostream>
using namespace std;
class Test1;
class Test2
{
    int x;
public:
    void setvalue(int i)
    {
        x = i;
    }
    friend void max(Test2, Test1);
};
class Test1
{
    int a;
```

```

public:
    void setvalue(int i)
    {
        a = i;
    }
    friend void max(Test2, Test1);
};

void max(Test2 number1, Test1 number2)
{
    if (number1.x >= number2.a)
        cout << "The max is " << number1.x << endl;
    else
        cout << "The max is " << number2.a << endl;
}

int main()
{
    Test1 Test1;
    Test1.setvalue(5);
    Test2 Test2;
    Test2.setvalue(6);
    max(Test2, Test1);
    return 0;
}

```

Input/Output:

```

deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ^C
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ g++ 4.cpp
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out
The max is 6
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ █

```


5. Write a Program to swap private data members of classes named as class_1, class_2 using friend function.

Ans:

Code:

```
#include <iostream>
using namespace std;
class class1;
class class2
{
    int value1;
public:
    void indata(int a)
    {
        value1 = a;
    }
    void display(void)
    {
        cout << value1 << "\n";
    }
    friend void exchange(class2 &, class1 &);
};
class class1
{
    int value2;
public:
    void indata(int a)
    {
        value2 = a;
    }
    void display(void)
    {
        cout << value2 << "\n";
    }
    friend void exchange(class2 &, class1 &);
};
void exchange(class2 &x, class1 &y)
{
    int temp = x.value1;
    x.value1 = y.value2;
    y.value2 = temp;
}
int main()
```

```

{
    class2 C1;
    class1 C2;
    C1.indata(6);
    C2.indata(4);
    cout << "Values before "
          << "\n";
    C1.display();
    C2.display();
    exchange(C1, C2);
    cout << "Values after" << "\n";
    C1.display();
    C2.display();
    return 0;
}

```

Input/Output:

```

deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ^C
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ g++ 5.cpp
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out
Values before
6
4
Values after
4
6
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ 

```

6. Write a Program to design a class complex to represent complex numbers. The complex class should use an external function (use it as a friend function) to add two complex numbers. The function should return an object of type complex representing the sum of two complex numbers.

Ans:

Code:

```
#include <iostream>
using namespace std;
class complex
{
    float x;
    float y;
public:
    void input(float real, float img)
    {
        x = real;
        y = img;
    }
    friend complex sum(complex, complex);
    void show(complex);
};

complex sum(complex c1, complex c2)
{
    complex c3;
    c3.x = c1.x + c2.x;
    c3.y = c1.y + c2.y;
    return (c3);
}

void complex ::show(complex c)
{
    cout << c.x << " + j " << c.y << "\n";
}

int main()
{
    complex A, B, C;
    A.input(1, 9.4);
    B.input(2.5, 5.2);
    C = sum(A, B);
    cout << "A = ";
    A.show(A);
    cout << "B = ";
```

```
B.show(B);  
cout << "C = ";  
C.show(C);  
return 0;  
}
```

Input/Output:

```
g++: fatal error: no input files  
compilation terminated.  
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ g++ 6.cpp  
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out  
A = 1 + j 9.4  
B = 2.5 + j 5.2  
C = 3.5 + j 14.6  
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$
```

7. Write a Program using copy constructor to copy data of an object to another object.

Ans:

Code:

```
#include <iostream>
using namespace std;
class CopyObject
{
    int id;
public:
    CopyObject() {}
    CopyObject(int a)
    {
        id = a;
    }
    CopyObject(CopyObject &x)
    {
        id = x.id;
    }
    void display(void)
    {
        cout << id;
    }
};
int main()
{
    CopyObject A(310);
    CopyObject B(A);
    CopyObject C = A;
    CopyObject D;
    D = A;
    cout << "\n The id of the A ";
    A.display();
    cout << "\n The id of the B ";
    B.display();
    cout << "\n The id of the C ";
    C.display();
    cout << "\n The id of the D ";
    D.display();
    cout << endl;
    return 0;
}
```

Input / output:

```
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ g++ 7.cpp
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out

The id of the A 310
The id of the B 310
The id of the C 310
The id of the D 310
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$
```

8. Write a Program to allocate memory dynamically for an objects of a given class using class's constructor.

Ans:

Code:

```
#include <iostream>
#include <string.h>
using namespace std;

class String
{
    char *name;
    int length;
public:
    String()
    {
        length = 0;
        name = new char[length + 1];
    }
    String(char *s)
    {
        length = strlen(s);
        name = new char[length + 1];
        strcpy(name, s);
    }
    void display(void)
    {
        cout << name << "\n";
    }
}
```

```

    }
    void join(String &a, String &b);
};
void String ::join(String &a, String &b)
{
    length = a.length + b.length;
    delete name;
    name = new char[length + 1];
    strcpy(name, a.name);
    strcat(name, b.name);
};
int main()
{
    char *first = "place1";
    String name1(first), name2("place2 "), name3("place3"), s1, s2;
    s1.join(name1, name2);
    s2.join(s1, name3);
    name1.display();
    name2.display();
    name3.display();
    s1.display();
    s2.display();
    return 0;
}

```

Input/Output:

```

String name1(first), name2("place2 "), name3("place3"), s1, s2;
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out
place1
place2
place3
place1place2
place1place2 place3
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$

```

9. Write a Program to design a class to represent a matrix. The class should have the functionality to insert and retrieve the elements of the matrix.

Ans:

Code:

```
#include <iostream>
using namespace std;
class matrix
{
    int **p;
    int d1, d2;
public:
    matrix(int x, int y);
    void get_element(int i, int j, int value)
    {
        p[i][j] = value;
    }
    int &put_element(int i, int j)
    {
        return p[i][j];
    }
};

matrix ::matrix(int x, int y)
{
    d1 = x;
    d2 = y;
    p = new int *[d1];
    for (int i = 0; i < d1; i++)
        p[i] = new int[d2];
}

int main()
{
    int m, n;
    cout << "Enter size of matrix";
    cin >> m >> n;
    matrix A(m, n);
    cout << "Enter Matrix Element row by row:";
    int i, j, value;
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
        {
            cin >> value;
```



```

        A.get_element(i, j, value);
    }
    cout << "\n";
    cout << A.put_element(1, 2);
    return 0;
}

```

Input/Output:

```

0deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out
Enter size of matrix3
3
Enter Matrix Element row by row:1 2 3
4 5 6
7 8 9
6deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ █

```

10. Write a program to design a class representing complex numbers and having the functionality of performing addition & multiplication of two complex numbers using operator overloading.

Ans:

Code:

```

#include<iostream>
using namespace std;

class complex
{
private:
float real, imag;
public:
complex( )
{

```

```

}
complex( float r, float i )
{
    real = r;
    imag = i;
}
void getdata( )
{
    float r,
    i;
    cout << endl << "Enter real and imaginary part ";
    cin >> r >> i;
    real = r;
    imag = i;
}
void setdata()
{
    real = real;
    imag = imag;
}
void displaydata( )
{
    cout << endl << "real = " << real;
    cout << endl << "Imaginary = " << imag;
}
complex operator + (complex c)
{
    complex t;
    t.real = real + c.real;
    t.imag = imag + c.imag;
}

complex operator *( complex c )
{
    complex t;
    t.real = real * c.real - imag * c.imag;
    t.imag = real * c.imag + c.real * imag;
    return t;
}
};

int main( )
{
    complex c1 (2.0, 2.0 ), c2 ( 1.2, -2.5 ), c3, c4;

```

```
c1.setdata();  
c3 = c1 + c2;  
c3.displaydata( );  
c4.getdata( );  
complex c5 ( 2.5, 3.0 ),  
c6;  
c6 = c4 * c5;  
c6.displaydata( );  
complex c7;  
c7 = c1 + c2 * c3;  
c7.displaydata( );  
}
```

Input/Output:

```
10.cpp:28:9: note: candidate expects 0 arguments, 2 provided  
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ g++ 10.cpp  
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out  
  
real = -5.61627e+25  
Imaginary = 4.59121e-41  
Enter real and imaginary part
```

11. Write a Program to overload operators like *, <<, >> using friend function. The following overloaded operators should work for a class vector.

Ans:

Code:

```
#include <iostream>
using namespace std;
const int size = 3;
class vector
{
    int v[size];
public:
    vector();
    vector(int *x);
    friend vector operator*(int a, vector b);
    friend vector operator*(vector b, int a);
    friend istream &operator>>(istream &, vector &);
    friend ostream &operator<<(ostream &, vector &);
};

vector ::vector()
{
    for (int i = 0; i < size; i++)
        v[i] = 0;
}

vector ::vector(int *x)
{
    for (int i = 0; i < size; i++)
        v[i] = x[i];
}

vector operator*(int a, vector b)
{
    vector c;
    for (int i = 0; i < size; i++)
        c.v[i] = a * b.v[i];
    return c;
}

vector operator*(vector b, int a)
{
    vector c;
    for (int i = 0; i < size; i++)
        c.v[i] = b.v[i] * a;
    return c;
}
```

```

istream &operator>>(istream &din, vector &b)
{
    for (int i = 0; i < size; i++)
        din >> b.v[i];
    return (din);
}

ostream &operator<<(ostream &dout, vector &b)
{
    dout << "(" << b.v[0];
    for (int i = 1; i < size; i++)
        dout << "," << b.v[i];
    dout << ")";
    return (dout);
}

int x[size] = {2, 4, 6};
int main()
{
    vector m;
    vector n = x;
    cout << "Enter Elements of vector m" << "\n";
    cin >> m;
    cout << "\n";
    cout << "m=" << m << "\n";
    vector p, q;
    p = 2 * m;
    q = n * 2;
    cout << "\n";
    cout << "p=" << p << "\n";
    cout << "q=" << q << "\n";
    return 0;
}

```

Input/Output:

```
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out
Enter Elements of vector m
1 2 3

m=(1,2,3)

p=(2,4,6)
q=(4,8,12)
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$
```

12. Write a program for developing a matrix class which can handle integer matrices of different dimensions. Also overload the operator for addition, multiplication & comparison of matrices.

Ans:

Code:

```
#include <iostream>
#include <iomanip>
using namespace std;
class matrix
{
    int row, col;
    int *ptr;
public:
    matrix(int r, int c)
    {
        row = r;
        col = c;
        ptr = new int[r * c];
    }
    void getmat()
    {
        int i, j, mat_off, temp;
        cout << endl
              << "enter elements matrix:" << endl;
        for (i = 0; i < row; i++)
        {
            for (j = 0; j < col; j++)
```

```

        {
            mat_off = i * col + j;
            cin >> ptr[mat_off];
        }
    }
}

void printmat()
{
    int i, j, mat_off;
    for (i = 0; i < row; i++)
    {
        cout << endl;
        for (j = 0; j < col; j++)
        {
            mat_off = i * col + j;
            cout << setw(3) << ptr[mat_off];
        }
    }
}

int delmat()
{
    matrix q(row - 1, col - 1);
    int sign = 1, sum = 0, i, j, k, count;
    int newsize, newpos, pos, order;
    order = row;
    if (order == 1)
    {
        return (ptr[0]);
    }
    for (i = 0; i < order; i++, sign *= -1)
    {
        for (j = 1; j < order; j++)
        {
            for (k = 0, count = 0; k < order;
                k++)
            {
                if (k == i)
                    continue;
                pos = j * order + k;
                newpos = (j - 1) * (order - 1) + count;
                q.ptr[newpos] = ptr[pos];
                count++;
            }
        }
    }
}

```

```

        sum = sum + ptr[i] * sign * q.delmat();
    }
    return (sum);
}

matrix operator+(matrix b)
{
    matrix c(row, col);
    int i, j, mat_off;
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            mat_off = i * col + j;
            c.ptr[mat_off] = ptr[mat_off] + b.ptr[mat_off];
        }
    }
    return (c);
}

matrix operator*(matrix b)
{
    matrix c(b.col, row);
    int i, j, k, mat_off1, mat_off2, mat_off3;
    for (i = 0; i < c.row; i++)
    {
        for (j = 0; j < c.col; j++)
        {
            mat_off3 = i * c.col + j;
            c.ptr[mat_off3] = 0;
            for (k = 0; k < b.row; k++)
            {
                mat_off2 = k * b.col + j;
                mat_off1 = i * col + k;
                c.ptr[mat_off3] += ptr[mat_off1] * b.ptr[mat_off2];
            }
        }
    }
    return (c);
}

int operator==(matrix b)
{
    int i, j, mat_off;
    if (row != b.row || col != b.col)
        return (0);
    for (i = 0; i < row; i++)

```



```

        {
            for (j = 0; j < col; j++)
            {
                mat_off = i * col + j;
                if (ptr[mat_off] != b.ptr[mat_off])
                    return (0);
            }
        }
        return (1);
    }
};

int main()
{
    int rowa, cola, rowb, colb;
    cout << endl << "Enter dimensions of matrix A ";
    cin >> rowa >> cola;
    matrix a(rowa, cola);
    a.getmat();
    cout << endl << "Enter dimensions of matrix B";
    cin >> rowb >> colb;
    matrix b(rowb, colb);
    b.getmat();
    matrix c(rowa, cola);
    c = a + b;
    cout << endl << "The sum of two matrices = ";
    c.printmat();
    matrix d(rowa, colb);
    d = a * b;
    cout << endl << "The product of two matrices = ";
    d.printmat();
    cout << endl << "Determinant of matrix a =" << a.delmatt();
    if (a == b)
        cout << endl << "a & b are equal";
    else
        cout << endl << "a & b are not equal";
    return 0;
}
}

```

Input/Output:

```
deep@deep-Inspiron-15-3567: ~/4thSem/OOP/Assignments$ g++ 12.cpp
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out

Enter dimensions of matrix A 3
3

enter elements matrix:
1 2 3
4 5 6
7 8 9

Enter dimensions of matrix B3
3

enter elements matrix:
1 2 3
4 5 6
7 8 9

The sum of two matrices =
2 4 6
8 10 12
```

13. Write a program to overload new/delete operators in a class.

Ans:

Code:

```
#include<iostream>
#include<cstdlib>
using namespace std;
class Employee
{
    string name;
    int age;
public:
    Employee()
    {
        cout<< "Constructor is called\n" ;
    }
    Employee(string name, int age)
    {
        this->name = name;
        this->age = age;
    }
    void display()
    {
```

```

        cout<< "Name:" << name << endl;
        cout<< "Age:" << age << endl;
    }
    void * operator new(size_t size)
    {
        cout<< "Overloading new operator with size: " << size << endl;
        void * p = ::operator new(size);

        return p;
    }
    void operator delete(void * p)
    {
        cout<< "Overloading delete operator " << endl;
        free(p);
    }
};
int main()
{
    Employee * p = new Employee("Deepjyoti", 24);
    p->display();
    delete p;
}

```

Input/Output:

```

deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ g++ 13.cpp
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ ./a.out
Overloading new operator with size: 40
Name:Deepjyoti
Age:24
Overloading delete operator
deep@deep-Inspiron-15-3567:~/4thSem/OOP/Assignment3$ █

```