Object Oriented Programming Lab

Assignment 8

Submitted by:

Navdeep Singh

7th October 2025

Roll No: 24124073
Group: 3
Branch: Information Technology
Year: 2nd Year

Q1. Program to Overload Uniary Operator

Code

```cpp
#include<bits/stdc++.h>
using namespace std;

class Number{
    int x;
    public:
    Number(){}
    Number(int val){
        x = val;
    }
    Number operator -(){ // unary operator overloading
        Number temp;
        temp.x = -x;
        return temp;
    }
    void display(){
        cout<<x<<endl;
    }
    friend ostream& operator<<(ostream& os, const Number& n);
    friend istream& operator>>(istream& is, Number& n);
};


ostream& operator<<(ostream& os, const Number& n) {
    os << n.x;
    return os;
}

istream& operator>>(istream& is, Number& n) {
    is >> n.x;
    return is;
}

int main(){
    Number n1(5),n2;
    n2 = -n1; // unary operator   not using n1.x = -n2.x
    cout<<n1<<endl; // using friend function
    cout<<n2<<endl; // using friend function
    return 0;
}
```

Sample Output

```
5
-5
```

Q2. Program to show Hybrid Inheritance

Code

```cpp
class A{
```

```
    public:
    void showA(){
        cout<<"Class A"<<endl;
    }
};

class B:public A{
    public:
    void showB(){
        cout<<"Class B"<<endl;
    }
};

class C: public A{
    public:
    void showC(){
        cout<<"Class C"<<endl;
    }
};

class D:public B,public C{
    public:
    void showD(){
        cout<<"Class D"<<endl;
    }
};

int main(){
    D obj;
    // obj.showA(); // ambiguity error
    obj.B::showA(); // to resolve ambiguity
    obj.C::showA(); // to resolve ambiguity
    obj.showB();
    obj.showC();
    obj.showD();
    return 0;
}
```

Sample Output

```
Class A
Class A
Class B
Class C
Class D
```