```
//Overloading unary minus with operator member function
#include<iostream>
using namespace std;
class Number
{
private:
int x,y,z;
public:
Number(int n,int n1,int n2)
{
x=n;
y=n1;
z=n2;
}
void operator-()
{
x=-x;
y=-y;
z=-z;
}
void show_data()
{
cout<<"\n x="<<x<<"\n";
cout<<"\n y="<<y<<"\n";
cout<<"\n z="<<z<<"\n";
}
};
int main()
{
Number N(7,8,9);
N.show_data();
-N;//Indirect way of calling operator member function
//N.operator-();//Direct way of calling operator member function
N.show_data();
}
#include<iostream>
using namespace std;
class Number
{
private:
int x,y,z;
public:
Number(int n,int n1,int n2)
{
x=n;
y=n1;
z=n2;
cout<<"\n Before overloading:";
cout<<x<<endl;
```

```
cout<<y<<endl;
cout<<z<<endl;
}
friend void operator-(Number);
};
void operator-(Number obj)
{
cout<<"After overloading:\n"<<-obj.x<<endl;
cout<<-obj.y<<endl;
cout<<-obj.z;
}
int main()
{
Number N(7,8,9);
-N;
//operator-(N);
}
#include<iostream>
using namespace std;
class Number
{
private:
int x,y,z;
public:
Number(int n,int n1,int n2)
{
x=n;
y=n1;
z=n2;
cout<<"\n Before overloading:";
cout<<x<<endl;
cout<<y<<endl;
cout<<z<<endl;
}
void show_data()
{
cout<<"\n x="<<x<<"\n";
cout<<"\n y="<<y<<"\n";
cout<<"\n z="<<z<<"\n";
}
friend void operator-(Number&);//By reference
};
void operator-(Number &obj)//obj will become alias for passed object N
{
obj.x=-obj.x;
obj.y=-obj.y;
obj.z=-obj.z;
}
int main()
```

```
{
Number N(7,8,9);
-N;
N.show_data();
//operator-(N);
}
#include<iostream>
using namespace std;
class Number
{
int x;
public:
Number(int n)
{
x=n;
}
void operator++()
{
cout<<"\nPrefix increment:";
cout<<++x<<endl;
}
friend void operator--(Number);
};
void operator--(Number N)
{
cout<<"\nPrefix decrement:";
cout<<--N.x<<endl;
}
int main()
{
Number N1(10),N2(20);
++N1;
//N1.operator++();
--N2;
//operator--(N2);
return 0;
}
#include<iostream>
using namespace std;
class Number
{
int x;
public:
Number(int n)
{
x=n;
}
void operator++(int)
{
```

```
cout<<"\nPostfix increment:";
cout<<x++;
}
friend void operator--(Number,int);
};
void operator--(Number obj1,int)
{
cout<<"\nPostfix decrement:";
cout<<obj1.x--;
}
int main()
{
Number N1(10);
N1++;
//N1.operator++(0);//Member function//Instead of 0 any integer can be passed(no difference)
N1--;
//operator--(N1,0);//Friend function
return 0;
}
#include<iostream>
using namespace std;
class Number
{
private:
int x,y;
public:
Number(int n,int n1)
{
x=n;
y=n1;
cout<<"\n Before overloading:";
cout<<x<<endl;
cout<<y<<endl;
}
friend void operator!(Number);
};
void operator!(Number obj)
{
cout<<"After overloading:\n"<<!obj.x<<endl;
cout<<!obj.y<<endl;

}
int main()
{
Number N(7,8);
!N;
//operator-(N);
}
```