Object Oriented Programming Laboratory

Assignment – 5

Submitted By:

Yishap Khanal

21052960

CSE-34

Date: 20th October,2022

Question 1:

```cpp
// Write a program to demonstrate hybrid inheritance.

#include <iostream>
using namespace std;

class Base
{
    public:
        int a;
};

class Derived1 : virtual public Base
{
    public:
        int b;
};

class Derived2 : virtual public Base
{
    public:
        int c;
};

class Derived3 :public Derived2, public Derived1
{
    public:
        int total;
};

int main()
{
    Derived3 obj;
    obj.a = 10;
    obj.b = 20;
    obj.c = 30;
    obj.total = obj.a + obj.b + obj.c;
    cout << "Total = " << obj.total << endl;
    return 0;
}
```

Output:

```
PS C:\Users\KIIT01\Desktop\programming> cd "c:\Users\KII
; if ($?) { .\q1_hybrid }
Total = 60
PS C:\Users\KIIT01\Desktop\programming\OOP\20 october>
```

Question 2:

```cpp
// Write a program to demonstrate multiple inheritance.

#include <iostream>
using namespace std;

class Base1
{
    public:
        int a;
};

class Base2
{
    public:
        int b;
};

class Derived : public Base1, public Base2
{
    public:
        int total;
};

int main()
{
    Derived obj;
    obj.a = 10;
    obj.b = 20;
    obj.total = obj.a + obj.b;
    cout << "Total = " << obj.total << endl;
    return 0;
}
```

Output:

```
PS C:\Users\KIIT01\Desktop\programming> cd "c:\Users\KI
?) { .\q4_multiple_inhrtnc }
Total = 30
PS C:\Users\KIIT01\Desktop\programming\OOP\20 october>
```

Question 3:

```cpp
// Write a program to demonstrate multilevel inheritance.

#include <iostream>
using namespace std;

class Base
{
    public:
        int a;
};

class Derived1 : public Base
{
    public:
        int b;
};

class Derived2 : public Derived1
{
    public:
        int total;
};

int main()
{
    Derived2 obj;
    obj.a = 10;
    obj.b = 20;
    cout << "a = " << obj.a << endl;
    cout << "b = " << obj.b << endl;
    obj.total = obj.a + obj.b;
    cout << "Total = " << obj.total << endl;
    return 0;
}
```

Output:

```
PS C:\Users\KIIT01\Desktop\programming> cd "c:\Users\KII
q5_multilevel_inhrtnc } ; if ($?) { .\q5_multilevel_inhr
a = 10
b = 20
Total = 30
PS C:\Users\KIIT01\Desktop\programming\OOP\20 october>
```

Question 4:

```cpp
// Write a program to demonstrate ++ unary operator overloading.

#include <iostream>
using namespace std;

class Test{
    int a,b,c;
    public:
        void read_data(int x, int y, int z){
            a = x;
            b = y;
            c = z;
        }
        void operator ++(){
            a++;
            b++;
            c++;
        }
        void display(){
            cout << "a = " << a << endl;
            cout << "b = " << b << endl;
            cout << "c = " << c << endl;
        }
};
```

```cpp
int main(){
    Test obj;
    obj.read_data(10,20,30);
    cout<<"Before incrementing: "<<endl;
    obj.display();
    ++obj;
    cout<<"After incrementing: "<<endl;
    obj.display();
    return 0;
}
```

Output:

```
PS C:\Users\KIIT01\Desktop\programming> cd "c:\Users\KIIT01
_operator_ovrldng } ; if ($?) { .\q2_operator_ovrldng }
Before incrementing:
a = 10
b = 20
c = 30
After incrementing:
a = 11
b = 21
c = 31
PS C:\Users\KIIT01\Desktop\programming\OOP\20 october>
```

Question 5:

```cpp
// Write a program to demonstrate * binary operator overloading.

#include <iostream>
using namespace std;

class Test{
    int a,b,c;
    public:
        void read_data(int x, int y, int z){
            a = x;
            b = y;
            c = z;
        }
        void operator *(Test obj){
            a = a * obj.a;
            b = b * obj.b;
            c = c * obj.c;
        }
        void display(){
            cout << "a = " << a << endl;
            cout << "b = " << b << endl;
            cout << "c = " << c << endl;
        }
};
```

```cpp
int main(){
    Test obj1, obj2;
    obj1.read_data(10,20,30);
    obj2.read_data(2,3,4);
    cout<<"Before multiplication: "<<endl;
    obj1.display();
    obj2.display();
    obj1 * obj2;
    cout<<"After multiplication: "<<endl;
    obj1.display();
    return 0;
}
```

Output:

```
PS C:\Users\KIIT01\Desktop\programming> cd "c:\Users\KIIT01\Desktop\programming\OO
f ($?) { .\q3_binary_overloading }
Before multiplication:
a = 10
b = 20
c = 30
a = 2
b = 3
c = 4
After multiplication:
a = 20
b = 60
c = 120
PS C:\Users\KIIT01\Desktop\programming\OOP\20 october>
```