Term work

of

OOPs With C++ Lab (PCS-307)

Submitted in partial fulfillment of the requirement for the III semester

Bachelor of Technology

By

Aniruddh Joshi

University Roll No

2161076

Under the Guidance of

Mr. Ravindra Koranga

Assistant Professor

Deptt. of CSE

Graphic Era
HILL UNIVERSITY
University under section 2(f) of UGC Act, 1956
Bhimtal Campus

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS

SATTAL ROAD, P.O. BHOWALI

DISTRICT- NAINITAL-263132

2022-2023

CERTIFICATE

The term work of OOPs with C++ Lab (PCS-307), being submitted by……. d/o s/o……. , Enrollment no…… , Roll no……. to Graphic Era Hill University Bhimtal Campus for the award of bona fide work carried out by him/her. He/She has worked under my guidance and supervision and fulfilled the requirement for the submission of this work report.

(……) **(……)**

Subject Professor **HOD, CSE Dept.**

ACKNOWLEDGEMENT

I take immense pleasure in thanking Honorable **Mr. Ravindra Koranga** (Assistant Professor, CS, GEHU Bhimtal Campus) for allowing us to carry out this project work under his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance and useful suggestions that have helped me in developing my subject concepts as a student.

I want to extend thanks to our President **"Prof. (Dr.) Kamal Ghanshala"** for providing us all infrastructure and facilities to work in need without which this work would not be possible.

-ANIRUDDH JOSHI

aniruddh.joshi2904@gmail.com

STUDENT'S DECLARATION

I, **Aniruddh Joshi**, hereby declare the work, which is being presented in the report, entitled **Term work of OOPs with C++ Lab (PCS-307)** in partial fulfillment of the requirement for the award of the degree **Bachelor of Technology (Computer Science)** in the session **2022-2023** for semester III, is an authentic record of my own work carried out under the supervision of **Mr. Ravindra Koranga** (Graphic Era Hill University, Bhimtal)

The matter embodied in this project has not been submitted by me for the award of any other degree.

Date: …….. …….

(Full signature of student)

Graphic Era
HILL UNIVERSITY
University under section 2(f) of UGC Act, 1956
Bhimtal Campus

Computer Science and Engineering Department
OOPS LAB (PCS-307)

Requirements:

- Unix/Linux based Computer System

Index/List of Experiments

1	WAP in C++ to show the use of inline function
2	WAP in C++ to demonstrate Function Overloading
3	WAP in C++ in which static data member maintain values common to the entire class.
4	WAP in C++ to demonstrate the concept of friend function
5	WAP in C++ to demonstrate the concept of parameterized constructor
6	WAP in C++ to show how unary minus operator is overloaded
7	WAP in C++ to demonstrate the concept of Multiple Inheritance
8	WAP in C++ that shows the ambiguity resolution in inheritance through virtual base class
9	WAP in C++ to demonstrate the use of virtual function
10	WAP in C++ to create a file and perform write , read and update operation on file
11	WAP in C++ to demonstrate Exception Handling by using try and catch block
12	WAP in C++ to create a Linked List using STL. Sort this list using sort algorithm in STL.

1-WAP in C++ to show the use of inline function

```
#include<iostream>
using namespace std;
inline int mul(int x, int y)
{
	int prod;
	prod=x*y;
	return prod;
}

int main()
{
	int a,b;
	cout<<"Enter two numbers "<<endl;
	cin>>a>>b;
	cout<<"The product is "<<mul(a,b)<<endl;

}
```

OUTPUT:

```
Enter two numbers
5
10
The product is 50
```

2-WAP in C++ to demonstrate Function Overloading

```
#include<iostream>
using namespace std;
int add(int x, int y)
{
	int sum=x+y;
	return sum;
}
int add(int x, int y, int z)
{
	int sum=x+y+z;
	return sum;
}

int main()
{
	int a,b,c;
	cout<<"Enter two numbers "<<endl;
	cin>>a>>b;
	cout<<"The sum is "<<add(a,b)<<endl;
	cout<<"Enter three numbers "<<endl;
	cin>>a>>b>>c;
	cout<<"The sum is "<<add(a,b,c)<<endl;

}
```

OUTPUT:

```
Enter two numbers
10
20
The sum is 30
Enter three numbers
10
20
30
The sum is 60
```

3-WAP in C++ in which static data member maintain values common to the entire class.

```
#include<iostream>
using namespace std;
class student
{
    private:
        int roll, marks1, marks2;
        static int count;
    public:
        void set(int x, int y, int z);
        void get();
        void add();
        student();
        ~student();
};
int student::count=0;
void student::set(int x, int y, int z)
{
    roll=x;
    marks1=y;
    marks2=z;
}
void student::get()
{
    cout<<endl<<"Printing Student Details : "<<endl;
    cout<<"Roll no.: "<<roll<<" Marks1: "<<marks1<<" Marks2: "<<marks2<<endl;

}
void student::add()
{
    cout<<"Total marks is "<<marks1+marks2<<endl<<endl;
}
student::student()
{
    count++;
    cout<<"Object is created "<<endl;
    cout<<"No. of objects = "<<count<<endl;
}
student::~student()
{
    count--;
    cout<<"Object is destroyed "<<endl;
    cout<<"No. of objects = "<<count<<endl;
}
int main()
{
    student obj1;
    obj1.set(1,10,20);
```

```
    obj1.get();
    obj1.add();
    student obj2;
    obj2.set(2,40,40);
    obj2.get();
    obj2.add();


}
```

OUTPUT:

```
Object is created
No. of objects = 1

Printing Student Details :
Roll no.: 1 Marks1: 10 Marks2: 20
Total marks is 30

Object is created
No. of objects = 2

Printing Student Details :
Roll no.: 2 Marks1: 40 Marks2: 40
Total marks is 80

Object is destroyed
No. of objects = 1
Object is destroyed
No. of objects = 0
```

4-WAP in C++ to demonstrate the concept of friend function

```
#include<iostream>
using namespace std;
class student
{
    private:
        int roll, marks1, marks2;
    public:
        void set(int x, int y, int z);
        void get();
        friend void add(student po);

};
void student::set(int x, int y, int z)
{
    roll=x;
    marks1=y;
    marks2=z;
}
void student::get( )
{
    cout<<"Printing student details : "<<endl;
    cout<<"Roll no.: "<<roll<<"   Marks1: "<<marks1<<"   Marks2: "<<marks2<<endl;

}
void add(student po)
{
    cout<<"Total marks is "<<po.marks1+po.marks2<<endl;
}
int main()
{
    student obj1;
    obj1.set(1,90,90);
    obj1.get();
    add(obj1);
}
```

OUTPUT:

```
Printing student details :
Roll no.: 1   Marks1: 90   Marks2: 90
Total marks is 180
```

5-WAP in C++ to demonstrate the concept of parameterized constructor

```
#include<iostream>
using namespace std;
class student
{
    private:
        int roll, marks1, marks2;
    public:
        void set(int x, int y, int z);
        void get();
        void add();
        student(int x, int y, int z);

};
void student::set(int x, int y, int z)
{
    roll=x;
    marks1=y;
    marks2=z;
}
void student::get()
{
    cout<<"Printing Student Details: "<<endl;
    cout<<"Roll no.: "<<roll<<"   Marks1: "<<marks1<<"   Marks2: "<<marks2<<endl;
}
void student::add()
{
    cout<<"Total marks is "<<marks1+marks2<<endl;
}
student::student(int x, int y, int z)
{
    cout<<"Parameterized Constructor is called "<<endl;
    roll=x;
    marks1=y;
    marks2=z;
}

int main()
{
    student obj1(1,50,60);
    obj1.get();
    obj1.add();
}
```

OUTPUT:

```
Parameterized Constructor is called
Printing Student Details:
Roll no.: 1   Marks1: 50   Marks2: 60
Total marks is 110
```

6-WAP in C++ to show how unary minus operator is overloaded

```
#include<iostream>
using namespace std;
class student
{
    private:
        int roll, marks1, marks2;
    public:
        void set(int x, int y, int z);
        void get();
        void add();
        student operator-(student po);
};
void student::set(int x, int y, int z)
{
    roll=x;
    marks1=y;
    marks2=z;
}
void student::get()
{
    cout<<"Printing Student Details: "<<endl;
    cout<<"Roll no.: "<<roll<<"   Marks1: "<<marks1<<"   Marks2: "<<marks2<<endl;

}
void student::add()
{
    cout<<"Total marks is "<<marks1+marks2<<endl;
}
student student::operator-(student po)
{
    student temp;
    temp.roll=3;
    temp.marks1 = marks1-po.marks1;
    temp.marks2 = marks2-po.marks2;
    return temp;
}
int main()
{
    student obj1, obj2;
    obj1.set(1,50,60);
    obj1.get();
    obj1.add();

    obj2.set(2,40,50);
    obj2.get();
    obj2.add();

    student obj3;
    cout<<"Create obj3. It will contain difference of obj1 and obj2 "<<endl;
```

```
        obj3=obj1-obj2;
        obj3.get();


}
```

OUTPUT:

```
Printing Student Details:
Roll no.: 1   Marks1: 50   Marks2: 60
Total marks is 110
Printing Student Details:
Roll no.: 2   Marks1: 40   Marks2: 50
Total marks is 90
Create obj3. It will contain difference of obj1 and obj2
Printing Student Details:
Roll no.: 3   Marks1: 10   Marks2: 10
```

7-WAP in C++ to demonstrate the concept of Multiple Inheritance

```
#include<iostream>
using namespace std;
class A
{
    protected:
        int i,j;
    public:
        void add()
        {
            cout<<i+j;
        }

};
class B
{
    protected:
        int k,l;
    public:
        void mul()
        {
            cout<<k*l;
        }
};
class C:public A, public B
{

    public:
        void set(int w, int x, int y, int z)
        {
            i=w;
            j=x;
            k=y;
            l=z;
        }
        void get()
        {
            cout<<"The values of four variables are :"<<endl;
            cout<<i<<"  "<<j<<"  "<<k<<"  "<<l<<"  "<<endl;
        }
        void add()
        {
            cout<<"The sum of four variables is "<<i+j+k+l<<endl;
        }
        void mul()
        {
            cout<<"The product of four variables is "<<i*j*k*l<<endl;
        }
};
int main()
```

```
{
    C obj;
    obj.set(10,20,30,40);
    obj.get();
    obj.add();
    obj.mul();


}
```

OUTPUT:

```
The values of four variables are :
10  20  30  40
The sum of four variables is 100
The product of four variables is 240000
```

8-WAP in C++ that shows the ambiguity resolution in inheritance through virtual base class

```
#include<iostream>
using namespace std;
class A
{
    protected:
        int i,j;
    public:
        void add()
        {
            cout<<"The sum is ";
            cout<<i+j<<endl;
        }

};
class B: virtual public A
{
    protected:

    public:
        void mul()
        {
            cout<<"The product is ";
            cout<<i*j<<endl;
        }
};
class C: virtual public A
{
    protected:

    public:
        void div()
        {
            cout<<i/j;
        }
};
class D:public B, public C
{

    public:
        void set(int x, int y)
        {
            i=x;
            j=y;
        }
        void get()
        {
            cout<<"The values of two variables are :"<<endl;
            cout<<i<<"  "<<j<<"  "<<endl;
```

```
        }

};
int main()
{
    D obj;
    obj.set(10,20);
    obj.get();
    obj.add();
    obj.mul();


}
```

OUTPUT:

```
The values of two variables are :
10  20
The sum is 30
The product is 200
```

9-WAP in C++ to demonstrate the use of virtual function

```
#include<iostream>
using namespace std;
class A
{
    protected:
        int i,j;
    public:
        virtual void add()
        {
            cout<<"Sum of two numbers is "<<i+j<<endl;
        }
        void set(int x , int y)
        {
            i=x;
            j=y;
        }
};
class B:public A
{
    protected:
        int k;
    public:
        void add()
        {
            cout<<"Sum of three numbers is "<<i+j+k<<endl;
        }
        void set(int x , int y, int z)
        {
            i=x;
            j=y;
            k=z;

        }
};
class C: public B
{
    protected:
        int l;
    public:
        void add()
        {
            cout<<"Sum of four numbers is "<<i+j+k+l<<endl;
        }
        void set(int w, int x , int y, int z)
        {
            i=w;
            j=x;
            k=y;
            l=z;
```

```
        }
};
int main()
{
    A obja;
    B objb;
    C objc;

    obja.set(10,20);
    objb.set(10,20,30);
    objc.set(10,20,30,40);

    A &ref1 = obja;
    cout<<"Calling virtual function through Base Class Reference to object of Class
A"<<endl;
    ref1.add();

    A &ref2=objb;
    cout<<"Calling virtual function through Base Class Reference to object of Class
B"<<endl;
    ref2.add();

    A &ref3=objc;
    cout<<"Calling virtual function through Base Class Reference to object of Class
C"<<endl;
    ref3.add();

}
```

OUTPUT:

```
Calling virtual function through Base Class Reference to object of Class A
Sum of two numbers is 30
Calling virtual function through Base Class Reference to object of Class B
Sum of three numbers is 60
Calling virtual function through Base Class Reference to object of Class C
Sum of four numbers is 100
```

10-WAP in C++ to create a file and perform write , read and update operation on file

```
#include<iostream>
#include<fstream>
#include<string>
using namespace std;
int main()
{
    ofstream myout("file.txt", ios::app);
    if(!myout)
    {
        cout<<"Can't open file "<<endl;
        return 0;
    }

    char text[255];
    cout<<"Enter text to write to file"<<endl;
    cin.getline(text, sizeof(text));

    cout<<"Writing text to file"<<endl;
    char *c;
    c=text;
    myout<<endl;
    while(*c != NULL)
    {
        myout.put(*c);
        c++;
    }
    myout.close();

    ifstream myin("file.txt");
    cout<<"Reading text from file "<<endl;
    myin.read(text, sizeof(text));
    cout<<text<<endl;
    myin.close();

    cout<<endl<<endl<<"Updating the file"<<endl;
    cout<<"Secify the position at which you want to updat the character in file"<<endl;
    int pos;
    cin>>pos;
    cout<<"Secify the new character at this position"<<endl;
    char ch;
    cin>>ch;

    myout.open("file.txt",ios::in|ios::out);
    myout.seekp(pos, ios::beg);
    myout.put(ch);
    myout.close();

    myin.open("file.txt");
```

```
    cout<<"Reading text from file "<<endl;
    myin.read(text, sizeof(text));
    cout<<text<<endl;
    myin.close();
}
```

OUTPUT:

```
Enter text to write to file
Happi New Year
Writing text to file

Reading text from file
Happi New Year

Updating the file
Secify the position at which you want to updat the character in file
6
Secify the new character at this position
Y

Reading text from file
HappY New Year
```

11-WAP in C++ to demonstrate Exception Handling by using try and catch block

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    cout<<"Enter a number"<<endl;
    cin>>a;
    try
    {
        if(a==0)
            throw 100;
        cout<<"5000 divided by a is "<<5000/a<<endl;
    }
    catch(int x)
    {
        cout<<"Can't Divide by zero"<<endl;
        cout<<"Error code is "<<x<<endl;
    }

}
```

OUTPUT:

```
Enter a number
50
5000 divided by a is 100
```

12-WAP in C++ to create a Linked List using STL. Sort this list using sort algorithm in STL.

```
#include<iostream>
#include<list>
#include<algorithm>
#include<iterator>
using namespace std;
int main(){
    list<int> l;
    cout<<"How many elements do you want in the list"<<endl;
    int n;
    cin>>n;
    cout<<"Enter the elements"<<endl;
    int i,x;
    for(i=0;i<n;i++)
    {
        cin>>x;
        l.push_back(x);
    }
    cout<<"Printing the list"<<endl;
    list<int>::iterator itr;
    itr=l.begin();
    while(itr!=l.end())
    {
        cout<<*itr<<"-> ";
        itr++;
    }
    cout<<endl<<"Sorting the list"<<endl;
    l.sort();
    cout<<"The sorted list is"<<endl;
    itr=l.begin();
    while(itr!=l.end())
    {
        cout<<*itr<<"-> ";
        itr++;
    }
}
```

OUTPUT:

```
How many elements do you want in the list
5
Enter the elements
1 2 3 4
5
Printing the list
1-> 2-> 3-> 4-> 5->
Sorting the list
The sorted list is
1-> 2-> 3-> 4-> 5->
```