

Group A

Assignment 4

Problem Statement

Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float). Each of these three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display its data. Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata(), and then displaying the data with putdata().

Objectives:

To learn the concept of Class, Object, Data Hiding, Inheritance

Outcomes:

The performer will be able to design a program in c++ using concept Class, Object, Data Hiding, Inheritance

Software Requirements:

1. 64-bit Open source Linux or its derivative
2. Open Source C++ Programming tool like G++/GCC.

Algorithm:

1. Design a Base Class publication which contains fields to store title and price..
2. Design two derived classes to hold properties of Tape and Book respectively.
3. Create functions for getting data and putting data in the base class as well as in the derived class.
4. Create main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata(), and then displaying the data with putdata().

Program

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

class publication
{
    char title[10];
    float price;

public:
    void read()
    {
        cout<<"\nEnter Title of the Publication :";
        cin>>title;
        cout<<"\nEnter the price of Publication :";
        cin>>price;
    }
}
```

```

        void show()
        {
            cout<<"Title is : "<<title<<"\n";
            cout<<"Price is : "<<price<<"\n";
        }
};

class book :public publication
{
    int page_count;

    public :

        void read()
        {
            cout<<"\nEnter the page count of Book : ";
            cin>> page_count;
        }
        void show()
        {
            cout<<"Page count of Book is : "<< page_count<<"\n";
        }
};

class tape :public publication
{
    float playing_time;

    public :

        void read()
        {
            cout<<"\nEnter playing time in minutes : ";
            cin>>playing_time;
        }
        void show()
        {
            cout<<"Playing time in minutes : "<<playing_time<<"\n";
        }
};

int main()
{
    publication p;

```

```
p.read() ;  
p.show() ;  
  
book b;  
b.read() ;  
b.show() ;  
  
tape t;  
t.read() ;  
t.show() ;  
  
}
```

Output

Enter Title of the Publication :arihant

Enter the price of Publication :500

Title is :arihant

Price is :500

Enter the page count of Book :399

Page count of Book is :399

Enter playing time in minutes :42

Playing time in minutes :42

Problem Statement

Create employee bio-data using following classes i) Personal record ii) Professional record iii) Academic record Assume appropriate data members and member function to accept required data & print bio-data. Create bio-data using multiple inheritance using C++

Objectives:

To learn the concept of multiple inheritance in c++ programming

Outcomes:

The performer will be able to design a program in c++ using multiple inheritance .

Software Requirements:

1. 64-bit Open source Linux or its derivative
2. Open Source C++ Programming tool like G++/GCC

Algorithm:

1. Design a class personal with display function to display personal details
 2. Design a class academics with display function to display academic details
 3. Design a class professional with display function to display professional details
 4. Design a class biodata which inherits from the classes personal, academics and professional
- class biodata:public personal,public academics,public professional
5. Create a function to accept data in class biodata
 6. Create object b of biodata to accept and display data
1. It

Program:

```
#include<iostream>
using namespace std;
class personal
{
protected:
    char name[50];
    char address[50];
    char birthdate[50];
    char gender;
public:
    void get_personal();
};
class professional
{
protected:
    int noofyearsexp;
    char orgname[50];
    char projname[50];
    char projdetails[50];
public:
    void get_professional();
};
```

```

};

class academic
{
protected:
    int year;
    int marks;
    int percentage;
    char Class[50];
public:
    void get_academic();
};

class biodata: public personal,public academic,public professional
{
public: void display();
};

void personal::get_personal()
{
    cout<<"Enter name ";
    cin>>name;
    cout<<"\nEnter Address::";
    cin>>address;
    cout<<"\nEnter Birthdate(dd/mm/yyyy)::";
    cin>>birthdate;
    cout<<"\nEnter gender(M/F)::";
    cin>>gender;
}

void professional::get_professional()
{
    cout<<"\nEnter number of years of exp::";
    cin>>noofyearsexp;
    cout<<"\nEnter organization name::";
    cin>>orgname;
    cout<<"\nEnter project name::";
    cin>>projname;
    cout<<"\nEnter project Details::";
    cin>>projdetails;
}

void academic::get_academic()
{
    cout<<"\nEnter academic year::";
    cin>>year;
}

```

```

cout<<"\nEnter total marks::";
cin>>marks;
cout<<"\nEnter percentage::";
cin>>percentage;
cout<<"\nEnter class::";
cin>>Class;
}

void biodata::display()
{

    cout<<"Employee Biodata"<<endl;
    cout<<"Personal Details"<<endl;
    cout<<"Name::"<<name<<endl;
    cout<<"address::"<<address<<endl;
    cout<<"birthdate::"<<birthdate<<endl;
    cout<<"Gender::"<<gender<<endl;
    cout<<"\nAcademic Details"<<endl;
    cout<<"Academic Year "<<"marks "<<"percentage "<<"class "<<endl;
    cout<<year<<"\t\t0"<<marks<<"\t"<<percentage<<"\t"<<Class<<endl;
    cout<<"Professional Details"<<endl;
    cout<<"\nOrganization Name::"<<orgname;
    cout<<"\nYears of Experince::"<<noofyearsexp;
    cout<<"\nProject Done::"<<projname;
    cout<<"\nProject Details::"<<projdetails;
}

int main()
{
    biodata b;
    b.get_personal();
    b.get_professional();
    b.get_academic();
    b.display();
}

```

Output:

```

PS C:\Users\Admin\OneDrive\C++\OOPS_Assignment> cd
"c:\Users\Admin\OneDrive\C++\OOPS_Assignment\" ; if ($?) { g++ assignment2.cpp -o assignment2 } ; if
($?) { .\assignment2 }
Enter name Sangram

Enter Address::Karad

```

Enter Birthdate(dd/mm/yyyy)::18/05/2002

Enter gender(M/F)::M

Enter number of years of exp::5

Enter organization name::DYPIEMR

Enter project name::Flask

Enter project Details::website

Enter academic year::2021

Enter total marks::497

Enter percentage::97.98

Enter class::Employee Biodata

Personal Details

Name::Sangram

address::Karad

birthdate::18/05/2002

Gender::M

Academic Details

Academic Year marks percentage class

2021	0497	97	.98
------	------	----	-----

Professional Details

Organization Name::DYPIEMR

Years of Experince::5

Project Done::Flask

Project Details::website

PS C:\Users\Admin\OneDrive\C++\OOPS_Assignment>

Assignment 3

Problem Statement

Implement a class Complex which represents the Complex Number data type. Implement the following operations:

1. Constructor (including a default constructor which creates the complex number $0+0i$).
2. Overloaded operator+ to add two complex numbers.
3. Overloaded operator* to multiply two complex numbers.
4. Overloaded << and >> to print and read Complex Numbers.

Objectives:

1. Understand the concept of constructors and how to design it.
2. Understand the OOP concept of operator overloading.

Outcomes:

1. Will be able to design constructors for classes designed in programming.
2. Will be able to program using operator functions.

Software Requirements:

1. 64-bit Open source Linux or its derivative
2. Open Source C++ Programming tool like G++/GCC.

Algorithm

1. Design a class Complex which represents the Complex Number data type.
2. Create a constructor which creates the complex number $0+0i$

```
Complex(): real(0), imag(0){ }
```

3. Use operator overloading to overload operator+ to add two complex numbers.

```
Complex operator + (Complex c2)
```

4. Use operator overloading to overload operator* to multiply two complex numbers.

```
Complex operator * (Complex c3)
```

5. Use operator overloading to overload operator<< and >> to print and read Complex Numbers.

```
friend ostream& operator << (ostream &out, Complex &c); //overloading '<<' operator
```

```
friend istream& operator >> (istream &in, Complex &c); //overloading '>>' operator
```

Program

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



```

{
public:
    float real,img;
    complex()
    {
        real=0;
        img=0;
    }
    complex operator +(complex);
    complex operator *(complex);
    friend ostream &operator<<(ostream&,complex&);
    friend istream &operator>>(istream&,complex&);
};

```

```

complex complex::operator +(complex obj)

```

```

{
    complex temp;
    temp.real=real+obj.real;
    temp.img=img+obj.img;
    return (temp);
}

```

```

complex complex::operator *(complex obj)

```

```

{
    complex temp;
    temp.real=(real*obj.real)-(img*obj.img);
    temp.img=(real*obj.img)+(img*obj.img);
    return (temp);
}

```

```

istream &operator>>(istream& is,complex& obj)

```

```

{
    is>>obj.real;
    is>>obj.img;
    return is;
}

```

```

ostream &operator<<(ostream& os,complex& obj)

```

```

{
    os<<obj.real;
    os<<"+"<<obj.img<<"i";
}

```

```

        return os;
    }

int main()
{
    complex a,b,c,d;
    //cout<<"\n Enter first complex number"<<endl;
    cout<<"\n Enter real and imaginary part of first complex number:";
    cin>>a;

    //cout<<"\n Enter second complex number"<<endl;
    cout<<"\n Enter real and imaginary part of second complex number:";
    cin>>b;

    cout<<"\n Arithmetic operations are :";
    c=a+b;
    cout<<"\n Addition is:"<<c;

    d=a*b;
    cout<<"\n Multiplication is:"<<d<<"\n";
    return 0;
}

```

Output

PS C:\Users\Admin\OneDrive\C++\OOPS_Assignment> cd
 "c:\Users\Admin\OneDrive\C++\OOPS_Assignment\" ; if (\$?) { g++ assignment3.cpp -o assignment3 } ; if
 (\$?) { .\assignment3 }

Enter real and imaginary part of first complex number:5
 6

Enter real and imaginary part of second complex number:6
 5

Arithmetic operations are :
 Addition is:11+11i
 Multiplication is:0+36i

PS C:\Users\Admin\OneDrive\C++\OOPS_Assignment>