**Session 8 (unit-5): Inheritance**

**1.       A publishing house maintains its publications in terms of physical copies i.e. the books as well as in terms of virtual copies i.e. the video files. Create a class publication that stores the title and price of the publication. From this class, derive two classes:**

1. **Book; which adds a page count (type int)**
2. **Video ; which adds the size of the file in MBs (type int)**

**Each of these three class must have a getdata() function to get the data from the user and a display() function to display the output. WAP that can well depict the above scenario.**

#include<iostream>

#include<stdio.h>

using namespace std;

class publication

{

private:

char title[25];

int price;

public :

void getdata()

{

cout<<"\nEnter the title of the book \n";

gets(title);

cout<<" \nPrice Of the Book \n";

cin>>price;

}

void display()

{

cout<<"\nTitle Of Book :"<<title<<endl;

cout<<"\nPrice :"<<price<<endl;

}

};

class Book: public publication

{

private:

int pgcount;

public:

void getdata()

{

cout<<"\nEnter the page number of book ";

cin>>pgcount;

}

void display()

{

cout<<"\nNo. of Pages in book: ";

cout<<pgcount;

}

};

class Video : public publication

{

private:

int video;

public:

void getdata()

{

cout<<"\nEnter video memory ";

cin>>video;

}

void display()

{

cout<<"\n Total memory of video: "<< video<<" MB";

}

};

int main()

{

Book b;

Video v;

cout<<"\t Data Input\n";

b.publication::getdata();

b.getdata();

v.getdata();

cout<<"\t Data Output\n";

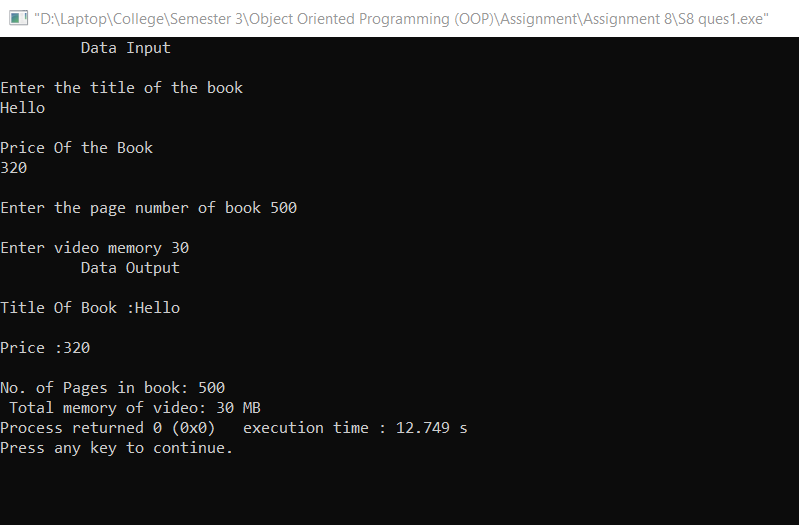
b.publication::display();

b.display();

v.display();

return 0;

}



**2. Depict the following class hierarchy and assume the suitable member functions:**

#include<iostream>

#include<stdio.h>

using namespace std;

class customer

{

protected:

int id,accno,bal,loannno,loanamt;

char name[50];

public:

void getdata()

{

cout<<"\nEnter your Name : ";

gets(name);

cout<<"Enter your Id :";

cin>>id;

}

void display()

{

cout<<"\nCustomer Name: "<<name;

cout<<"\n Customer ID: "<<id<<endl;

}

};

class depositor:public customer

{

int accno,bal;

public:

void getdata()

{

cout<<"\nEnter your account number :";

cin>>accno;

cout<<"\nEnter the money you want to deposit:";

cin>>bal;

}

void display()

{

cout<<"\nAccount Number: "<<accno;

cout<<"\nMoney Deposited: "<<bal;

}

};

class borrower:public depositor

{

int loanno,loanamt;

public:

void getdata()

{

cout<<"\nEnter your loan number:";

cin>>loanno;

cout<<"\nEnter the loan money you want:";

cin>>loanamt;

}

void display()

{

cout<<"\nLoan Number: "<<loanno;

cout<<"\nLoan Amount: "<<loanamt;

}

};

int main()

{

depositor d;

borrower b;

cout<<"\t Data Input";

d.customer::getdata();

d.getdata();

b.getdata();

cout<<"\n\n\t Data Output";

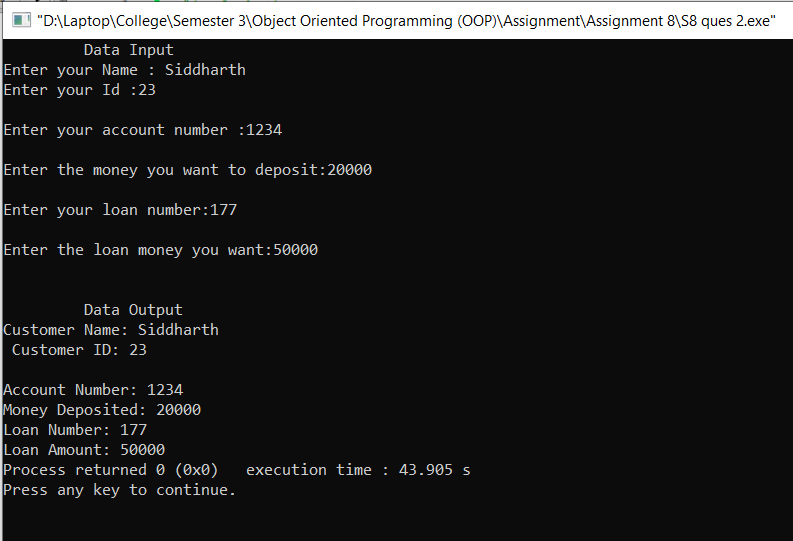
d.customer::display();

d.display();

b.display();

return 0;

}



3. Show with the help of a suitable example code: "**ambiguity problem in multiple inheritance**".

#include<iostream>

using namespace std;

class ClassA

{

public:

int a;

};

class ClassB : virtual public ClassA //a,b

{

public: int b;

};

class ClassC : virtual public ClassA //a,c

{

public:

int c;

};

class ClassD : public ClassB, public ClassC //b,c,a,d

{

public: int d;

};

int main()

{

ClassD obj;

obj.a = 10; //Statement 3

obj.a = 100; //Statement 4

obj.b = 20;

obj.c = 30;

obj.d = 40;

cout<< "\n A : "<< obj.a;

cout<< "\n B : "<< obj.b;

cout<< "\n C : "<< obj.c;

cout<< "\n D : "<< obj.d;

}

