VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Name : Nithish.G

REG no : 19BCS0012

Course : OBJECT ORIENTED PRORAMMING

Course code : CSC2002

Assignment : PROPLEM SOLVING

1. **Define the class to represent a bank account. Include the following members. Data Members: Name of the depositor, Account Number, Type of account and Balance amount in the account.**

<u>Member Functions:</u> To assign initial values, To deposit an amount, To withdraw an amount after checking the balance, To display the name and balance

Write a main program to test the program.

SOURCE CODE:

```
#include<iostream>
#include<string>
using namespace std;
class Bank
{
   public:
      void input();
      void deposit();
      void withdraw();
      void display();
   private:
      string name;
      int acno;
      char type;
      int bal;
};
void Bank:: input()
{
        cout<<"Enter Account Number: ";
        cin>>acno;
        cin.ignore();
        cout<<"Enter Account Holder's Name: ";
        getline(cin,name);
        cout<<"Enter Type of Account, S for Savings and C for Current: ";
        cin>>type;
        cout<<"Enter Balance Amount: ";
        cin>>bal;
}
void Bank::deposit()
{
        int amt;
        cout<<"Enter Amount to be deposited: ";
        cin>>amt;
        bal+=amt;
        cout<<"Available Balance: "<<bal<<endl;
```

```
}
void Bank::withdraw()
{
	int amt;
	cout<<"Available Balance: "<<bal<<endl;
	cout<<"Enter Amount to Withdraw: ";
	cin>>amt;
	if(amt>bal)
		cout<<"Insufficient Balance."<<endl;
	else
		bal-=amt;
	cout<<"Available Balance: "<<bal<<endl;
}
void Bank::display()
{
	cout<<"Account Number: "<<acno<<endl;
	cout<<"Account Holder's Name: "<<name<<endl;
	cout<<"Type of Account: "<<type<<endl;
	cout<<"Balance Amount: "<<bal<<endl;
}
int main()
{
	Bank b;
	int n=0;
	b.input();
	while(n!=3)
	{
		cout<<"1.Deposit\n2.Withdraw\n3.Exit\nEnter Your Choice: ";
		cin>>n;
		if(n==1)
			b.deposit();
		else if(n==2)
			b.withdraw();
		else if(n!=3)
			cout<<"Invalid option, please try again.";
	}
	b.display();
	return 0;
}
```

2. **Create twS classes DM and DB which store the value of distances. DM stores distance in metes and centimeters and DB is feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB.**

   Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required.
   The display should be in the format of feet and inches or meters and centimeters depending on the object on display.

SOURCE CODE:

```
#include<iostream>
using namespace std;
class DB;
class DM
{
        public:
                void input();
                friend void add(DM &,DB &);
                void display();
                int m;
                int c;
};
class DB
{
        public:
                void input();
                friend void add(DM &,DB &);
                int f;
                int i;
};
void DM::input()
{
        cout<<"Enter Distance in Metres: ";
        cin>>m;
        cout<<"Enter Distance in Centimetres: ";
        cin>>c;
}
void DB::input()
{
        cout<<"Enter Distance in Feet: ";
        cin>>f;
        cout<<"Enter Distance in Inches: ";
```

```
        cin>>i;
        if(i>=12)
        {
                f+=i/12;
                i=i%12;
        }
}
void add(DM &a,DB&b)
{
  DM d;
  int c=(a.m*100+a.c+b.f*30.48+b.i*2.54);
  if(c>=100)
  {
d.m=c/100;
d.c=c%100;
  }
  else
  {
d.m=0;
d.c=c;
  }
d.display();
}
void DM::display()
{
        cout<<"Distance in Metres: "<<m<<endl;
        cout<<"Distance in Centimetres: "<<c<<endl;
}
int main()
{
        DM d1;
        DB d2;
        d1.input();
        d2.input();
        add(d1,d2);
        return 0;
}
```

A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title & author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then, the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed. Otherwise, the message "Required copies not in stock" is displayed.

Design a system using a class called Books with suitable member functions and
Constructors. Use "new" operator in constructors to allocate memory space required.

SOURCE CODE:

```
#include<iostream>
using namespace std;
#include<stdlib.h>
#include<string.h>
#include<stdio.h>
class stock
{
        char author[50];
        char title[50];
        char pub[50];
        double price;
        int numcopies;
   public:
stock();
        int access_title(char a[]);
        void input();
        void getdata(int);

};
stock::stock()
{
    char author[50]={"abc"};
    char title[50]={"efg"};
```

```
    char pub[50]={"hij"};
    price=500;
numcopies=50;
}
int stock::access_title(char a[])
{
    if(strcmp(title,a)==0)
        return 1;
}
void stock::getdata(int num)
{
    if(numcopies>=num)
cout<<"\nCost of "<<num<<" books is Rs. "<<(price*num);
    else
cout<<"\nSorry! These many copies are unavailable!";
}
void stock::input()
{
cin.ignore();
            cout<<"\nTitle: ";
    gets(title);
cout<<"\nAuthor:";
    gets(author);
cout<<"\nPublisher:";
    gets(pub);
cout<<"\nPrices:";
cin>>price;
cout<<"\nCopies available:";
cin>>numcopies;
}

int main()
{
  system("cls");

  stock obj[2];
  int n;
  char ttle[50];
cout<<"Enter details of 3 books";
for(int i=0;i<=2;++i)
obj[i].input();

cout<<endl;
cin.ignore();
cout<<"\n Enter title of required book\n";
  gets(ttle);

for(int i=0;i<2;i++)
  {
                    if(obj[i].access_title(ttle))
```

```
        {
cout<<"\nHow many copies? ";
cin>>n;
obj[i].getdata(n);
            break;
        }
        else
cout<<"\nBook unavailable";
   }
   return 0;
}
```