~~Aim :-~~

Assignment NO. (2)

Aim — Define a class to represent a bank account. Include the following members.

Data members :— Name of depositor, Account Number, Type of account, Balance amount in account,

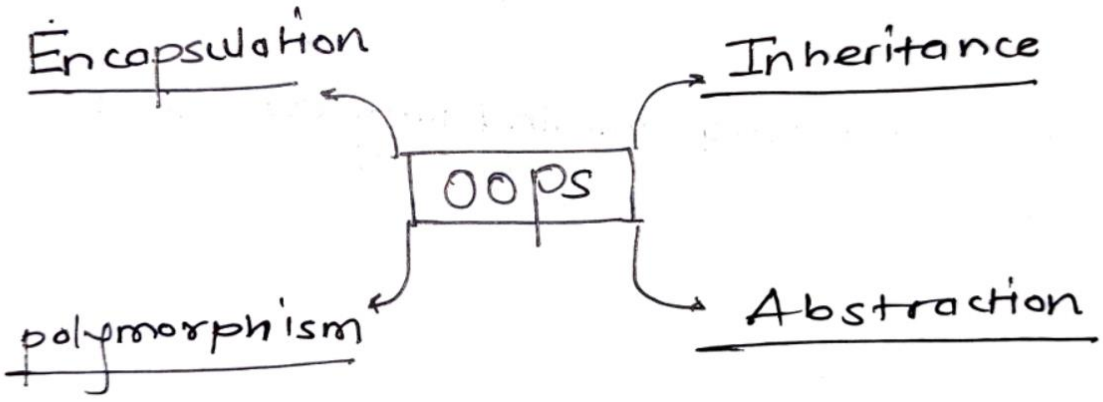
Member function :— To assign initial values, To deposit an amount, To withdraw after checking the balance, To display name and balance.

Write a main program to test program using class and obj.

Theory —

Object oriented programming

Object oriented programming is a programming style associated with the concepts of class and objects and various other concepts revolving around these two, like inheritance, polymorphism, Abstraction, Encapsulation, etc.



rudraksh.karpe.cs@ghrcem.raisoni.net

Class ⟶

A class is a blueprint of any functional entity which defines its properties and its functions. Like Human Being, having body parts, and performing various actions.

Inheritance ⟶

Inheritance is a way to reuse once written code again and again. The class inherited is called Base class and the class which inherits called as derived class. They are also called parent and child class.

So, when a derived class inherits a base class, the derived class can use all the functions which are defined in base class, hence making code reusable.

polymorphism ⟶

It is a feature that lets user create functions with same name but different arguments, that performs various actions.

Encapsulation ⟶

It can also be said data binding. Encapsulation is all about binding the data variables and functions ~~together~~ in class.

objects ⟶

Objects are basic units of oop. These are instances of class, which have data members and uses various member function to perform tasks.

rudraksh.karpe.cs@ghrcem.raisoni.net

Program code:

```
#include <iostream>
#include <stdio.h>
#include <string.h>

using namespace std;

class Bank
{
   int acno;
   char AccountHolderName[100], Account_Type[100];
   float bal;

public:
   Bank(int acc_no, char *name, char *acc_type, float Balance)
   {
      acno = acc_no;
      strcpy(AccountHolderName, name);
      strcpy(Account_Type, acc_type);
      bal = Balance;
   }
   void deposit();
   void withdraw();
   void Show();
};
void Bank::deposit()
{
   int DepositAmmount;
```

```
    cout << " Enter Deposit Amount = " ;
    cin >> DepositAmmount;
    bal += DepositAmmount;
}
void Bank::withdraw()
{
    int WithdrawAmmount;
    cout << " Enter Withdraw Amount = " ;
    cin >> WithdrawAmmount;
    if (WithdrawAmmount > bal)
        cout << " Cannot Withdraw Amount" ;
    bal -= WithdrawAmmount;
}
void Bank::Show()
{
    // cout << " _____" << endl;
    cout << " Accout No. : " << acno;
    cout << " Name : " << AccountHolderName;
    cout << " Account Type : " << Account_Type;
    cout << " Balance : " << bal;
}
int main()
{
    int acc_no;
    char name[100], acc_type[100];
    float Balance;
    cout << " ------------------Welcome to RK BANK---------- " << endl;
    cout << "_____" << endl;
    cout << " Please add account accout No.: ";
```

```
    cin >> acc_no;
    cout << " Please Enter Account Holder Name : ";
    cin >> name;
    cout << " Mention Account Type : ";
    cin >> acc_type;
    cout << " Initial Balance : ";
    cin >> Balance;

    Bank b1(acc_no, name, acc_type, Balance);
    b1.deposit();
    b1.withdraw();
    b1.Show();
    return 0;
}
```

Output of the program:

```
ystem.cpp -o bank_system } ; if ($?) { .\bank_system }
 ------------------Welcome to RK BANK-------------------

_____________________________________________________
 Please add account accout No.: 12345
 Please Enter Account Holder Name : Rudraksh_Karpe
 Mention Account Type : Current
 Initial Balance : 1000
 Enter Deposit Amount = 50
 Enter Withdraw Amount = 10
 Accout No. : 12345 Name : Rudraksh_Karpe Account Type : Current Balance : 1040
PS C:\Users\rudra\OneDrive\Desktop\Learnings\CPP TheNewboston\Day 3>
```