**CPP\_Assignment\_1c**

|  |  |  |
| --- | --- | --- |
| Requirement Tag | Requirement Description | Comments |
| BNK/01 | The class should have following data members : Customer Name (char \*), Account Number, Type of Account (Savings/Current etc), Account Balance. |  |
| BNK/02 | The class should allow basic operations like creating a new account, deposit an amount, withdraw money after checking the balance, display account details. | Hint: 1.Use overloaded constructors to create and initialize object members. 2. Access the members using this pointer 3. Perform validation and ensure sufficient balance before withdrawal. Else display error message "Insufficient balance" |
| BNK/03 | Add a non member function updateName() to receive a reference to an account object and a new name string. This function should update name and return the reference object. |  |
| BNK/04 | Add a default constructor to create account object with data members as below: Customer Name:"defaultName" Account Number:"123456789012" Type of Account:Savings Account Balance :0 |  |
| BNK/05 | a. Add a method BulkCreate() to create n number of objects with name initialized using given prefix and other attributes with default value as of ref object. Allocate dynamic memory for  customer name. Let each created object be stored in a global array.  int BukCreate(const Account &ref, int count, char \*prefix); count - number of objects to be created prefix - name prefix string eg. if n = 3, prefix="ALT\_NAME\_" then create 3 objects with name string as below and other attributes with default value as of ref object.  ALT\_NAME\_1  ALT\_NAME\_2  ALT\_NAME\_3  b. define all constructors as explicit | Should define and use a copy constructor |
| BNK/06 | Add destructor to deallocate memory allocated for account object |  |

**Ans:**

**#include<iostream>**

**#include<cstring>**

**using namespace std;**

**class Bank**

**{ private:**

**char customername[100];**

**int accountnumber;**

**string accounttype;**

**float accountbalance;**

**public:**

**Bank(){}**

**void account()**

**{**

**string customer\_name="defaultName";**

**cout<<"Customer Name:"<<customer\_name<<endl;**

**long int Account\_Number=123456789012;**

**cout<<"Account\_Number:"<<Account\_Number<<endl;**

**string Type\_of\_Account="Savings";**

**cout<<"Type\_of\_Account:"<<Type\_of\_Account<<endl;**

**int Account\_Balance=0;**

**cout<<"Account\_Balance:"<<Account\_Balance<<endl;**

**}**

**Bank(char \*cn,int an,string at,float ab)**

**{**

**strcpy(customername,cn);**

**accountnumber=an;**

**accounttype=at;**

**accountbalance=ab;**

**}**

**~Bank(){}**

**void deposit()**

**{**

**int depositamount;**

**cout<<"\nDeposit Amount:";**

**cin>>depositamount;**

**}**

**void withdraw()**

**{**

**int withdrawamount;**

**cout<<"\nEnter Withdraw Amount:";**

**cin>>withdrawamount;**

**if(withdrawamount>accountbalance)**

**{**

**cerr<<"Insufficient Balance:"<<endl;**

**accountbalance-=withdrawamount;**

**}**

**}**

**void display()**

**{**

**} cout<<"\n-------------------";**

**cout<<"\nCustomer Name:"<<customername;**

**cout<<"\nAccount Number:"<<accountnumber;**

**cout<<"\nAccount Type:"<<accounttype;**

**cout<<"\nAccount Balance:"<<accountbalance<<endl;**

**};**

**int main()**

**{**

**char customername [100];**

**int accountnumber;**

**string accounttype;**

**float accountbalance;**

**cout<<"\nEnter details:";**

**cout<<"\n-------------------";**

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

**cin>>customername;**

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

**cin>>accountnumber;**

**cout<<"\nAccount Type:";**

**cin>>accounttype;**

**cout<<"\nAccount Balance:";**

**cin>>accountbalance;**

**Bank obj;**

**Bank b1(customername,accountnumber,accounttype,accountbalance);**

**b1.display();**

**b1.deposit();**

**b1.withdraw();**

**obj.account();**

**Bank \*ptr;**

**ptr=new Bank;**

**delete ptr;**

**}**

**return 0;**

**}**

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence