chstylebackgroundcolor= basicstyle=, commentstyle=, keywordstyle=, stringstyle=, showstringspaces=false, numbers=left, numberstyle=, numbersep=10pt, tabsize=2, frame=L, framerule=1pt, rulecolor=, breaklines=true, inputpath=code

BANK MANAGEMENT SYSTEM

Objective:

The ”Bank Management System” has been developed to overcome the problems prevailing in the current manual system . The Bank Account Management System is an application for maintaining a person's account in a bank. In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks.

Basic functionalities provided are as follows :

- Getting user information
- Opening and closing account
- Pin Verification
- Displaying account information .
- Withdrawing and Depositing Amount
- Showing all account only to admin .

Programming Languages used in the Project :

- **C++**
- **Java**

Function Description

Bank Class :

Bank class is backbone of this program containing all the important variable and function's declaration which are important.

Open Account:

This function allows the user to open his/her new account in Bank.This function collect basic user information like first name,last name and initial depositing amount followed by pin creation which is 4 digit unique number.And the information is stored in a map whose key is account no. provided by bank.

Balance Enquiry:

This function allows the user to see his account information like net account balance.The user is required to enter correct pin and account number to see his account information.

Deposit :

This function allows the user to Deposit money in its bank account.This function is responsible to increase the user's account balance.Followed by pin verification.

Withdraw function:

This function allows the user to Withdraw money from its bank account.This function is responsible to decrease the user's account balance.Followed by pin verification.

Delete account Function:

This function provide option to user to create its account from bank.Also the information is removed from system.

Store:

This is one of the main function in this program responsible for storing all the information related to user user name, pin , account number,balance.

Admin Verification function :

This function is especially created for verifying admin password. This function ensures that no person except admin is able to see all account information.

Display all function:

This function allow only admin to view all the information related to user account.The data is fetched from map shown to screen with display function

Create Pin function

This is important from security point of view.This function make sure that user create only four digit correct pin. This function give user three chance to create a correct pin. If failed the function takes user to main window.

Pin verify:

This function verify the pin entered by user.Redirecting to main menu if pin not entered correctly in three chances.

C++ Code :

```
#include<bits/stdc++.h>
#include<windows.h>
#include<conio.h>
using namespace std;
void delay()     //for delay in time
{
    for(int i=0;i<3;i++)
    {
                cout<<".";
                Sleep(1000);
        }
}
/*This Class contains all information related to account */
class Bank
{
    private:
        int account_no;
        string first_name;
        string last_name;
        int balance;
        static map<int,Bank> accounts;
        static int next_account;
        int pin;
    public:
        Bank()
        {
            account_no=0;
            balance=0;

        }
        int get_account_no()
```

```
        {
            return account_no;
        }

        //Declaration of all function
        void open_account();              -
        void balance_enquiry(int account_no);

        void deposit(int account_no,int amount);
        void withdraw(int account_no,int amount);
        void store(Bank p);
        void display(int account_no);
        void display_all();
        bool creat_pin();
        bool pin_verify(int);
        void delete_account(int);
        static bool admin_verification();

};
int Bank::next_account=0;  //This holds Account no. of last Account
map<int,Bank>Bank::accounts; //Map stores all the data entered by user
void Bank::open_account() //opening an account
{
    int flag=3;
    Bank new_account;
/*---Collecting personal information---*/
    cout<<"Enter your First Name:";
    cin>>new_account.first_name;
    cout<<"Enter your Last Name:";
    cin>>new_account.last_name;
    label:
    cout<<"Enter depositing amount:";
    cin>>new_account.balance;
    if(new_account.balance>=500)  //checking initial amount greater then
    {
        next_account++;
        new_account.account_no=next_account;
        if(new_account.creat_pin())
        {

            new_account.store(new_account);
            cout<<"\t\tYour new account is creating please wait";
            delay();
            cout<<"\nCongratulation your account has been created!!";
            cout<<"\nHere are your details:\n\n";
            new_account.display(new_account.get_account_no());
            cout<<"Press any key to continue";
            getch();
        }
    }
```

```
    else
    {
        cout<<"Enter Initial Amount greater than 500\n";
        flag--;
        if(flag>0)
            goto label;
        cout<<"Sorry try Again1";
    }
}
void Bank::balance_enquiry(int account_no)
{

    map<int,Bank>::iterator itr=accounts.find(account_no);
    if(accounts.find(account_no)!=accounts.end())
    {
        if(pin_verify(account_no))
        {
            //displaying the account info
            cout<<"Here are your details:\n\n";
            display(account_no);
            cout<<"Press any key to continue";
            getch();
        }
        else
        {
            cout<<"Try Again\n";
            cout<<"Press any key to continue";
            getch();
        }
    }
    else                //if account not found
    {
        cout<<"Account don't exist\n";
        cout<<"Try again";
        cout<<"Press any key to continue";
        getch();
    }


}
void Bank::deposit(int account_no,int amount)
{
    map<int,Bank>::iterator itr=accounts.find(account_no);
    if(accounts.find(account_no)!=accounts.end())
    {
        if(pin_verify(account_no))
        {
            itr->second.balance+=amount; //incrementing the balance
            cout<<"Here are your details:\n\n";
            display(account_no);
```

```
            cout<<"Press any key to continue";
            getch();
        }
        else{
            cout<<"Try Again\n";
            cout<<"Press any key to continue";
            getch();
        }
    }
    else
    {
        cout<<"Account don't exist\n";
        cout<<"Try again";
        cout<<"Press any key to continue";
        getch();
    }
}
void Bank::withdraw(int account_no,int amount)
{
    map<int,Bank>::iterator itr=accounts.find(account_no);

    if(accounts.find(account_no)!=accounts.end())
    {
        if(pin_verify(account_no))
        {
            amount= itr->second.balance-amount;
            if(amount<0){ //checking if current balance is sufficient
                cout<<"Insufficient Balance\n";
                cout<<"Press any key to continue";
                getch();
            }
            else
            {
                itr->second.balance=amount; //deducting amount
                cout<<"Here are your details:\n";
                display(account_no);
                 cout<<"Press any key to continue";
                 getch();
            }
        }
        else{
                cout<<"Try Again";
        }
    }
    else
    {
        cout<<"Account don't exist\n";
        cout<<"Try again";
        cout<<"Press any key to continue";
        getch();
```

```
    }

}
void Bank::delete_account(int account_no)     //deleteing an account
{
    map<int,Bank>::iterator itr=accounts.find(account_no);
    if(accounts.find(account_no)!=accounts.end())
    {
        if(pin_verify(account_no))
        {
            accounts.erase(account_no);
            cout<<"Removing your account please wait";
            delay();
            cout<<"Your account has been removed";
            getch();
        }
        else{
            cout<<"Try Again";
            cout<<"Press any key for home page";
            getch();
        }
    }
    else
    {
        cout<<"Account don't exist\n";
        cout<<"Try again";
        getch();
    }
}
void Bank::store(Bank p)              //storing all the information
{
    accounts.insert({p.get_account_no(),p});

}
void Bank::display(int account_no)
{
    //displaying account information
    map<int,Bank>::iterator itr=accounts.find(account_no);
    cout<<"\n-----------------------------------------\n";
    cout<<"Account Number:"<<itr->second.account_no<<endl;
    cout<<"First Name:"<<itr->second.first_name<<endl;
    cout<<"Last Name:"<<itr->second.last_name<<endl;
    cout<<"Balance:"<<itr->second.balance;
    cout<<"\n-----------------------------------------\n";
}
bool Bank:: admin_verification()
{
    string admin_password,entered_pass;
    cout<<"\nEnter Password:";
    cin>>entered_pass;
```

```
    admin_password="kunal@123456";   //admin password
    if(entered_pass==admin_password)
    {
        return true;
    }
    else
        return false;
}
void Bank::display_all()  //dispalying all account information
{
    map<int,Bank>::iterator it;
    cout<<"Only admin can access it";
    if( admin_verification())
    {
        cout<<"All account details:\n";
        for(it=accounts.begin(); it!=accounts.end(); it++) //traversing
        {
            if(accounts.size()!=0){
                display(it->second.account_no);
                cout<<"\n";
            }
            else{
                cout<<"No Account to show ";  //if no account exist
            }
        }
    }
    else
    {
        cout<<"Entered Wrong Password\nTry Again";
        cout<<"Press Enter to continue ";
        getch();

    }
}
bool Bank::creat_pin()
{
    int pin=0,flag=3,temp=1,i;
    label:
    cout<<"Enter four digit Pin:";
    cin>>pin;
    temp=pin;
    for(i=0;temp!=0;i++)    //checking length of pin
        temp=temp/10;
    if(i==4)
    {
        this->pin=pin;
        return true;
    }
    else
    {
```

```
        cout<<"Please enter valid Pin\nTry again\n";
        flag--;
        cout<<"You have "<<flag<<" chances";
        if(flag!=0)
             goto label;
        cout<<"Sorry You have reached your limit";
        cout<<"Try Again";
        return false;

    }
}
bool Bank::pin_verify(int account_no)    //pin verfication
{
    map<int,Bank>::iterator itr=accounts.find(account_no);
    int flag_pin=0,flag=3;
    if(itr!=accounts.end())
    {
        label:
        cout<<"Enter your Pin:";
        cin>>flag_pin;
        if(flag_pin==itr->second.pin)
        {
            cout<<"Pin verified\n";
            return true;
        }
        else
        {
            cout<<"Pin not verified\n";
            cout<<"Try Again\n";
            flag--;
            if(flag==0)
            {
                cout<<"Sorry You have reached your limit";
                cout<<"Try Again";
                return false ;
            }
            cout<<"You have "<<flag<<" chances\n";
            goto label;
        }
    }
    else
        cout<<"Account don't exist";
}
void start()       //adding some animation
{
    char arr[]={'B','A','N','K',' ','M','A','N','A','G','E','M','E'
                        ,'N','T',' ','S','Y','S','T','E','M'};
                        cout<<"\n\n\n\n\t\t\t";
                        for(int i=0;i<22;i++){
                                cout<<arr[i];
```

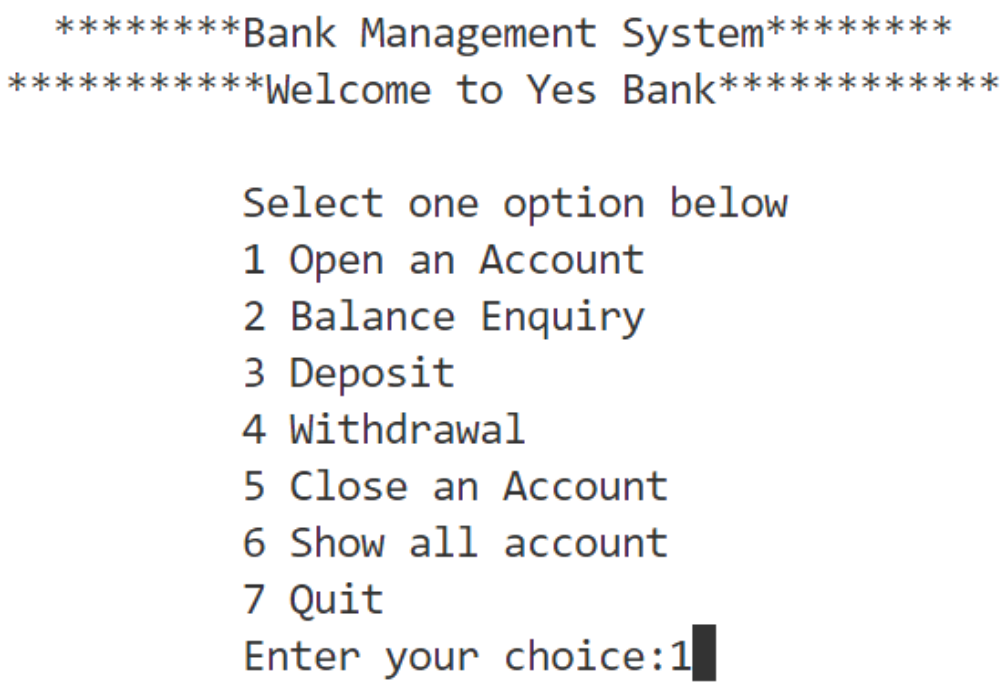
```
                                Sleep(30);
                            }
                            Sleep(1000);
                            system("CLS");
}
int main()
{
    int choice=1,amount=0;
    Bank b,c;
    int account_no;
    string fname,lname;
    start();
    while(choice!=7)
    {
        // cout<<"\t\t----------------------------------------\n";
        // cout<<"\t\t|
|\n";
        // cout<<"\t\t|           Bank Management System
|\n";

        cout<<"\n\n\t\t\t********Bank Management System********"<<endl;
        cout<<"\t\t      ***********Welcome to Yes Bank***********";
    cout<<"\n\n\t\t\t\tSelect one option below "<<endl;
        cout<<"\t\t\t\t1 Open an Account"<<endl;
        cout<<"\t\t\t\t2 Balance Enquiry"<<endl;
        cout<<"\t\t\t\t3 Deposit"<<endl;
        cout<<"\t\t\t\t4 Withdrawal"<<endl;
        cout<<"\t\t\t\t5 Close an Account"<<endl;
        cout<<"\t\t\t\t6 Show all account"<<endl;
        cout<<"\t\t\t\t7 Quit"<<endl;
        cout<<"\t\t\t\tEnter your choice:";
         cin>>choice;    //selecting choice

        system("ClS");
        switch(choice)
        {
            case 1:                             //opening account
                b.open_account();
                system("ClS");
                break;
            case 2:                             //balance enquiry
                cout<<"Enter account number:";
                cin>>account_no;
                b.balance_enquiry(account_no);
                system("ClS");
                break;
            case 3:                             //depositing amount
                 cout<<"Enter account number:";
                cin>>account_no;
                cout<<"Enter Depositing amount:";
```

```
                cin>>amount;
                b.deposit(account_no,amount);
                system("ClS");
                break;
            case 4:                          //withdrawing amount
                 cout<<"Enter account number:";
                cin>>account_no;
                cout<<"Enter withdrawal amount:";
                cin>>amount;
                b.withdraw(account_no,amount);
                system("ClS");
                break;
            case 5:                         //deleting account
                cout<<"Enter account no:";
                cin>>account_no;
                b.delete_account(account_no);
                system("ClS");
                break;
            case 6:              //diplaying all account information
               b.display_all();
               getch();
               system("ClS");
               break;
            case 7:
                system("ClS");
                break;
            default:
                cout<<"Enter a valid choice";
        }
    }
    return 0;
}
```

// C++ Code Output :



```
Enter your First Name:kunal
Enter your Last Name:mali
Enter depositing amount:100
Enter Initial Amount greater than 500
Enter depositing amount:500
Enter four digit Pin:12
Please enter valid Pin
Try again
You have 2 chancesEnter four digit Pin:1234
                Your new account is creating please wait...
Congratulation your account has been created!!
Here are your details:


------------------------------------------------
Account Number:1
First Name:kunal
Last Name:mali
Balance:500
------------------------------------------------
Press any key to continue
```

```
Enter account number:1
Enter your Pin:123
Pin not verified
Try Again
You have 2 chances
Enter your Pin:1234
Pin verified
Here are your details:


-----------------------------------------------
Account Number:1
First Name:kunal
Last Name:mali
Balance:500
-----------------------------------------------
Press any key to continue


Enter account number:1
Enter Depositing amount:300
Enter your Pin:1234
Pin verified
Here are your details:


-----------------------------------------------
Account Number:1
First Name:kunal
Last Name:mali
Balance:800
-----------------------------------------------
Press any key to continue
```

```
Enter account no:1
Enter your Pin:1234
Pin verified
Removing your account please wait...Your account has been removed
```

// C++ Code Debugging :

```
345         while(choice!=7)
(gdb) n
351             cout<<"\n\n\t\t\t********Bank Management System********"<<endl;
(gdb) n


                        ********Bank Management System********
352             cout<<"\t\t      ***********Welcome to Yes Bank************";
(gdb) n
                      ***********Welcome to Yes Bank************353         cout<<"\n\n\t\t\t\tSelect one option below "<<endl;
(gdb) n


                                Select one option below
354             cout<<"\t\t\t\t1 Open an Account"<<endl;
(gdb) n
                                1 Open an Account
355             cout<<"\t\t\t\t2 Balance Enquiry"<<endl;
(gdb) n
                                2 Balance Enquiry
356             cout<<"\t\t\t\t3 Deposit"<<endl;
(gdb) n
                                3 Deposit
357             cout<<"\t\t\t\t4 Withdrawal"<<endl;
(gdb) n
                                4 Withdrawal
358             cout<<"\t\t\t\t5 Close an Account"<<endl;
(gdb) n
                                5 Close an Account
359             cout<<"\t\t\t\t6 Show all account"<<endl;
(gdb) n
                                6 Show all account
360             cout<<"\t\t\t\t7 Quit"<<endl;
(gdb) n
                                7 Quit
361             cout<<"\t\t\t\tEnter your choice:";
(gdb) n
                                Enter your choice:362           cin>>choice;    //selecting choice
(gdb) n
1
```

```
365             switch(choice)
(gdb) n
368                     b.open_account();
(gdb) n
Enter your First Name:kunal
Enter your Last Name:mali
Enter depositing amount:500
Enter four digit Pin:1234
                Your new account is creating please wait...
Congratulation your account has been created!!
Here are your details:


-----------------------------------------------
Account Number:1
First Name:kunal
Last Name:mali
Balance:500
-----------------------------------------------
Press any key to continue
```

```
365             switch(choice)
(gdb) n
372                     cout<<"Enter account number:";
(gdb) n
Enter account number:373                        cin>>account_no;
(gdb) n
1
374                     b.balance_enquiry(account_no);
(gdb) n
Enter your Pin:1234
Pin verified
Here are your details:


-----------------------------------------------
Account Number:1
First Name:kunal
Last Name:mali
Balance:500
-----------------------------------------------
Press any key to continue
```

```
365             switch(choice)
(gdb) n
378                      cout<<"Enter account number:";
(gdb) n
Enter account number:379                        cin>>account_no;
(gdb) n
1
380                     cout<<"Enter Depositing amount:";
(gdb) n
Enter Depositing amount:381                     cin>>amount;
(gdb) n
300
382                     b.deposit(account_no,amount);
(gdb) n
Enter your Pin:1234
Pin verified
Here are your details:


-----------------------------------------------
Account Number:1
First Name:kunal
Last Name:mali
Balance:800
-----------------------------------------------
Press any key to continue
```

```
365             switch(choice)
(gdb) n
386                      cout<<"Enter account number:";
(gdb) n
Enter account number:387                        cin>>account_no;
(gdb) n
1
388                     cout<<"Enter withdrawal amount:";
(gdb) n
Enter withdrawal amount:389                     cin>>amount;
(gdb) 800
Undefined command: "800".  Try "help".
(gdb) n
800
390                     b.withdraw(account_no,amount);
(gdb) n
Enter your Pin:1234
Pin verified
Here are your details:

-----------------------------------------------
Account Number:1
First Name:kunal
Last Name:mali
Balance:0
-----------------------------------------------
Press any key to continue
```

```
365             switch(choice)
(gdb) n
400                     b.display_all();
(gdb) n
Only admin can access it
Enter Password:kunal@123456
All account details:

------------------------------------------------
Account Number:1
First Name:kunal
Last Name:mali
Balance:0
------------------------------------------------

401                     getch();
(gdb) n
402                     system("ClS");
(gdb)
```

```
406                     break;
(gdb) n
345         while(choice!=7)
(gdb) n
411         return 0;
(gdb) n
343         string fname,lname;
(gdb) n
341         Bank b,c;
(gdb) n
412     }
(gdb) n
0x00401288 in _Jv_RegisterClasses ()
(gdb) n
Single stepping until exit from function _Jv_RegisterClasses,
which has no line number information.
0x0040128a in _Jv_RegisterClasses ()
(gdb) n
Single stepping until exit from function _Jv_RegisterClasses,
which has no line number information.
0x0040128f in _Jv_RegisterClasses ()
(gdb) n
Single stepping until exit from function _Jv_RegisterClasses,
which has no line number information.
0x00401292 in _Jv_RegisterClasses ()
(gdb)
```

Java Code :

```
import jdk.jfr.Label;
import java.util.*;
import java.lang.*;
import java.util.Map.Entry;
class Bank
{
    Scanner sc = new Scanner(System.in);
    private
    int account_no;
    private String first_name;
    private String last_name;
    private int balance;
    private static HashMap<Integer, Bank> accounts = new HashMap<Integer,
    private static int next_account = 0;
    private int pin;
    public Bank()
    {
        account_no = 0;
        balance = 0;

    }

    public int get_account_no() {
        return account_no;
    }

    public void open_account()
    {
        int flag = 3;
        Bank new_account = new Bank();
        System.out.println("Enter your First Name:");
        new_account.first_name = sc.next();
        System.out.println("Enter your Last Name:");
        new_account.last_name = sc.next();
        do
        {
            System.out.println("Enter depositing amount:");
            new_account.balance = sc.nextInt();
            if (new_account.balance >= 500)
            {
                next_account++;
                new_account.account_no = next_account;
                if (new_account.creat_pin())
                {
                    System.out.println("Please Wait\n");
                    new_account.store(new_account);
                    System.out.println("\nCongratulation your account has
                    System.out.println("\nHere are your details:\n");
```

```
                new_account.display(new_account.get_account_no());
                break;
            }
        }
        else
        {

            flag--;
            if (flag == 0)
            {
                System.out.println("Sorry try Again");
                break;
            }
            System.out.println("Enter Initial Amount greater than 500
        }

    }while (flag > 0);


}

public void balance_enquiry(int account_no)
{

    //HashMap<Integer,Bank>::iterator itr=accounts.find(account_no);
    if (accounts.get(account_no) != null) {
        if (pin_verify(account_no))
            display(account_no);
        else
            System.out.println("Try Again");
    } else {
        System.out.println("Account don't exist\n");
        System.out.println("Try again");
    }


}
public void deposit(int account_no, int amount)
{
    //  accounts:iterator itr=accounts.find(account_no);
    if (accounts.get(account_no) != null)
    {
        if (pin_verify(account_no))
        {
            accounts.get(account_no).balance += amount;
            //accounts.getKey();
            display(account_no);
        } else
            System.out.println("Try Again");
    }
```

```
    else
    {
        System.out.println("Account don't exist\n");
        System.out.println("Try again");
    }
}

public void withdraw(int account_no, int amount)
{
    // map<Integer,Bank>::iterator itr=accounts.find(account_no);

    if (accounts.get(account_no) != null)
    {
        if (pin_verify(account_no))
        {
            amount = (accounts.get(account_no).balance) - amount;
            if (amount < 0)
                System.out.println("Insufficient Balance");
            else {
                accounts.get(account_no).balance = amount;
                display(account_no);
            }
        }
        else
            System.out.println("Try Again");
    } else
    {
        System.out.println("Account don't exist\n");
        System.out.println("Try again");
    }

}

public void delete_account(int account_no)
{
    // map<int,Bank>::iterator itr=accounts.find(account_no);
    if (accounts.get(account_no) != null)
    {
        if (pin_verify(account_no))
        {
            accounts.remove(account_no);
            System.out.println("Your account has been removed");
        }
        else
            System.out.println("Try Again");
    }
    else
    {
        System.out.println("Account don't exist\n");
        System.out.println("Try again");
```

```
        }


    }

    public void store(Bank p) {
        accounts.put(p.get_account_no(), p);

    }

    void display(int account_no)
    {

        System.out.println("Account Number:" + accounts.get(account_no).a
        System.out.println("First Name:" + accounts.get(account_no).first_
        System.out.println("Last Name:" + accounts.get(account_no).last_na
        System.out.println("Balance:" + accounts.get(account_no).balance

    }

    boolean admin_verification()
    {

        String admin_password;
        String entered_pass;
        System.out.print("\nEnter Password:");
        entered_pass = sc.next();
        admin_password = "kunal@123456";
        if (admin_password.equals(entered_pass))
            return true;
        else
            return false;


    }


    public void display_all()
    {

        //map<Integer,Bank>::iterator it;
        System.out.println("Only admin can access it");
        if (admin_verification())
        {
            for (Entry<Integer, Bank> entry : accounts.entrySet())
            {
                //System.out.println("inside true");
                display(entry.getValue().account_no);
                //System.out.println("\n");
            }
```

```
        }
        else
        {
            System.out.println("Entered Wrong Password\nTry Again");

        }
    }

    public boolean creat_pin()
    {
        int pin = 0, flag = 3, temp = 1, i;
        label:
        while (flag > 0)
        {
            System.out.println("Enter four digit Pin:");
            pin = sc.nextInt();
            temp = pin;
            for (i = 0; temp != 0; i++)
                temp = temp / 10;
            if (i == 4)
            {
                this.pin = pin;
                return true;
            }
            else
            {
                System.out.println("Please enter valid Pin\nTry again\n");
                flag--;
                System.out.println("You have " + flag + " chances");
                if (flag != 0)
                    continue;
                System.out.println("Sorry You have reached your limit");
                System.out.println("Try Again");
                return false;

            }
        }
        return true;

    }

    public boolean pin_verify(int account_no)
    {
        //map<Integer,Bank>::iterator itr=accounts.find(account_no);
        int flag_pin = 0, flag = 3;
        if (accounts.containsKey(account_no))
        {
            label:
            while (flag > 0) {
                System.out.println("Enter your Pin:");
```

```
                flag_pin = sc.nextInt();
                if (flag_pin == accounts.get(account_no).pin) {
                    System.out.println("Pin verified");
                    return true;
                } else {
                    System.out.println("Pin not verified");
                    System.out.println("Try Again");
                    flag--;
                    if (flag == 0)
                    {
                        System.out.println("Sorry You have reached your l
                        System.out.println("Try Again");
                        return false;
                    }
                    System.out.println("You have " + flag + " chances lef
                }
            }
        }
        else
        {
            System.out.println("Account don't exist");

        }
        return true;


    }
}


// map<int,Bank>accounts;
public class Main
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int choice=1,amount=0;
        Bank b= new Bank();
        int account_no;
        String fname,lname;
        System.out.println("\n\n\t\t\t********Bank Management System*****
        System.out.println("\t\t    **********Welcome to Yes Bank********
        while(choice!=7)
        {
            System.out.println("\t\tSelect one option below ");
            System.out.println("\t\t1 Open an Account");
            System.out.println("\t\t2 Balance Enquiry");
            System.out.println("\t\t3 Deposit");
            System.out.println("\t\t4 Withdrawal");
            System.out.println("\t\t5 Close an Account");
```

```
                System.out.println("\t\t6 Show all account");
                System.out.println("\t\t7 Quit");
                System.out.print("Enter your choice:");
                choice=sc.nextInt();
                switch(choice)
                {
                    case 1:
                        b.open_account();
                        break;

                    case 2:
                        System.out.println("Enter account number:");
                        account_no=sc.nextInt();
                        b.balance_enquiry(account_no);
                        break;

                    case 3:
                        System.out.println("Enter account number:");
                        account_no=sc.nextInt();
                        System.out.println("Enter Balance:");
                        amount=sc.nextInt();
                        b.deposit(account_no,amount);
                        break;


                    case 4:
                        System.out.println("Enter account number:");
                        account_no=sc.nextInt();
                        System.out.println("Enter withdrawal amount");
                        amount=sc.nextInt();
                        b.withdraw(account_no,amount);
                        break;
                    case 5:
                        System.out.println("Enter account no");
                        account_no=sc.nextInt();
                        b.delete_account(account_no);
                        break;

                    case 6:
                        b.display_all();
                        break;
                    case 7:
                        break;
                    default:
                        System.out.println("Enter a valid choice");

                }
            }
        }
}
```

Java Code Output :

```
            ********Bank Management System********
           ***********Welcome to Yes Bank************
         Select one option below
         1 Open an Account
         2 Balance Enquiry
         3 Deposit
         4 Withdrawal
         5 Close an Account
         6 Show all account
         7 Quit
Enter your choice:1
Enter your First Name:
kunal
Enter your Last Name:
mali
Enter depositing amount:
50
Enter Initial Amount greater than 500

Enter depositing amount:
500
Enter four digit Pin:
1234
Please Wait
```

```
Enter depositing amount:
500
Enter four digit Pin:
1234
Please Wait


Congratulation your account has been created!!

Here are your details:

Account Number:1
First Name:kunal
Last Name:mali
Balance:500
        Select one option below
        1 Open an Account
        2 Balance Enquiry
        3 Deposit
        4 Withdrawal
        5 Close an Account
        6 Show all account
        7 Quit
Enter your choice:
```

```
        Select one option below
        1 Open an Account
        2 Balance Enquiry
        3 Deposit
        4 Withdrawal
        5 Close an Account
        6 Show all account
        7 Quit
Enter your choice:2
Enter account number:
1
Enter your Pin:
1234
Pin verified
Account Number:1
First Name:kunal
Last Name:mali
Balance:500
        Select one option below
        1 Open an Account
        2 Balance Enquiry
        3 Deposit
        4 Withdrawal
        5 Close an Account
```

```
        5 Close an Account
        6 Show all account
        7 Quit
Enter your choice:3
Enter account number:
1
Enter Balance:
600
Enter your Pin:
1234
Pin verified
Account Number:1
First Name:kunal
Last Name:mali
Balance:1100
        Select one option below
        1 Open an Account
        2 Balance Enquiry
        3 Deposit
        4 Withdrawal
        5 Close an Account
        6 Show all account
        7 Quit
Enter your choice:4
```

```
        5 Close an Account
        6 Show all account
        7 Quit
Enter your choice:4
Enter account number:
1
Enter withdrawal amount
90
Enter your Pin:
1234
Pin verified
Account Number:1
First Name:kunal
Last Name:mali
Balance:1010
        Select one option below
        1 Open an Account
        2 Balance Enquiry
        3 Deposit
        4 Withdrawal
        5 Close an Account
        6 Show all account
        7 Quit
Enter your choice:6
```

```
          5 Close an Account
          6 Show all account
          7 Quit
Enter your choice:6
Only admin can access it

Enter Password:kunal@123456
Account Number:1
First Name:kunal
Last Name:mali
Balance:1010
Account Number:2
First Name:ayush
Last Name:verma
Balance:700
          Select one option below
          1 Open an Account
          2 Balance Enquiry
          3 Deposit
          4 Withdrawal
          5 Close an Account
          6 Show all account
          7 Quit
Enter your choice:7
```

C++ profiling:

```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00       52     0.00     0.00  std::_Rb_tree_iterator<std::pair<int const, Bank> >::_Rb_tree_iterator(std::_Rb_tree_node_base*)
  0.00      0.00     0.00       36     0.00     0.00  __gnu_cxx::__aligned_membuf<std::pair<int const, Bank> >::_M_ptr() const
  0.00      0.00     0.00       36     0.00     0.00  __gnu_cxx::__aligned_membuf<std::pair<int const, Bank> >::_M_addr() const
  0.00      0.00     0.00       36     0.00     0.00  std::_Select1st<std::pair<int const, Bank> >::operator()(std::pair<int const, Bank> const&) const
  0.00      0.00     0.00       36     0.00     0.00  std::_Rb_tree_node<std::pair<int const, Bank> >::_M_valptr() const
  0.00      0.00     0.00       36     0.00     0.00  std::less<int>::operator()(int const&, int const&) const
  0.00      0.00     0.00       36     0.00     0.00  std::_Rb_tree<int, std::pair<int const, Bank>, std::_Select1st<std::pair<int const, Bank> >,
std::less<int>, std::allocator<std::pair<int const, Bank> > >::_S_key(std::_Rb_tree_node<std::pair<int const, Bank> > const*)
  0.00      0.00     0.00       30     0.00     0.00  __gnu_cxx::__aligned_membuf<std::pair<int const, Bank> >::_M_ptr()
  0.00      0.00     0.00       30     0.00     0.00  __gnu_cxx::__aligned_membuf<std::pair<int const, Bank> >::_M_addr()
  0.00      0.00     0.00       30     0.00     0.00  std::_Rb_tree_node<std::pair<int const, Bank> >::_M_valptr()
  0.00      0.00     0.00       28     0.00     0.00  std::_Rb_tree_iterator<std::pair<int const, Bank> >::operator->() const
  0.00      0.00     0.00       28     0.00     0.00  std::_Rb_tree<int, std::pair<int const, Bank>, std::_Select1st<std::pair<int const, Bank> >,
std::less<int>, std::allocator<std::pair<int const, Bank> > >::end()
  0.00      0.00     0.00       21     0.00     0.00  std::_Rb_tree<int, std::pair<int const, Bank>, std::_Select1st<std::pair<int const, Bank> >,
std::less<int>, std::allocator<std::pair<int const, Bank> > >::_M_mbegin() const
  0.00      0.00     0.00       21     0.00     0.00  std::_Rb_tree<int, std::pair<int const, Bank>, std::_Select1st<std::pair<int const, Bank> >,
std::less<int>, std::allocator<std::pair<int const, Bank> > >::_M_begin()
  0.00      0.00     0.00       20     0.00     0.00  std::_Rb_tree<int, std::pair<int const, Bank>, std::_Select1st<std::pair<int const, Bank> >,
std::less<int>, std::allocator<std::pair<int const, Bank> > >::_M_end()
  0.00      0.00     0.00       19     0.00     0.00  std::_Rb_tree<int, std::pair<int const, Bank>, std::_Select1st<std::pair<int const, Bank> >,
std::less<int>, std::allocator<std::pair<int const, Bank> > >::_S_left(std::_Rb_tree_node_base*)
  0.00      0.00     0.00       18     0.00     0.00  std::_Rb_tree<int, std::pair<int const, Bank>, std::_Select1st<std::pair<int const, Bank> >,
std::less<int>, std::allocator<std::pair<int const, Bank> > >::_M_lower_bound(std::_Rb_tree_node<std::pair<int const, Bank> >*, std::_Rb_tree_node_base*,
int const&)
  0.00      0.00     0.00       18     0.00     0.00  std::operator==(std::_Rb_tree_iterator<std::pair<int const, Bank> > const&,
std::_Rb_tree_iterator<std::pair<int const, Bank> > const&)
  0.00      0.00     0.00       17     0.00     0.00  std::map<int, Bank, std::less<int>, std::allocator<std::pair<int const, Bank> > >::find(int const&)
  0.00      0.00     0.00       17     0.00     0.00  std:: Rb tree<int, std::pair<int const, Bank>, std:: Select1st<std::pair<int const, Bank> >,
```

```
 0.00      0.00     0.00        2     0.00     0.00  std::operator==(std::_Rb_tree_const_iterator<std::pa:
td::_Rb_tree_const_iterator<std::pair<int const, Bank> > const&)
 0.00      0.00     0.00        2     0.00     0.00  operator new(unsigned long, void*)
 0.00      0.00     0.00        1     0.00     0.00  __static_initialization_and_destruction_0(int, int)
 0.00      0.00     0.00        1     0.00     0.00  Bank::display_all()
 0.00      0.00     0.00        1     0.00     0.00  Bank::open_account()
 0.00      0.00     0.00        1     0.00     0.00  Bank::delete_account(int)
 0.00      0.00     0.00        1     0.00     0.00  Bank::balance_enquiry(int)
 0.00      0.00     0.00        1     0.00     0.00  Bank::admin_verification()
 0.00      0.00     0.00        1     0.00     0.00  Bank::store(Bank)
 0.00      0.00     0.00        1     0.00     0.00  Bank::deposit(int, int)
 0.00      0.00     0.00        1     0.00     0.00  Bank::withdraw(int, int)
 0.00      0.00     0.00        1     0.00     0.00  Bank::creat_pin()
 0.00      0.00     0.00        1     0.00     0.00  Bank::Bank(Bank&&)
```

```
                     Call graph (explanation follows)


granularity: each sample hit covers 4 byte(s) no time propagated

index % time    self  children    called     name
                0.00    0.00       1/52          std::_Rb_tree<int, st
std::less<int>, std::allocator<std::pair<int const, Bank> > >::_M_get_
                0.00    0.00       1/52          std::_Rb_tree_iterato
std::_Select1st<std::pair<int const, Bank> >, std::less<int>, std::all
std::_Rb_tree<int, std::pair<int const, Bank>, std::_Select1st<std::pa
>::_Alloc_node>(std::_Rb_tree_node_base*, std::_Rb_tree_node_base*, st
std::_Select1st<std::pair<int const, Bank> >, std::less<int>, std::all
                0.00    0.00       1/52          std::_Rb_tree<int, st
std::less<int>, std::allocator<std::pair<int const, Bank> > >::_M_uppe
int const&) [99]
                0.00    0.00       3/52          std::_Rb_tree<int, st
std::less<int>, std::allocator<std::pair<int const, Bank> > >::begin()
                0.00    0.00      18/52          std::_Rb_tree<int, st
std::less<int>, std::allocator<std::pair<int const, Bank> > >::_M_lowe
int const&) [24]
                0.00    0.00      28/52          std::_Rb_tree<int, st
std::less<int>, std::allocator<std::pair<int const, Bank> > >::end() [
[8]      0.0    0.00    0.00      52          std::_Rb_tree_iterator<st
-----------------------------------------------
                0.00    0.00      36/36          std::_Rb_tree_node<st
[9]      0.0    0.00    0.00      36          __gnu_cxx::__aligned_memb
                0.00    0.00      36/36          __gnu_cxx::__aligned_
-----------------------------------------------
                0.00    0.00      36/36          __gnu_cxx::__aligned_
[10]     0.0    0.00    0.00      36          __gnu_cxx::__aligned_memb
-----------------------------------------------
                0.00    0.00      36/36          std::_Rb_tree<int, st
std::less<int>, std::allocator<std::pair<int const, Bank> > >::_S_key(
[11]     0.0    0.00    0.00      36          std:: Select1st<std::pair
               0.00    0.00      30/30          __gnu_cxx::__aligned_membuf<std::p
-----------------------------------------------
                0.00    0.00       1/28          Bank::deposit(int, int) [56]
                0.00    0.00       1/28          Bank::display_all() [50]
                0.00    0.00       2/28          Bank::withdraw(int, int) [57]
                0.00    0.00       4/28          Bank::pin_verify(int) [34]
                0.00    0.00      20/28          Bank::display(int) [33]
[18]     0.0    0.00    0.00      28          std::_Rb_tree_iterator<std::pair<int c
                0.00    0.00      28/30          std::_Rb_tree_node<std::pair<int c
-----------------------------------------------
```

```
-----------------------------------------------
                0.00    0.00       2/17          Bank::balance_enquiry(int) [53]
                0.00    0.00       2/17          Bank::deposit(int, int) [56]
                0.00    0.00       2/17          Bank::withdraw(int, int) [57]
                0.00    0.00       2/17          Bank::delete_account(int) [52]
                0.00    0.00       4/17          Bank::pin_verify(int) [34]
                0.00    0.00       5/17          Bank::display(int) [33]
[26]     0.0    0.00    0.00      17         std::map<int, Bank, std::less<int>, st
                0.00    0.00      17/17          std::_Rb_tree<int, std::pair<int c
std::less<int>, std::allocator<std::pair<int const, Bank> > >::find(int const&) [27
-----------------------------------------------


-----------------------------------------------
                0.00    0.00       1/10          Bank::balance_enquiry(int) [53]
                0.00    0.00       1/10          Bank::deposit(int, int) [56]
                0.00    0.00       1/10          Bank::withdraw(int, int) [57]
                0.00    0.00       1/10          Bank::delete_account(int) [52]
                0.00    0.00       2/10          Bank::display_all() [50]
                0.00    0.00       4/10          Bank::pin_verify(int) [34]
[29]     0.0    0.00    0.00      10         std::map<int, Bank, std::less<int>, std::allocat
                0.00    0.00      10/28          std::_Rb_tree<int, std::pair<int const, Bank
std::less<int>, std::allocator<std::pair<int const, Bank> > >::end() [19]
-----------------------------------------------
                0.00    0.00       1/10          Bank::balance_enquiry(int) [53]
                0.00    0.00       1/10          Bank::deposit(int, int) [56]
                0.00    0.00       1/10          Bank::withdraw(int, int) [57]
                0.00    0.00       1/10          Bank::delete_account(int) [52]
                0.00    0.00       2/10          Bank::display_all() [50]
                0.00    0.00       4/10          Bank::pin_verify(int) [34]
[30]     0.0    0.00    0.00      10         std::operator!=(std::_Rb_tree_iterator<std::pair
```

END