Date :

Bank

```
import java.util.Scanner;

Class Amount
{
    String customerName;
    long amountNumber;
    String amountType;
    double balance;

    public Amount(String customerName, long
    amountNumber, String amountType, double balance)
    {
        this.customerName = customerName;
        this.amountNumber = amountNumber;
        this.amountType = amountType;
        this.balance = balance;
    }

    public void deposit(double amount)
    {
        balance += amount;
        System.out.println("Deposit successful
        Updated balance: " + balance);
    }
```

Date :

```
public void displayBalance()
{
    System.out.println("Account Number: " + accountNumber);
    System.out.println("Customer Name: " + customerName);
    System.out.println("Account Type: " + accountType);
    System.out.println("Balance: " + balance);
}

class SavAct extends Account
{
    public SavAct(String customerName, long accountNumber, double balance)
    {
        super(customerName, accountNumber, "Saving", balance);
    }

    public void computeAndDepositInterest(double rate)
    {
        double interest = balance * rate / 100;
        balance += interest;
        System.out.println("Interest computed & deposited. Updated balance: " + balance);
    }
```

public void withdraw(double amount)
{
if(amount <= balance)
balance -= amount;
System.out.println("Withdrawal successful.
Updated balance: " + balance);
}
else
{
System.out.println("Insufficient funds.
Withdrawal failed.");
}
}
}

Class CurrAct extends Account
{
double minimumBalance;
double ServiceCharge;

public CurrAct(String customerName, long
accountNumber, double balance, double
minimumBalance, double ServiceCharge)

Date :

```
{
    super(customerName, accountNumber, "Current", balance);
    this.minimumBalance = minimumBalance;
    this.serviceCharge = serviceCharge;
}

private void checkMinimumBalance()
{
    if (balance < minimumBalance)
    {
        balance -= serviceCharge;
        System.out.println("Minimum balance not maintained. Service charge imposed. Updated balance: " + balance);
    }
}

public void withdraw(double amount)
{
    if (amount <= balance)
    {
        balance -= amount;
        System.out.println("Withdrawal successful. Updated balance: " + balance);
        checkMinimumBalance();
    }
```

```
else
{
    System.out.println("Insufficient funds. Withdrawal failed");
}
}
}
public class Bank
{
    public static void main(String[] args)
{
    Scanner SI = new Scanner(System.in);
    System.out.print("Enter Customer name for saving account:");
    String SCN = SI.nextLine();
    System.out.print("Enter initial balance for saving account");
    String SAN = SI.nextLong();
    System.out.print("Enter initial balance for savings account:");
    double SIB = SI.nextDouble()
    SavAcct SA = new SavAcct(SCN, SAN, SIB);
```

Date :

System.out.print("Enter custom name for Current Account: ");
String CCN = S1.next();
System.out.print("Enter account number for Current Account: ");
String CCN = S1.next();
System.out.print("Enter initial balance for Current Account: ");
double CIB = S1.nextDouble();
System.out.print("Enter minimum balance for Current Account: ");
double MB = S1.nextDouble();
System.out.print("Enter service charge for Current Account: ");
double SC = S1.nextDouble();
CurrAct CA = new CurrAct(CCN, CAN, CIB, MB, SC);
System.out.print("Enter deposit amount for Savings Account: ");
double SD A = S1.nextDouble();
SA.deposit(SDA)

System.out.print("Enter interest rate for Savings Account: ");

double SIR = SI.nextDouble();
SA.computeAndDepositInterest(SIR);

System.out.print("Enter withdraw amount for saving Account:");
double SWA=SI.nextDouble();
SA.withdraw(SWA);

System.out.print("Enter deposit amount for current Account:");
double CDA=SI.nextDouble();
CA.deposit(CDA);

System.out.print("Enter withdraw amount for current Account:");
double CWA=SI.nextDouble();
CA.withdraw(CWA);
System.out.println("\nFinal Balance:");
System.out.println("Saving Account");
SA.display Balance();
System.out.println("\nCurrent Account:");
CA.display Balance();
}
}

8/11/24

Deekshith B
1BM22CS082

```
import java.util.Scanner;


class Account
{
String customerName;
long accountNumber;
String accountType;
double balance;

Account(String customerName, long accountNumber, String accountType, double balance)
{
this.customerName = customerName;
this.accountNumber = accountNumber;
this.accountType = accountType;
this.balance = balance;
}

void deposit(double amount)
{
balance = balance + amount;
System.out.println("Deposit of " + amount + " was successful. Balance: " + balance);
}

void displayBalance()
{
System.out.println("\nAccount Number: " + accountNumber + "\nCustomer Name: " +
customerName + "\nAccount Type: " + accountType + "\nBalance: " + balance);
}

}
```

```
class SavingsAccount extends Account
{

SavingsAccount(String customerName, long accountNumber, double balance)
{
super(customerName, accountNumber,"Savings", balance);
}

void interest(double rate)
{
double interest = balance * rate / 100;
balance = balance + interest;
System.out.println("Interest computed and deposited. Updated balance: " + balance);
}



void withdraw(double amount)
{

if(amount<=balance)
{
balance = balance - amount;
System.out.println("Withdrawal of " + amount + " successful. Updated balance: " + balance);
}

else
{
System.out.println("Insufficient funds. Withdrawal failed.");
```

```
}

}

}
```

```
class CurrentAccount extends Account
{
double minimumBalance;
double serviceCharge;

CurrentAccount(String customerName, long accountNumber, double balance, double
minimumBalance, double serviceCharge)
{
super(customerName,accountNumber,"Current",balance);
this.minimumBalance=minimumBalance;
this.serviceCharge=serviceCharge;
}

void checkMinimumBalance()
{
if(balance<minimumBalance)
{
balance = balance - serviceCharge;
System.out.println("Minimum balance not maintained. Service charge imposed. Updated balance: "
+ balance);
}
else
{
```

```
System.out.println("Minimum balance maintained. Service charge not imposed. Updated balance: "
+ balance);
}
}

void cheque(double amount)
{


balance = balance - amount;
System.out.println("Withdrawal of " + amount + " successful. Updated balance: " + balance);
}

}


public class Bank
{
   public static void main(String[] args)
{
      Scanner s1 = new Scanner(System.in);

System.out.println("Savings Account: ");
      System.out.print("Enter customer name: ");
      String name = s1.nextLine();
      System.out.print("Enter account number: ");
      long no = s1.nextLong();
      System.out.print("Enter initial balance: ");
      double balance = s1.nextDouble();
      SavingsAccount SA = new SavingsAccount(name, no, balance);
System.out.print("\n");
```

```
System.out.println("Current Account: ");
        System.out.print("Enter customer name: ");
        name = s1.next();
        System.out.print("Enter account number: ");
        no = s1.nextLong();
        System.out.print("Enter balance: ");
        balance = s1.nextDouble();
        System.out.print("Enter minimum balance: ");
        double minBalance = s1.nextDouble();
        System.out.print("Enter service charge: ");
        double charge = s1.nextDouble();
        CurrentAccount CA = new CurrentAccount(name, no, balance, minBalance, charge);
System.out.print("\n");


        System.out.print("Enter deposit amount for Savings Account: ");
        double SDA = s1.nextDouble();
        SA.deposit(SDA);
System.out.print("\n");

        System.out.print("Enter interest rate for Savings Account: ");
        double SIR = s1.nextDouble();
        SA.interest(SIR);
System.out.print("\n");

        System.out.print("Enter withdrawal amount for Savings Account: ");
        double SWA = s1.nextDouble();
        SA.withdraw(SWA);
System.out.print("\n");
```

```
System.out.print("Enter deposit amount for Current Account:");
double CDA $1.nextDouble()
CA.deposit (CDA);
System.out.print("\n”);
System.out.print("Enter withdrawal amount for Current Account: ");
double CWA= s1.nextDouble():
CA.cheque(CWA);
System.out.print("\n”);
System.out.println("\nFinal Balances:");
System.out.println("Savings Account:"
SA displayBalance():
System.out.print("\n");
System.out.presin("InCurrent Account");
CA displayBalance();
}
}
```

Deekshith B 1BM22CS082

Output:

```
Savings Account:
Enter customer name: Deekshith B
Enter account number: 1000
Enter initial balance: 50000

Current Account:
Enter customer name:Nandan
Enter account number: 2000
Enter balance 150000
Enter minimum balance: 20000
Enter service charge: 10
```

Current Account:

Enter customer name:Nandan

Enter account number: 2000

Enter balance 150000

Enter minimum balance: 20000

Enter service charge: 10

Enter deposit amount for Savings Account: 10000

Deposit of 10000.0 was successful. Balance: 500000

Enter interest rate for Savings Account: 5

Interest computed and deposited. Updated balance: 63000.0

Enter withdrawal amount for Accoun Savings 3000

Withdrawal of 2000.0 successful, Updated balance: 610000

Enter deposit amount for Current Account: 20000

Deposit of 20000.0 was successful Balance: 1700000

Enter withdrawal amount for Current Account: 50000

Withdrawal of 100000 successful. Updated balance: 120000 0

Final Balances:

Savings Account

Account Number: 1000

Customer Name: Deekshith B

Account Type: Savings Balance: 61000.

Current Account:

Account Number: 2000

Customer Name: Nandan

Account Type: Current

Balance: 120000.0