Program 5 :-

Q Execute a program to demonstrate usage of get and set methods :-

→
```
import java.util.Scanner
class account {
    String cust_name;
    int accno;
    String acc_type;
    double balance;
    public account (String cust_name, int accno, String acc_type)
    {   this.cust_name = cust_name;
        this.accno = accno;
        this.acc_type = acc_type;
        this.balance = 0;
    }
    public void displayBal() {
        System.out.println("Acc no." + accno);
        System.out.println("Cust. no." + cust_name);
        System.out.println("Acc type" + acc_type);
        System.out.println("Balance" + balance);
    }
}
class Current extends account { double min_bal, service_charge;
    Current (String cust_name, int accno) { super(cust_name, accno,
    "current");
    this.min_bala = 500;
    this.service_charge = 20;
    }
```

```
public void withdrawal(double amt) {
    if(balance - amt >= min_balance) {
        balance -= amt; System.out.println("Withdrawal success. Current balance = " + Current...
    else
        System.out.println("Withdrawal not possible due to insufficient funds"); } }

public void Servicecharge() {
    if(balance < min_balance) {
        balance -= Service_charge;
        System.out.println("Service charge imposed. Current balance
            + balance); }
    }
}
class Savings extends Account { double interest_rate; Savings(String
    cust_name, int accno) { super(cust_name, accno, "Savings
    this.interest_rate = 0.5;
    }
public void DepositInterest() { balance += balance * interest_rate
    System.out.println("Interest deposited. Current Balance :"
        + balance); }
public void CompoundInterest(double initial_amt, int time) {
    double ci = initial_amt * Math.pow((1 + interest_rate), time);
    balance += ci;
    System.out.println("Compound interest applied Current
    balance" + balance); }
}
public class f {
    public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    System.out.println("1. Savings");
    System.out.println("2. Current");
    System.out.println("Enter choice 1 or 2");
```

```
int choice = in.nextInt();
System.out.println("Enter Customer name:");
String cust_name = in.next();
System.out.println("Enter account number:");
int accno = in.nextInt();
if(choice == 1){
    Savings SavAcc = new Savings(cust_name, accno);
    System.out.println("Enter initial balance");
    double initial_balance = in.nextDouble();
SavAcc.balance = initial_balance;
System.out.println("Enter withdrawal amt");
double withdrawal = in.nextDouble();
    SavAcc.balance -= withdrawal;
    System.out.println("withdrawal successful. Current
      balance" + SavAcc.balance);
  System.out.println("Enter interest rate:");
  double interest_rate = in.nextDouble();
SavAcc.interest_rate = interest_rate;
    SavAcc.displayBal();
    System.out.println("Enter time(in yrs) to calculate
    compound interest");
      int time = in.nextInt();
      SavAcc.compoundInterest(initial_Balance, time);
      SavAcc.displayBal();
}
else if(choice == 2){
      Current curAcc = new Current(cust_name, accno);
      System.out.println("Enter initial balance:");
      double initial_balance = in.nextDouble();
```

```
curAcc.balance = initial_balance;
        System.out.println("Enter withdrawal amt");
            double amt = in.nextDouble();
            curAcc.withdrawal(amt);
            curAcc.servicecharge();
            curAcc.displayBal();
        }
        else{
            System.out.println("Invalid choice"); }
                } }
```

Algorithm :-

1. Start.
2. Create a class named account
3. Create variable to store the account holdiss details in account class
4. Create a constructor named account.
5. Create a func^n to display balance
6. Create a class current with super class as account
7. Create a constructor
8. Create a function for withdrawal and for displaying balance & service charge
9. Create savings class with super class account
10. Create function to calculate deposit interest & Compound Interest
11. Create a main function and create the necessary variables to take input from the ~~variables~~ user and pass them to functions as and when required
12. Stop.

Output:

1 Choose account type:
1. Current
2 Savings
Enter choice (1 or 2): 1
Enter customer name: Anagh
Enter initial balance: 500000
Enter withdrawal amount: 2900
Withdrawal successful. Current balance: 497100
Account number = 200570
Customer name = Anagh
Account type = current
Balance = 497100.0

2 Choose account type
1. Current
2. Savings
Enter choice (1 or 2): 2
enter customer name: Anagh
enter account no.: 200510
Enter initial balance: 40000
Enter withdrawal amt: 500
Withdrawal Successful, Current Balance: 39500.0
Enter inbuilt Kali: 5
Account Number: 5
Customer name: Anagh
Account type: savings
Balance: 39500.0
Enter Term (in yrs) for compd interest calculations: 4
Comp interest deposited, Current balance: 5.1838
Account number: 200510
Account type: savings
Balance: 5.18

19/01/24

```
C:\Users\abdes\OneDrive\Documents\Java>javac java_code/Bank.java

C:\Users\abdes\OneDrive\Documents\Java>java java_code.Bank
Anagh.B.Deshpande(1BM22CS037)
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 1
Enter customer name: ad
Enter account number: 200510
Enter initial balance: $50000
Enter withdrawal amount: $2905
Withdrawal successful. Current Balance: $47095.0
Account Number: 200510
Customer Name: ad
Account Type: Current
Balance: $47095.0

C:\Users\abdes\OneDrive\Documents\Java>java java_code.Bank
Anagh.B.Deshpande(1BM22CS037)
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 2
Enter customer name: ad
Enter account number: 200510
Enter initial balance: $40000
Enter withdrawal amount: $500
Withdrawal successful. Current Balance: $39500.0
Enter interest rate: 5
Account Number: 200510
Customer Name: ad
Account Type: Savings
Balance: $39500.0
Enter term (in years) for compound interest calculation: 4
Compound Interest deposited. Current Balance: Rs.5.18395E7
Account Number: 200510
Customer Name: ad
Account Type: Savings
Balance: $5.18395E7
```