EXECUTE IN BANK TRANSACTIONS SUCH AS VIEW BALANCE, DEPOSIT, WITHDRAWAL AND MONEY TRANSFER,ETC..... BY USING SINGLE LEVEL INHERITANCE. NOTE: INPUTS: USER INPUTS SHOULD BE RECIEVED IN BASE CLASS. PROCESS SHOULD BE DONE IN THE DERIVED CLASS. RESULT SHOULD BE DONE IN MAIN CLASS.

```
import java.util.Arrays;
import java.util.Scanner;
class Bank {
    int inputDeposit() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Deposit amount: ");
        int da = sc.nextInt();
        return da;
    }
    int inputWithdraw() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Withdraw amount: ");
        int wa = sc.nextInt();
        return wa;
    }
    int inputTransfer() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Transfer amount: ");
        int ta = sc.nextInt();
        return ta;
    }
}
class Transaction extends Bank {
    int deposit(int ta, int da) {
        return ta + da;
    }
    int withdraw(int ta, int wa) {
        return ta - wa;
    }
    int transfer(int ta, int tra) {
        return ta - tra;
    }
}
public class Main {
    public static void main(String[] args) {
        int total = 1000;
        int[] accountNumbers = {1001, 1002, 1011, 1012};
        Arrays.sort(accountNumbers);
        Scanner sc = new Scanner(System.in);
        Transaction t = new Transaction();
        int choice;
        while (true) {
            System.out.println("\nBank balance: " + total);
            System.out.println("Enter choice: Deposit = '1', Withdraw = '2', Transfer =
'3', Exit = '4'");
            choice = sc.nextInt();
            if (choice == 1) {
                int depAmou = t.inputDeposit();
                total = t.deposit(total, depAmou);
                System.out.println("Before Deposit: " + (total - depAmou));
                System.out.println("After Deposit: " + total);
            } else if (choice == 2) {
```

```
    int witAmou = t.inputWithdraw();

    if (witAmou > total) {

        System.out.println("Insufficient amount");

    } else {

        total = t.withdraw(total, witAmou);

        System.out.println("Before Withdraw: " + (total + witAmou));

        System.out.println("After Withdraw: " + total);

    }

} else if (choice == 3) {

    System.out.print("Enter account number to transfer to: ");

    int accNum = sc.nextInt();

    if (Arrays.binarySearch(accountNumbers, accNum) >= 0) {

        int traAmou = t.inputTransfer();

        if (traAmou > total) {

            System.out.println("Insufficient amount");

        } else {

            total = t.transfer(total, traAmou);

            System.out.println("Before Transfer: " + (total + traAmou));

            System.out.println("After Transfer: " + total);

        }

    } else {

        System.out.println("Invalid account number");

    }

} else if (choice == 4) {

    System.out.println("Exiting");

    break;

} else {

    System.out.println("Invalid choice");
```

```
            }
        }
    }
}
```

CREATE A ACCOUNT WITH A MINIMUM BALANCE OF 2000 IN THE BASE CLASS. IN ONE OF THE DERIVED CLASS, TRY TO TRANSFER THE AMOUNT 5000. TO YOUR FRIEND. IN ANOTHER DERIVED CLASS TRY TO WITHDRAW THE AMOUNT BY MAKING YOUR ACCOUNT BALANCE AS ZERO. USE MULTILEVEL INHERITENCE.

```
import java.util.Scanner;
class Account {
    protected int balance;
    public Account(int initialBalance) {
        if (initialBalance < 2000) {
            System.out.println("Minimum balance should be 2000. Setting balance to 2000.");
            this.balance = 2000;
        } else {
            this.balance = initialBalance;
        }
    }
    public void displayBalance() {
        System.out.println("Current balance: " + balance);
    }
}
class Transaction extends Account {
    public Transaction(int initialBalance) {
        super(initialBalance);
    }
    public void deposit(int amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    }
    public void withdraw(int amount) {
        if (balance - amount < 0) {
            System.out.println("Insufficient funds.");
        } else {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        }
    }
    public void transfer(int amount) {
        if (balance - amount < 0) {
            System.out.println("Insufficient funds.");
        } else {
            balance -= amount;
            System.out.println("Transferred: " + amount);
        }
    }
}
class Transfer extends Transaction {
    public Transfer(int initialBalance) {
        super(initialBalance);
    }
    public void performTransfer() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter amount to transfer: ");
        int amount = scanner.nextInt();
        transfer(amount);
    }
}
class Withdraw extends Transaction {
    public Withdraw(int initialBalance) {
        super(initialBalance);
    }
    public void performWithdraw() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter amount to withdraw: ");
        int amount = scanner.nextInt();
        withdraw(amount);
    }
}
public class Main {
    public static void main(String[] args) {
        int initialBalance = 2000;
        Transfer transferAccount = new Transfer(initialBalance);
        Withdraw withdrawAccount = new Withdraw(initialBalance);
        Scanner scanner = new Scanner(System.in);
        int choice;
        while (true) {
            transferAccount.displayBalance();
            System.out.println("Enter choice: Transfer = 1, Withdraw = 2, Exit = 3");
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    transferAccount.performTransfer();
                    break;
                case 2:
                    withdrawAccount.performWithdraw();
                    break;
                case 3:
                    System.out.println("Exiting.");
                    return;
                default:
                    System.out.println("Invalid choice.");
            }
        }
    }
}
```

(AC NO, INT BAL, DEP , DBAL, WITH, WITHBAL, TRANN, TRA, TBAL)ASSUME THAT YOU ARE GOING TO START A BANK WITH, FOUR BRANCHES. BRANCH 1: CREATE ACCOUNT. BRANCH2: DEPOSIT. BRANCH 3:WITHDRAWAL, BRANCH 4 : TRANFER.

```
import java.util.Arrays;
import java.util.Scanner;
class Bank {
    int inputDeposit() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Deposit amount: ");
        int da = sc.nextInt();
        return da;
    }
    int inputWithdraw() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Withdraw amount: ");
        int wa = sc.nextInt();
        return wa;
    }
    int inputTransfer() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Transfer amount: ");
        int ta = sc.nextInt();
        return ta;
    }
}
class Transaction extends Bank {
    int deposit(int ta, int da) {
        return ta + da;
    }
    int withdraw(int ta, int wa) {
        return ta - wa;
    }
    int transfer(int ta, int tra) {
        return ta - tra;
    }
}
class Deposit extends Transaction {
    void performDeposit(int total) {
        int depAmou = inputDeposit();
        total = deposit(total, depAmou);
        System.out.println("Before Deposit: " + (total - depAmou));
        System.out.println("After Deposit: " + total);
    }
}
class Withdraw extends Transaction {
    void performWithdraw(int total) {
        int witAmou = inputWithdraw();
        if (witAmou > total) {
            System.out.println("Insufficient amount");
        } else {
            total = withdraw(total, witAmou);
            System.out.println("Before Withdraw: " + (total + witAmou));
            System.out.println("After Withdraw: " + total);
        }
    }
}
class Transfer extends Transaction {
    void performTransfer(int total, int[] accountNumbers) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter account number to transfer to: ");
        int accNum = sc.nextInt();
        if (Arrays.binarySearch(accountNumbers, accNum) >= 0) {
            int traAmou = inputTransfer();
            if (traAmou > total) {
                System.out.println("Insufficient amount");
            } else {
                total = transfer(total, traAmou);
                System.out.println("Before Transfer: " + (total + traAmou));
                System.out.println("After Transfer: " + total);
            }
        } else {
            System.out.println("Invalid account number");
        }
    }
}
class ShowBalance extends Transaction {
    void displayBalance(int total) {
        System.out.println("\nBank balance: " + total);
    }
}
public class Main {
    public static void main(String[] args) {
        int total = 1000;
        int[] accountNumbers = {1001, 1002, 1011, 1012};
        Arrays.sort(accountNumbers);
        Scanner sc = new Scanner(System.in);
        Deposit deposit = new Deposit();
        Withdraw withdraw = new Withdraw();
        Transfer transfer = new Transfer();
        ShowBalance showBalance = new ShowBalance();
        int choice;
        while (true) {
            showBalance.displayBalance(total);
            System.out.println("Enter choice: Deposit = '1', Withdraw = '2', Transfer = '3', Exit = '4'");
            choice = sc.nextInt();
            if (choice == 1) {
                deposit.performDeposit(total);
            } else if (choice == 2) {
                withdraw.performWithdraw(total);
            } else if (choice == 3) {
                transfer.performTransfer(total, accountNumbers);
            } else if (choice == 4) {
                System.out.println("Exiting");
                break;
            } else {
                System.out.println("Invalid choice");
            }
        }
    }
}
```

CONSIDER A STUDENT WHO COMPLETED 10TH STANDARD AND COMPLETED 12TH. DISPLAY THE MARKS, AND STU DETAILS.(NAME, REG NO, TOTAL AVG.) 12 IN ONE BASE CLASS AND 10TH IN ANOTHER BASE CLASS. USE INTERFACE.

```
import java.util.Scanner;
interface TenthStandard {
    void display10thDetails();
}
interface TwelfthStandard {
    void display12thDetails();
}
class Student {
    String name;
    int regNo;
    Student(String name, int regNo) {

        this.name = name;

        this.regNo = regNo;

    }

}

class TenthStandardStudent implements TenthStandard {

    Student student;

    int[] marks;

    TenthStandardStudent(String name, int regNo, int[] marks) {

        this.student = new Student(name, regNo);

        this.marks = marks;

    }

    public void display10thDetails() {

        System.out.println("Name: " + student.name);

        System.out.println("Registration No: " + student.regNo);

        System.out.println("10th Standard Marks:");

        for (int i = 0; i < marks.length; i++) {

            System.out.println("Subject " + (i + 1) + ": " + marks[i]);

        }

    }

}

class TwelfthStandardStudent implements TwelfthStandard {

    Student student;

    int[] marks;

    TwelfthStandardStudent(String name, int regNo, int[] marks) {

        this.student = new Student(name, regNo);

        this.marks = marks;

    }
    public void display12thDetails() {
        System.out.println("Name: " + student.name);
        System.out.println("Registration No: " + student.regNo);
        System.out.println("12th Standard Marks:");
        for (int i = 0; i < marks.length; i++) {
            System.out.println("Subject " + (i + 1) + ": " + marks[i]);
        }
    }
}
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter name for 10th Standard Student: ");
        String name10th = scanner.nextLine();
        System.out.print("Enter registration number for 10th Standard Student: ");
        int regNo10th = scanner.nextInt();
        System.out.print("Enter number of subjects for 10th Standard Student: ");
        int numSubjects10th = scanner.nextInt();
        int[] marks10th = new int[numSubjects10th];
        for (int i = 0; i < numSubjects10th; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks10th[i] = scanner.nextInt();
        }
        TenthStandardStudent student10th = new TenthStandardStudent(name10th, regNo10th, marks10th);
        scanner.nextLine();
        System.out.print("Enter name for 12th Standard Student: ");
        String name12th = scanner.nextLine();
        System.out.print("Enter registration number for 12th Standard Student: ");
        int regNo12th = scanner.nextInt();
        System.out.print("Enter number of subjects for 12th Standard Student: ");
        int numSubjects12th = scanner.nextInt();
        int[] marks12th = new int[numSubjects12th];
        for (int i = 0; i < numSubjects12th; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks12th[i] = scanner.nextInt();
        }
        TwelfthStandardStudent student12th = new TwelfthStandardStudent(name12th, regNo12th, marks12th);
        student10th.display10thDetails();
        System.out.println();
        student12th.display12thDetails();

        scanner.close();
    }
}
```