

```
//-----//
/*
 * Account Holder (1)
 * Methods : Deposit, Withdraw, Display
 */

public class AccountHolder {
    private int accno;
    private String name;
    private float bal;
    //Add getters, setters & constructors if needed

    public void deposit(float amount) {
        this.bal = this.bal + amount;
        System.out.println("!!..Amount Deposited..!!");
    }
    public void withdraw(float amount) {
        if (this.bal == 0.0f) {
            System.out.println("!!..No Balance/Cannot Withdraw..!!");
        } else if (this.bal < amount) {
            System.out.println("!!..Low Balance..!!");
            System.out.println("Amount > Balance");
        } else {
            this.bal = this.bal - amount;
            System.out.println("!!..Amount Withdrawn..!!");
        }
    }
    public void display() {
        System.out.println("Account Number : " + accno);
        System.out.println("Name : " + name);
        System.out.println("Account Balance : " + bal);
    }
}

//-- MAIN --//

import java.util.Scanner;

public class ThisIsMain {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        AccountHolder[] a1 = new AccountHolder[5];
        for (int i = 0; i < a1.length; i++) {
```

```

        a1[i] = new AccountHolder();
    }
    int choice, ac, point = 0;
    String str;
    float balance, amnt;
    while (true) {
        System.out.println("Enter Choice : ");
        System.out.println("1. Add Record for Account Holder ");
        System.out.println("2. Display all Record ");
        System.out.println("3. Deposit in a particular account ");
        System.out.println("4. Withdraw from a particular account ");
        System.out.println("5. Exit ");
        choice = scan.nextInt();
        switch (choice) {
            case 1:
                if (point == 5) {
                    System.out.println("!!..Account Full..!!");
                    System.exit(0);
                } else {
                    System.out.println("Enter Account Number : ");
                    ac = scan.nextInt();
                    a1[point].setaccno(ac);
                    System.out.println("Enter Account Name : ");
                    str = scan.next();
                    a1[point].setName(str);
                    System.out.println("Enter Account Balance : ");
                    balance = scan.nextFloat();
                    a1[point].setBal(balance);
                    point++;
                }
                break;
            case 2:
                point = 0;
                for (int a = point; a < 5; a++) {
                    a1[a].display();
                }
                break;
            case 3:
                System.out.println("Enter the Account Number : ");
                ac = scan.nextInt();
                System.out.println("Enter Amount to be Deposit : ");
                amnt = scan.nextFloat();
                a1[ac].deposit(amnt);

```

```

        break;
    case 4:
        System.out.println("Enter the Account Number : ");
        ac = scan.nextInt();
        System.out.println("Enter Amount to be Withdraw : ");
        amnt = scan.nextFloat();
        a1[ac].withdraw(amnt);
        break;
    case 5:
        scan.close();
        System.exit(0);
    }
}

}

}

//-----//

/*
 * Student Record
 * Counting total number of objects created
 * Methods : Display using "toString"
 */

public class Student {

    static int count;    //To Count number of Objects
    private int rollno;
    private String name;
    private float percentage;
    //Add getters, setters & constructors if needed

    public void print() {
        System.out.println("Roll Number : " + rollno);
        System.out.println("Name : " + name);
        System.out.println("Percentage : " + percentage);
    }

    @Override
    public String toString() {
        return "Student [rollno=" + rollno + ", name=" + name + ", percentage=" +
percentage + "]";
    }
}

```

```
}
```

```
//-- MAIN --//
```

```
import java.util.Scanner;
```

```
public class ThisIsMain {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int no;
```

```
        System.out.println("Enter the Total Number of Student : ");
```

```
        no = sc.nextInt();
```

```
        Student[] students = new Student[no]; // 'no' of classes
```

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

```
            students[i] = new Student();
```

```
        }
```

```
        Student st = new Student(23, "John_Cena", 98);
```

```
        //Another class for checking static count
```

```
        System.out.println("Students Details : ");
```

```
        System.out.println("Total Student Object Count : " + Student.count);
```

```
        for (Student stu : students) {
```

```
            stu.print();
```

```
        }
```

```
        st.print();
```

```
        sc.close();
```

```
    }
```

```
}
```

```
//-----//
```

```
/*
```

```
 * Employee Details
```

```
 * Inheritance (SalesPerson+WageEmployee+Employee => Main)
```

```
*/
```

```
public class Employee {
```

```
    protected int id;
```

```
    protected String name;
```

```
    protected String dob;
```

```
    //Add getters, setters & constructors if needed
```

```
    public void display() {
```

```

        System.out.println("Employee ID : " +id);
        System.out.println("Employee Name : " +name);
        System.out.println("Employee's Date of Birth : " +dob);
    }
}

//----->>

public class WageEmployee extends Employee {
    protected int hours;
    protected float rate;
    //Add getters, setters & constructors if needed

    public void display() {
        super.display();
        System.out.println("Number of Hours Worked : " + hours);
        System.out.println("Rate per hour : " + rate);
    }
    public float salary() {
        return (hours * rate);
    }
}

//----->>

public class SalesPerson extends WageEmployee {
    protected int sold;
    protected float commission;
    //Add getters, setters & constructors if needed
    public void display() {
        super.display();
        System.out.println("Number of Items Sold : " + sold);
        System.out.println("Commission per Item : " + commission);
    }
    public float salary() {
        return (super.salary() + (sold * commission));
    }
}

//-- MAIN --//

import java.util.Scanner;

```

```

public class ThisIsMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Employee e1 = new SalesPerson(1, "John", "3-9-1956", 5, 10f, 50,
100f); //Polymorphic references
        Employee e2 = new SalesPerson(1, "XYZ", "01-11-1990", 2, 20f, 10, 200f);
        System.out.println("Sales Person 1 Details -- ");
        e1.display();
        System.out.println("Salary : " + ((SalesPerson) e1).salary());
        System.out.println("\nSales Person 2 Details -- ");
        e2.display();
        System.out.println("Salary : " + ((SalesPerson) e2).salary());
        sc.close();
    }
}

```

```

}

```

```

//-----//

```

```

/*
 * Cricket and Football Player
 * Using : Interface (Printable)
 */

```

```

public class CktPlayer implements Printable {
    String name;
    int runs;
    //Add getters, setters & constructors if needed

    public void printDetail() {
        System.out.println("Detail of Cricket Player : ");
        System.out.println("Name : " + name + "\nRuns : " + runs);
    }
}

```

```

//----->>

```

```

public class FtPlayer implements Printable {
    String name;
    int goals;
    //Add getters, setters & constructors if needed

```

```

        public void printDetail() {
            System.out.println("Detail of Football Player : ");
            System.out.println("Name : " + name + "\nGoals : " + goals);
        }
    }

//----->>

public interface Printable {
    public void printDetail();
}

//-- MAIN --//

public class ThisIsMain {
    public static void main(String[] args) {
        CktPlayer cc = new CktPlayer("AAA", 1000);
        FtPlayer ff = new FtPlayer("BBB", 100);
        cc.printDetail();
        ff.printDetail();
    }
}

//-----//

/*
 * Account Holder (2)
 * Error Handling
 * Methods : deposit(Overlimit (max. 15000),
 *             withdraw(Overlimit (max. 15000), Insufficient Balance(amount>balance))
 */

public class Account {
    private int accno;
    private float bal;
    //Add getters, setters,constructors

    public void deposit(float amount) throws OverLimit {
        if (amount > 15000f)
            throw new OverLimit();
        this.bal = this.bal + amount;
        System.out.println(amount + " Deposited..!!\nAvailable Balance : " + bal);
    }
}

```

```

        public void withdraw(float amount) throws OverLimit, InsufficientBalance {
            if (amount > 15000f)
                throw new OverLimit();
            if (amount > bal) {
                throw new InsufficientBalance(bal);
            }
            bal = bal - amount;
            System.out.println(amount + " Withdrawn..!!\nAvailable Balance : " + bal);
        }
        public void display() {
            System.out.println("Account Number : " + accno);
            System.out.println("Account Balance : " + bal);
        }
    }
}

```

//----->>

```

public class InsufficientBalance extends Exception {
    public InsufficientBalance(float ff) {
        System.out.println("Balance is Low");
        System.out.println("Available Balance : " + ff);
        System.out.println("Error Caused coz of Insufficient Balance");
    }
}

```

//----->>

```

public class OverLimit extends Exception {
    public OverLimit() {
        System.out.println("Limit is 15000");
    }
    public void printexception() {
        System.out.println("Error Caused coz of Over Limit");
    }
}

```

//-- MAIN --//

```

public class ThisIsMain {
    public static void main(String[] args) {
        Account one = new Account(1, 15000f);
        try {
            one.deposit(5000);

```



```

        one.withdraw(10000);
        one.withdraw(11000);
    } catch (OverLimit e) {
    } catch (InsufficientBalance e) {
    }
}

//-----//

/*
 * Account Holder (3)
 * Array-List
 */

import java.util.Scanner;

public class Employee {
    private int id;
    private String name;
    private float sal;
    //add getters, setters and constructors if needed
    public void getdata() {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter ID ");
        this.id = scan.nextInt();
        System.out.println("Enter Name : ");
        this.name = scan.nextLine() + scan.nextLine();
        System.out.println("Enter Salary : ");
        this.sal = scan.nextFloat();
    }
    //add 'toString' if needed
}

//-- MAIN --//

import java.util.ArrayList;
import java.util.Scanner;

public class ThisIsMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice, idd;

```

```

        ArrayList<Employee> arls = new ArrayList<>();

        while (true) {
            System.out.println("Enter a choice : ");
            System.out.println("1.Add Employee Information\t2.Edit Employee
Information");

            System.out.println("3.Display all\t4.Exit");
            choice = sc.nextInt();
            switch (choice) {
                case 1:
                    Employee e = new Employee();
                    e.getdata();
                    arls.add(e); // Add the Object 'e' to the Array List
                    break;
                case 2:
                    System.out.println("Enter Employee ID : ");
                    idd = sc.nextInt();
                    for (Employee emp : arls) {
                        if (idd == (emp.getId())) {
                            emp.getdata();
                        }
                    }
                    break;
                case 3:
                    for (Employee emp : arls) {
                        System.out.println(emp.toString());
                    }
                    break;
                case 4:
                    System.exit(1);
            }
        }
    }

    //-----//

    /*
    * Thread (Multithreading)
    */

    public class AddT1 extends Thread {

```

```

int num;
public AddT1(int num) {
    this.num = num;
}

@Override
public void run() {
    System.out.println("In Thread 1 : ");
    int i;
    for (i = 1; i <= 10; i++) {
        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.out.println("T1 : Increment - " + (num + i));
    }
}

}

//----->>

public class MultT2 extends Thread {

    int num;
    public MultT2(int num) {
        this.num = num;
    }

    @Override
    public void run() {
        System.out.println("In Thread 2 : ");
        int i;
        for (i = 1; i <= 10; i++) {
            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println("T2 : Multiplication - " + (num * i));
        }
    }
}

```

```
}
```

```
//-- MAIN --//
```

```
import java.util.Scanner;
```

```
public class Test {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter number");
```

```
        int n = sc.nextInt();
```

```
        AddT1 t1 = new AddT1(n);
```

```
        MulT2 t2 = new MulT2(n);
```

```
        t1.start();
```

```
        t2.start();
```

```
        sc.close();
```

```
    }
```

```
}
```