INHERITANCE

```
/*
Create a class ‘Person’ with data members Name, Gender, Address, Age
and a constructor to initialize the data members
*/

import java.util.Scanner;

class Person
{
    int age;
    String name,address,gender;

    Person(String name,int age,String gender,String address)
        {
            this.age = age;
            this.name = name;
            this.gender = gender;
            this.address = address;
        }
    void display1()
    {
        System.out.println("Name = "+name);
        System.out.println("Age = "+age);
        System.out.println("Address = "+address);
        System.out.println("Gender = "+gender);
    }
}

/*
another class
‘Employee’ that inherits the properties of class Person and also
contains its own data members like Empid, Company_name,
Qualification, Salary and its own constructor.
*/

class Employee extends Person
{
    int emp_id;
    String c_name,Qualification;
    float salary;

    Employee(String name,int age,String gender,String address,int emp_id,String c_name,String
Qualification,float salary)
    {
        super(name,age,address,gender);

        this.emp_id=emp_id;
        this.c_name=c_name;
        this.Qualification=Qualification;
        this.salary=salary;
    }

    void display2()
    {
        display1();
        System.out.println("Employee Id : "+emp_id);
        System.out.println("Company_name : "+c_name);
        System.out.println("Qualification : "+Qualification);
        System.out.println("Salary : "+salary);
    }
}

/*Create another class
'Teacher' that inherits the properties of class Employee and contains its
own data members like Subject, Department, Teacherid and also
contain constructors and methods to display the data members. Use
array of objects to display details of N teachers. */

class Teacher extends Employee
{
    String subject,Department;
    int t_id;

    Teacher(String name,int age,String gender,String address,int emp_id,String c_name,String Qualification,float salary,int t_id,String subject,String Department)
    {
        super(name,age,address,gender,emp_id,c_name,Qualification,salary);

        this.subject = subject;
        this.Department = Department;
        this.t_id=t_id;
    }
    void display3()
    {
        display2();
        System.out.println("Teacer ID : "+t_id);
        System.out.println("Subject : "+subject);
        System.out.println("Department : "+Department);
    }
}

class Main
{
    public static void main(String[] args)
    {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter the limit : ");
        int n = ob.nextInt();
        Teacher t[] = new Teacher[n];

        for(int i=0;i<n;i++)
        {
            System.out.println("Enter the name : ");
            String name = ob.next();
            System.out.println("Enter the Age : ");
            int age = ob.nextInt();
            System.out.println("Enter the Address : ");
            String address = ob.next();
            System.out.println("Enter the Gender : ");
            String gender = ob.next();
            System.out.println("Enter the Employee ID : ");
            int emp_id = ob.nextInt();
            System.out.println("Enter the Company Name : ");
            String c_name = ob.next();
            System.out.println("Enter the Qualification : ");
            String Qualification=ob.next();
            System.out.println("Enter the Salary : ");
            float salary = ob.nextFloat();
            System.out.println("Enter the Teacher ID : ");
            int t_id = ob.nextInt();
            System.out.println("Enter the Subject : ");
            String subject = ob.next();
            System.out.println("Enter the Department : ");
            String Department = ob.next();

            t[i] = new
Teacher(name,age,address,gender,emp_id,c_name,Qualification,salary,t_id,subject,Department);
            t[i].display3();
        }
    }
}



/*
Perform Multilevel inheritance.
Hint:- The base class ‘stud_details’ is created for assigning the Rol.no and name of the student
and display it.
The derived class “Marks” is created for inputting 3 subjects marks and display it. Again,
a derived class is created as “Total” for calculate the total mark of 3 subjects and display it.
Inputs are given through the keyboard.
*/

import java.util.*;
class stud_details
```

```
{
    int rno;
    String name;

    stud_details(int rno,String name)
    {
        this.rno = rno;
        this.name=name;
    }
    void display()
    {
        System.out.println("Roll no : "+rno);
        System.out.println("Name : "+name);
    }
}

class Marks extends stud_details
{
    int m1,m2,m3;

    Marks(int rno,String name,int m1,int m2,int m3)
    {
        super(rno,name);
        this.m1=m1;
        this.m2=m2;
        this.m3=m3;
    }
    void display2()
    {
        display();
        System.out.println("Mark 1 : "+m1);
        System.out.println("Mark 2 : "+m2);
        System.out.println("Mark 3 : "+m3);
    }
}

class Total extends Marks
{
    int totalMarks;
    Total(int rno,String name,int m1,int m2,int m3)
    {
        super(rno,name,m1,m2,m3);
    }
    void calculate()
    {
        totalMarks = m1 + m2 + m3;
    }
    void display3()
    {
        display2();
        System.out.println("Total Marks : "+totalMarks);
    }
}

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

        System.out.println("Enter Name : ");
        String name = ob.next();
        System.out.println("Enter Roll no : ");
        int rno = ob.nextInt();
        System.out.println("Enter Mark 1 : ");
        int m1 = ob.nextInt();
        System.out.println("Enter Mark 2 : ");
        int m2 = ob.nextInt();
        System.out.println("Enter Mark 3 : ");
        int m3 = ob.nextInt();

        Total t = new Total(rno,name,m1,m2,m3);
        t.calculate();
        t.display3();
    }
}
```

METHOD OVERRIDING

```
/*
Write a Java program to create a class Employee with a method called calculateSalary().
Create two subclasses Manager and Programmer. In
each subclass, override the calculateSalary() method to calculate and
return the salary based on their specific roles.
*/

class Employee
{
    int salary;

    Employee(int salary)
    {
        this.salary = salary;
    }

    int calculateSalary()
    {
        return salary;
    }
}

class Manager extends Employee
{
    int bonuses;
    Manager(int salary,int bonuses)
    {
        super(salary);
        this.bonuses = bonuses;
    }
    @Override
    int calculateSalary()
    {
        return bonuses + salary;
    }
}

class Programmer extends Employee
{
    int reward;
    Programmer(int salary,int reward)
    {
        super(salary);
        this.reward = reward;
    }
    @Override
    int calculateSalary()
    {
        return reward + salary;
    }
}

class Main
{
    public static void main(String[] args)
    {
        Manager m = new Manager(100000,200);
        Programmer p = new Programmer(20000,100);

        System.out.println("Salary of the Manager is : "+m.calculateSalary());
        System.out.println("Salary of the Manager is : "+p.calculateSalary());
    }
}
```

JAVA INTERFACE AND DYNAMIC DISPATCH

```
//Prepare the students mark list using inheritance and interface concepts.

import java.util.*;

interface Marks
{
    void calculateMarks();
}

class Person
{
    int rollno;
    String name;

    Person(int rollno,String name)
    {
        this.rollno=rollno;
        this.name=name;
    }
}

class Students extends Person implements Marks
{
    int p,m,c;

    Students(int rollno,String name,int p,int m,int c)
    {
        super(rollno,name);
        this.p=p;
        this.m=m;
        this.c=c;
    }

    public void calculateMarks()
    {
        int totalMarks=p+m+c;
        System.out.println("Name : "+name);
        System.out.println("Roll no : "+rollno);
        System.out.println("Total Marks : "+totalMarks);
    }
}

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

        System.out.println("Enter the Name : ");
        String name = ob.next();
        System.out.println("Enter the ROll no : ");
        int rollno = ob.nextInt();
        System.out.println("Enter the Physiscs : ");
        int p = ob.nextInt();
        System.out.println("Enter the Chemistry : ");
        int c = ob.nextInt();
        System.out.println("Enter the Maths : ");
        int m = ob.nextInt();

        Marks r = new Students(rollno,name,p,m,c);

        r.calculateMarks();
    }
}
```

/*

Create a class with Vehicle with serial no, type and name as instance variables and display the details. Create a subclass of Vehicle as Car with serial no, name and cost as instance variables and display the details. Implement runtime polymorphism (method overriding with dynamic method dispatch)

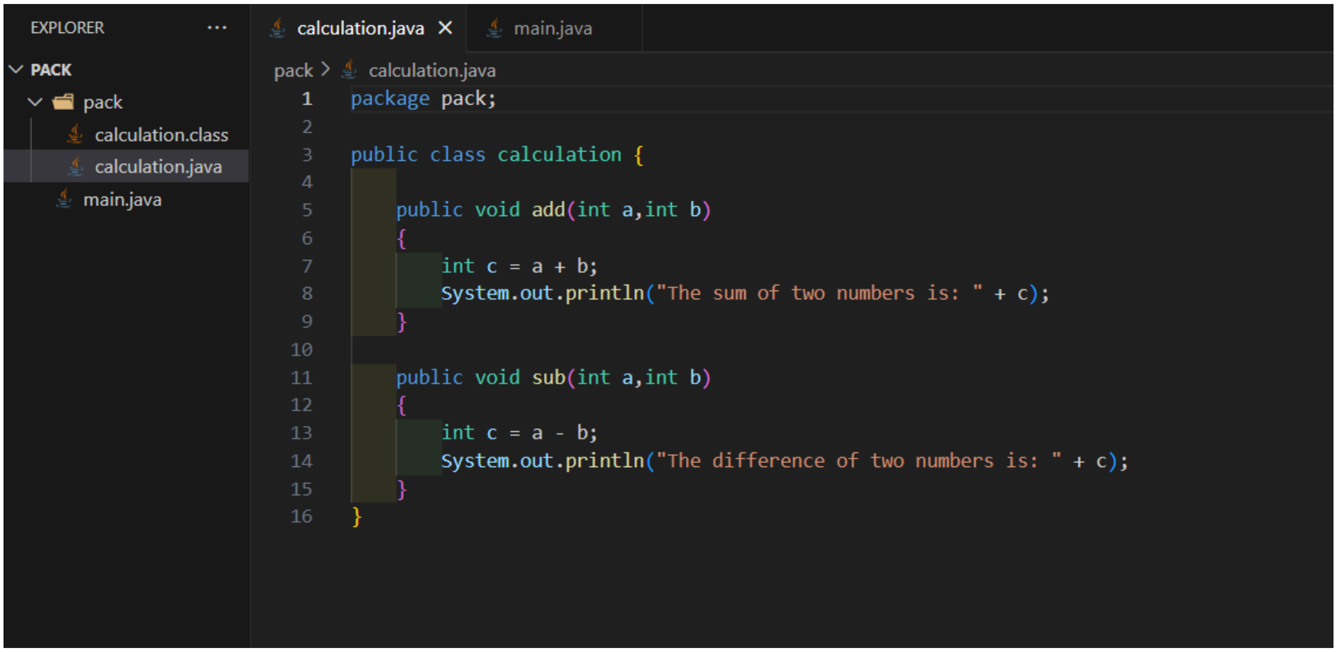
*/

```
import java.util.*;

class Vehicle
{
    int serial_no;
    String name,type;

    Vehicle(int serial_no,String name,String type)
    {
        this.serial_no=serial_no;
        this.name=name;
        this.type=type;
    }

    void display()
    {
        System.out.println("Serial No : "+serial_no);
        System.out.println("Name : "+name);
        System.out.println("Type : "+type);
    }
}

class Car extends Vehicle
{
    int cost;

    Car(int serial_no,String name,int cost)
    {
        super(serial_no,name,"Car");
        this.cost=cost;
    }
    @Override
    void display()
    {
        super.display();
        System.out.println("Cost : "+cost);
    }
}

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

        System.out.println("Enter the Serial Number : ");
        int serial_no = ob.nextInt();
        System.out.println("Enter the Name : ");
        String name = ob.next();
        System.out.println("Enter the cost : ");
        int cost = ob.nextInt();

        Vehicle v;
        v = new Car(serial_no,name,cost);
        v.display();

    }
}
```

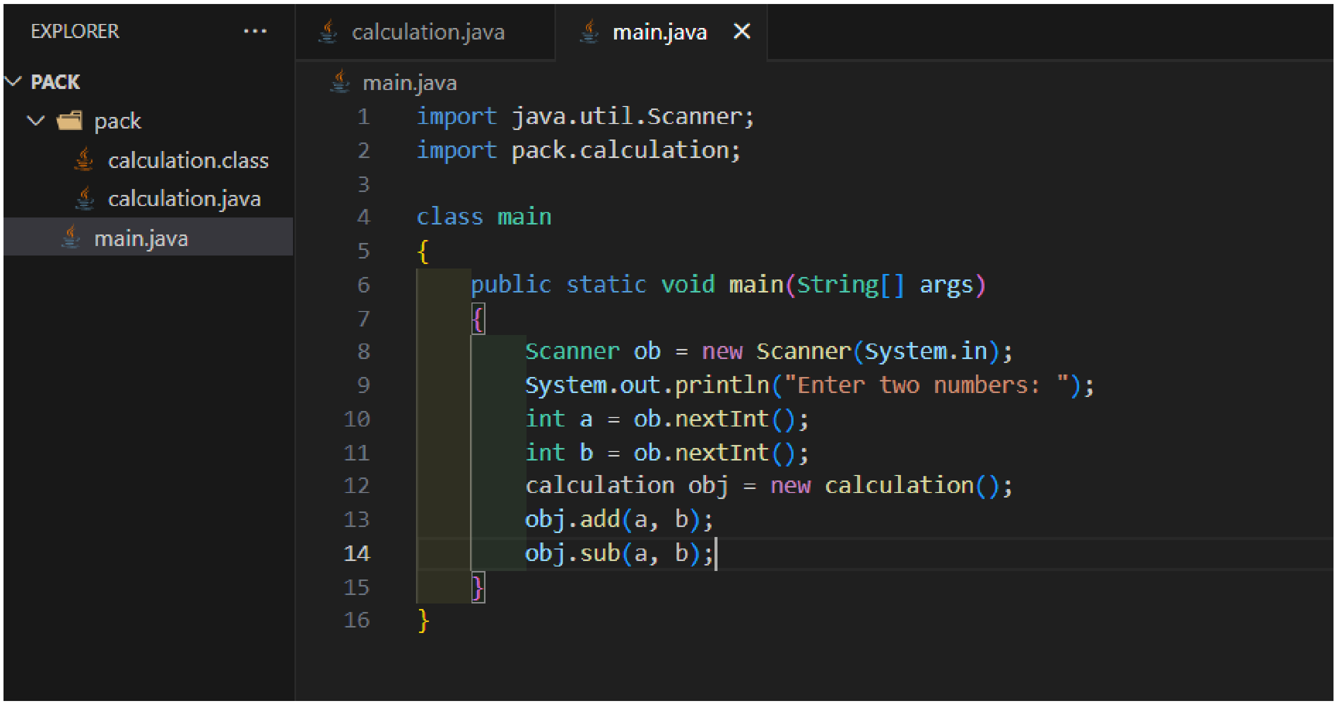
```
/*
Create an interface having prototypes of functions area() and perimeter(). Create
two classes Circle and Rectangle which implements the above interface. Create
a menu driven program to find area and perimeter of objects.
*/
import java.util.Scanner;
interface A_P
{
    void area();
    void perimeter();
}

class Circle implements A_P
{
    int r;

    Circle(int r)
    {
        this.r = r;
    }

    public void area()
    {
        double area = 3.14 * r * r;
        System.out.println("Area of the Circle : "+area);
    }

    public void perimeter()
    {
        double perimeter = 2 * 3.14 * r;
        System.out.println("Perimeter of the Circle : "+perimeter);
    }
}

class Rectangle implements A_P
{
    int l,b;

    Rectangle(int l,int b)
    {
        this.l = l;
        this.b = b;
    }

    public void area()
    {
        int area = l * b;
        System.out.println("Area of the Rectangle : "+area);
    }
    public void perimeter()
    {
        int perimeter = 2*(l+b);
        System.out.println("Perimeter of the Rectangle : "+perimeter);
    }
}

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

        do {
            System.out.println("\n--- Menu ---");
            System.out.println("1. Circle");
            System.out.println("2. Rectangle");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter radius of the circle: ");
                    int r = sc.nextInt();
                    A_P circle = new Circle(r);
                    circle.area();
                    circle.perimeter();
                    break;

                case 2:
                    System.out.print("Enter length and breadth of the rectangle: ");
                    int l = sc.nextInt();
                    int b = sc.nextInt();
                    A_P rectangle = new Rectangle(l, b);
                    rectangle.area();
                    rectangle.perimeter();
                    break;

                case 3:
                    System.out.println("Exiting program.");
                    break;

                default:
                    System.out.println("Invalid choice! Please try again.");
            }
        } while (choice != 3);

        sc.close();
    }
}
```

PACKAGES

Create a package named "Pack" with a class "Calculations". This class has methods for calculating the square, cube and square root of a given number. Import this package into our class and perform all the calculations inside the package

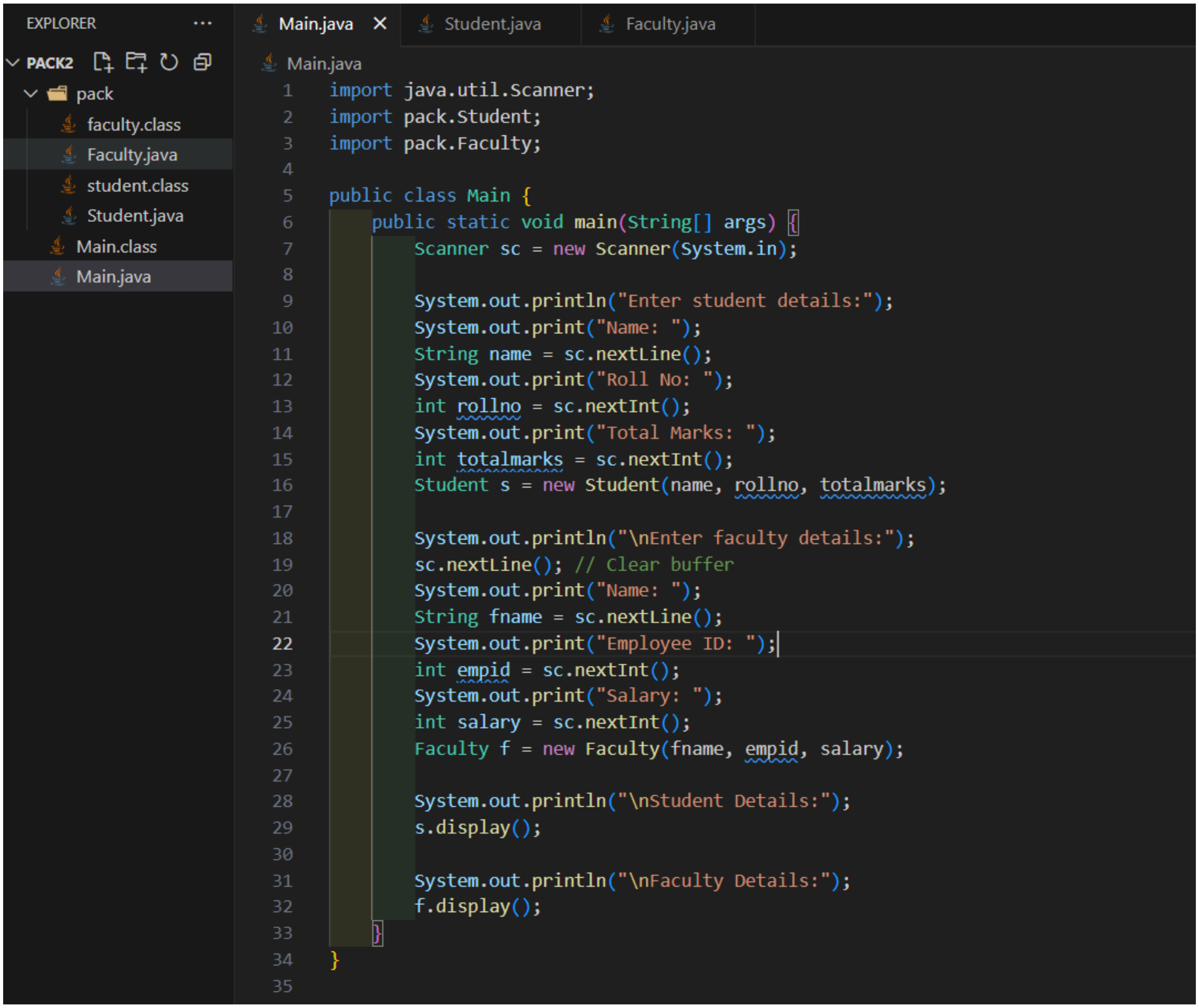


```java
package pack;

public class calculation {

    public void add(int a,int b)
    {
        int c = a + b;
        System.out.println("The sum of two numbers is: " + c);
    }

    public void sub(int a,int b)
    {
        int c = a - b;
        System.out.println("The difference of two numbers is: " + c);
    }
}
```

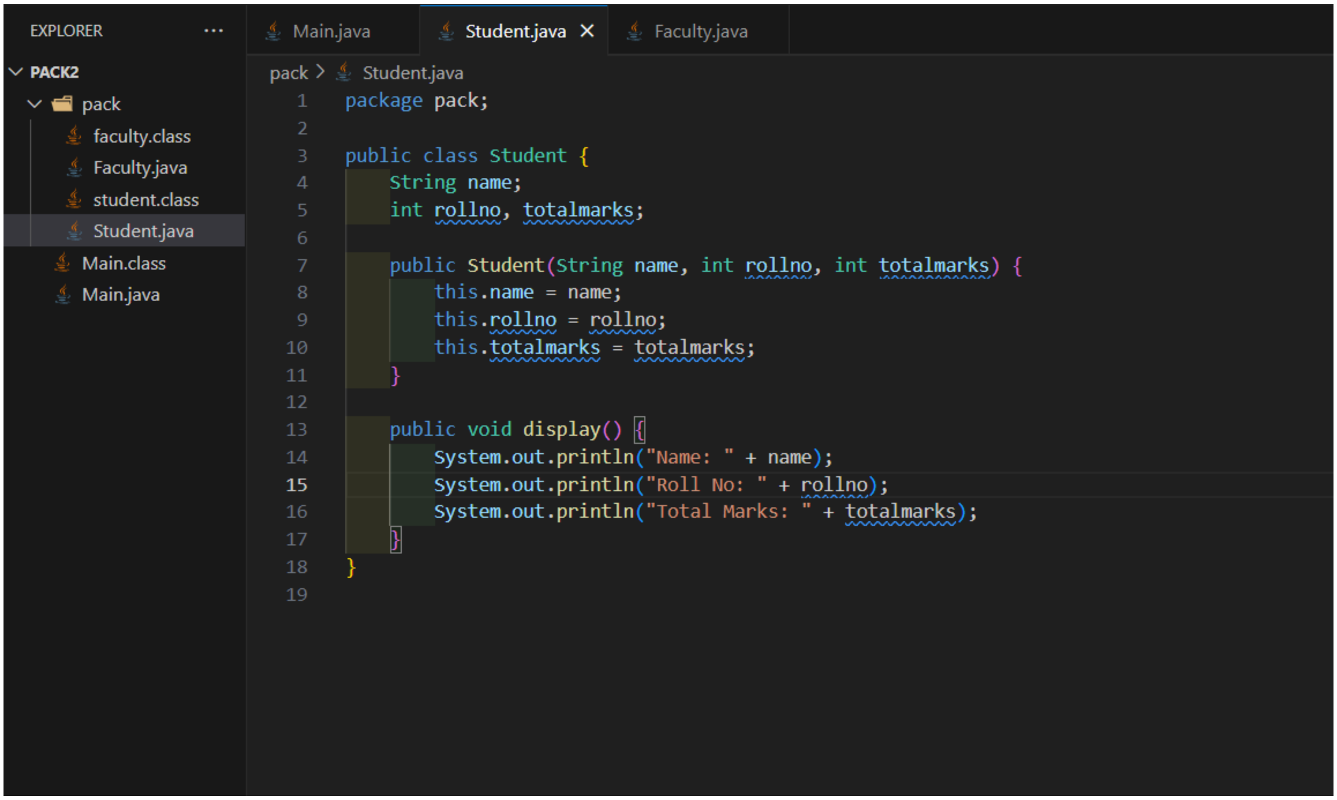


```java
import java.util.Scanner;
import pack.calculation;

class main
{
    public static void main(String[] args)
    {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter two numbers: ");
        int a = ob.nextInt();
        int b = ob.nextInt();
        calculation obj = new calculation();
        obj.add(a, b);
        obj.sub(a, b);
    }
}
```

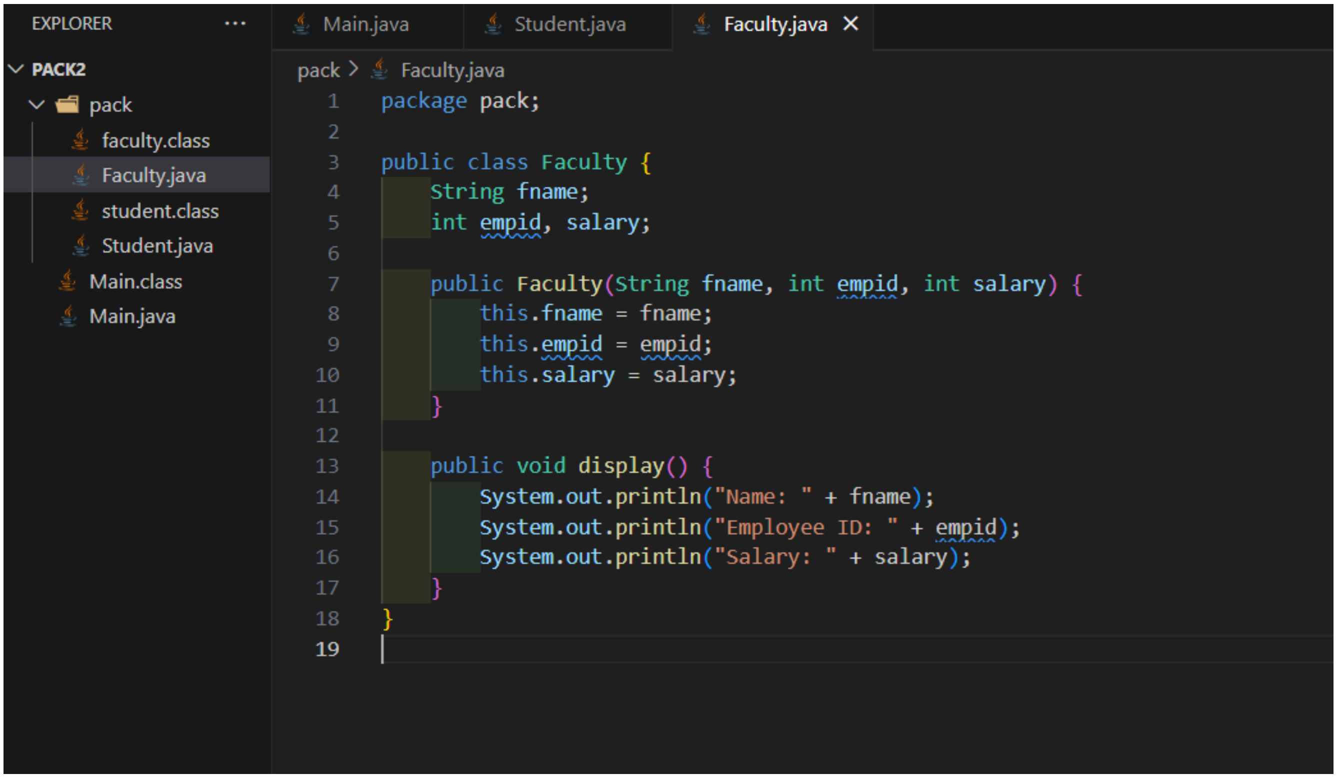
Create a package Pack2 with 2 classes Student and Faculty. Student class is used to accept students' details and display it. Faculty class is used to accept faculty details and display it. Import these two classes into our program and perform all operations



```java
import java.util.Scanner;
import pack.Student;
import pack.Faculty;

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

        System.out.println("Enter student details:");
        System.out.print("Name: ");
        String name = sc.nextLine();
        System.out.print("Roll No: ");
        int rollno = sc.nextInt();
        System.out.print("Total Marks: ");
        int totalmarks = sc.nextInt();
        Student s = new Student(name, rollno, totalmarks);

        System.out.println("\nEnter faculty details:");
        sc.nextLine(); // Clear buffer
        System.out.print("Name: ");
        String fname = sc.nextLine();
        System.out.print("Employee ID: ");
        int empid = sc.nextInt();
        System.out.print("Salary: ");
        int salary = sc.nextInt();
        Faculty f = new Faculty(fname, empid, salary);

        System.out.println("\nStudent Details:");
        s.display();

        System.out.println("\nFaculty Details:");
        f.display();
    }
}

```



```java
package pack;

public class Student {
    String name;
    int rollno, totalmarks;

    public Student(String name, int rollno, int totalmarks) {
        this.name = name;
        this.rollno = rollno;
        this.totalmarks = totalmarks;
    }

    public void display() {
        System.out.println("Name: " + name);
        System.out.println("Roll No: " + rollno);
        System.out.println("Total Marks: " + totalmarks);
    }
}
```



```java
package pack;

public class Faculty {
    String fname;
    int empid, salary;

    public Faculty(String fname, int empid, int salary) {
        this.fname = fname;
        this.empid = empid;
        this.salary = salary;
    }

    public void display() {
        System.out.println("Name: " + fname);
        System.out.println("Employee ID: " + empid);
        System.out.println("Salary: " + salary);
    }
}
```

EXCEPTION HANDLING

```
import java.util.Scanner;

public class MultipleExceptionsExample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] numbers = {10, 20, 30, 40, 50};

        try {
            // Arithmetic Exception block
            System.out.print("Enter numerator: ");
            int num = sc.nextInt();

            System.out.print("Enter denominator: ");
            int denom = sc.nextInt();

            int result = num / denom;
            System.out.println("Division result: " + result);

            // ArrayIndexOutOfBoundsException block
            System.out.print("Enter array index to access: ");
            int index = sc.nextInt();

            System.out.println("Element at index " + index + " is: " + numbers[index]);

        } catch (ArithmeticException e) {
            System.out.println("Error: Cannot divide by zero.");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: Invalid array index.");
        } catch (Exception e) {
            System.out.println("Some other exception occurred: " + e.getMessage());
        }

        System.out.println("Program execution completed.");
    }
}
```

USER DEFINED EXCEPTION HANDLER

```
import java.util.Scanner;

// User-defined Exception
class InvalidAgeException extends Exception {
    InvalidAgeException(String message) {
        super(message);
    }
}

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

        System.out.print("Enter your age: ");
        int age = sc.nextInt();

        try {
            if (age < 18) {
                // Throwing the user-defined exception
                throw new InvalidAgeException("You are not eligible to vote. Age must be 18 or above.");
            } else {
                System.out.println("You are eligible to vote.");
            }
        } catch (InvalidAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

USER AUTHENTICATION EXCEPTION HADNLING

```
import java.util.*;

class AuthenticationException extends Exception
{
    public AuthenticationException(String message){
        super(message);
    }
}

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

        try{
            System.out.println("Enter the username : ");
            String uname = ob.next();
            System.out.println("Enter the Password : ");
            String passwd = ob.next();

            if(!uname.equals("admin") || !passwd.equals("1234"))
            {
                throw new AuthenticationException("Failed to login");
            }
            else
            {
                System.out.println("Done");
            }
        }catch (AuthenticationException e)
        {
            System.out.println("Error "+e.getMessage());
        }
    }
}
```

EXAMPLE OF USING ABSTRACT CLASS AND DYNAMIC METHOD DISPATCH

```
abstract class Shape {
    abstract void draw();  // Abstract method
}

class Circle extends Shape {
    @Override
    void draw() {
        System.out.println("Drawing a Circle");
    }
}

class Square extends Shape {
    @Override
    void draw() {
        System.out.println("Drawing a Square");
    }
}

class Triangle extends Shape {
    @Override
    void draw() {
        System.out.println("Drawing a Triangle");
    }
}

public class Main {
    public static void main(String[] args) {
        Shape s;

        s = new Circle();    // s refers to Circle
        s.draw();            // Output: Drawing a Circle

        s = new Square();    // s refers to Square
        s.draw();            // Output: Drawing a Square

        s = new Triangle();  // s refers to Triangle
        s.draw();            // Output: Drawing a Triangle
    }
}
```

INNER CLASS AND STATIC CLASS

```
class CPU {
    int price;

    // Constructor for CPU
    CPU(int price) {
        this.price = price;
    }

    // Non-static inner class
    class Processor {
        int noOfCores;
        String manufacturer;

        Processor(int noOfCores, String manufacturer) {
            this.noOfCores = noOfCores;
            this.manufacturer = manufacturer;
        }

        void display() {
            System.out.println("Processor Information:");
            System.out.println("Cores: " + noOfCores);
            System.out.println("Manufacturer: " + manufacturer);
        }
    }

    // Static nested class
    static class RAM {
        int memory;
        String manufacturer;

        RAM(int memory, String manufacturer) {
            this.memory = memory;
            this.manufacturer = manufacturer;
        }

        void display() {
            System.out.println("RAM Information:");
            System.out.println("Memory: " + memory + " GB");
            System.out.println("Manufacturer: " + manufacturer);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        // Create outer class object
        CPU cpu = new CPU(30000);

        // Create inner class Processor object using CPU instance
        CPU.Processor processor = cpu.new Processor(6, "Intel");

        // Create static nested class RAM object
        CPU.RAM ram = new CPU.RAM(16, "Corsair");

        System.out.println("CPU Price: ₹" + cpu.price);
        processor.display();
        ram.display();
    }
}
```

RECURSION

FACTORAIL

```
import java.util.Scanner;

public class Main {

    // Static method to calculate factorial recursively
    static int factorial(int n) {
        if (n == 0 || n == 1)
            return 1; // Base case
        else
            return n * factorial(n - 1); // Recursive call
    }

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

        System.out.print("Enter a number to find its factorial: ");
        int num = sc.nextInt();
        if (num < 0) {
            System.out.println("Factorial is not defined for negative numbers.");
        } else {
            int result = factorial(num);
            System.out.println("Factorial of " + num + " is: " + result);
        }
    }
}
```

FIBONACCI

```
public class Main {
    // static recursive method to find nth Fibonacci number
    static int fib(int n) {
        if (n == 0)  // base case 1
            return 0;
        else if (n == 1)  // base case 2
            return 1;
        else
            return fib(n - 1) + fib(n - 2);  // recursive call
    }

    public static void main(String[] args) {
        int n = 10; // number of Fibonacci terms to print
        System.out.println("Fibonacci Series up to " + n + " terms:");

        for (int i = 0; i < n; i++) {
            System.out.print(fib(i) + " ");
        }
    }
}
```