**Lab3**

**Prac\_1:- Create a superclass Person with attributes name and age, and a method display(). Create a subclass Student that adds an attribute studentID. Write a program to create a Student object and display all its attributes.**

**Program:**

package prc22;

// Main class to test the Person and Student classes

public class PersonStudent {

public static void main(String[] args) {

// Create a student object with ID 2, name "riya", and age 12

student st = new student(2, "riya", 12);

// Call the display method to print the student details

st.display();

}

}

// Base class representing a person

class person {

private int age;

private String name;

// Constructor to initialize name and age

public person(String name, int age) {

this.name = name;

this.age = age;

}

// Method to display the name and age of the person

public void display() {

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

System.out.println("Age: " + age);

}

}

// Derived class representing a student, which is a type of person

class student extends person {

private int stdID;

// Constructor to initialize student ID, name, and age

// Calls the parent class constructor to initialize name and age

public student(int stdID, String name, int age) {

super(name, age);

this.stdID = stdID;

}

// Method to display the student details

// Calls the parent class display method to show name and age

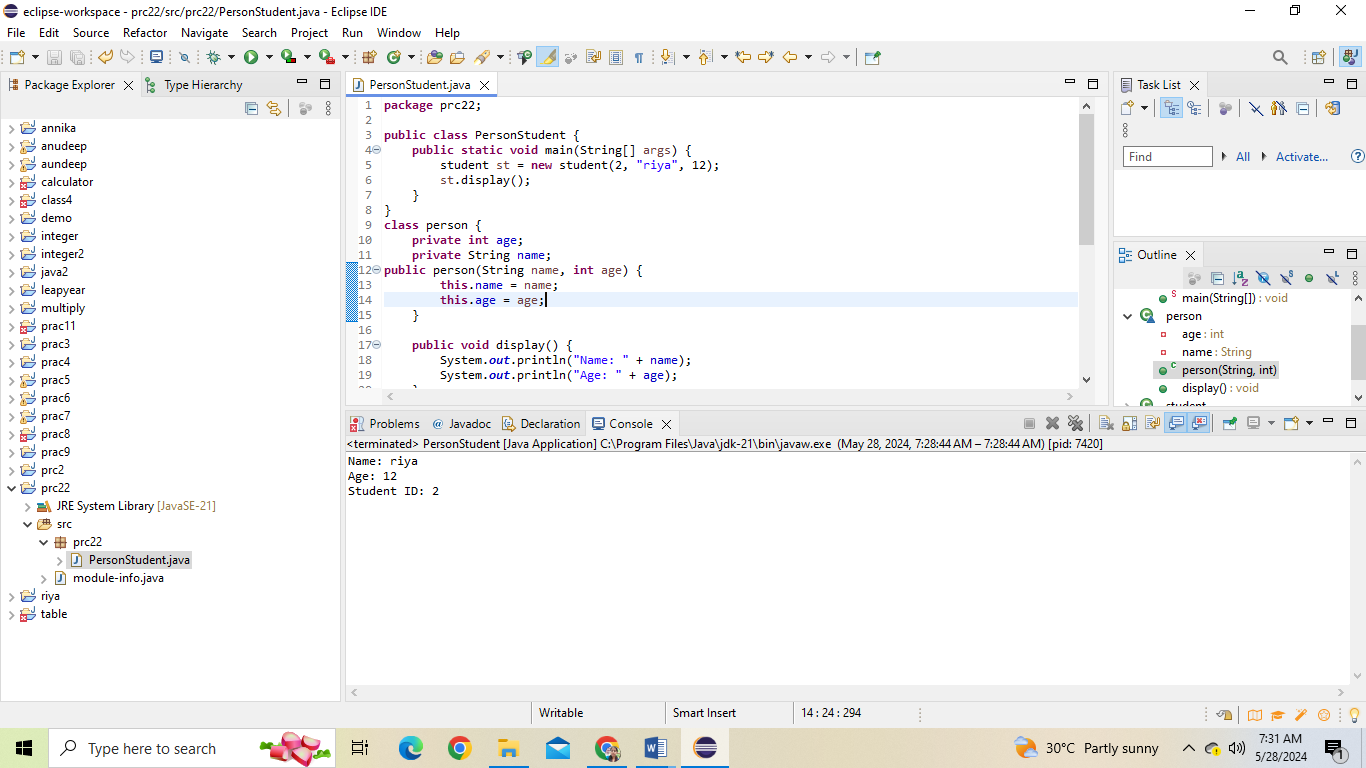
public void display() {

super.display();

System.out.println("Student ID: " + stdID);

}

}

Output: 

**Prac\_2:- Create a superclass Calculator with a method add(int a, int b). Create a subclass AdvancedCalculator that overloads the add method to handle three integers.**

Program: package prac23;

// Main class to test the Calculator and ACalculator classes

public class OverLoadCal\_M {

public static void main(String[] args) {

// Create an instance of ACalculator

ACalculator AC = new ACalculator();

// Call the add method with three arguments and print the result

System.out.println(AC.add(2, 3, 5));

}

}

// Base class representing a simple calculator

class Calculator {

int a;

int b;

// Method to add two integers and return the result

public int add(int a, int b) {

return a + b;

}

}

// Derived class extending the Calculator class

class ACalculator extends Calculator {

int c;

// Overloaded add method to add three integers

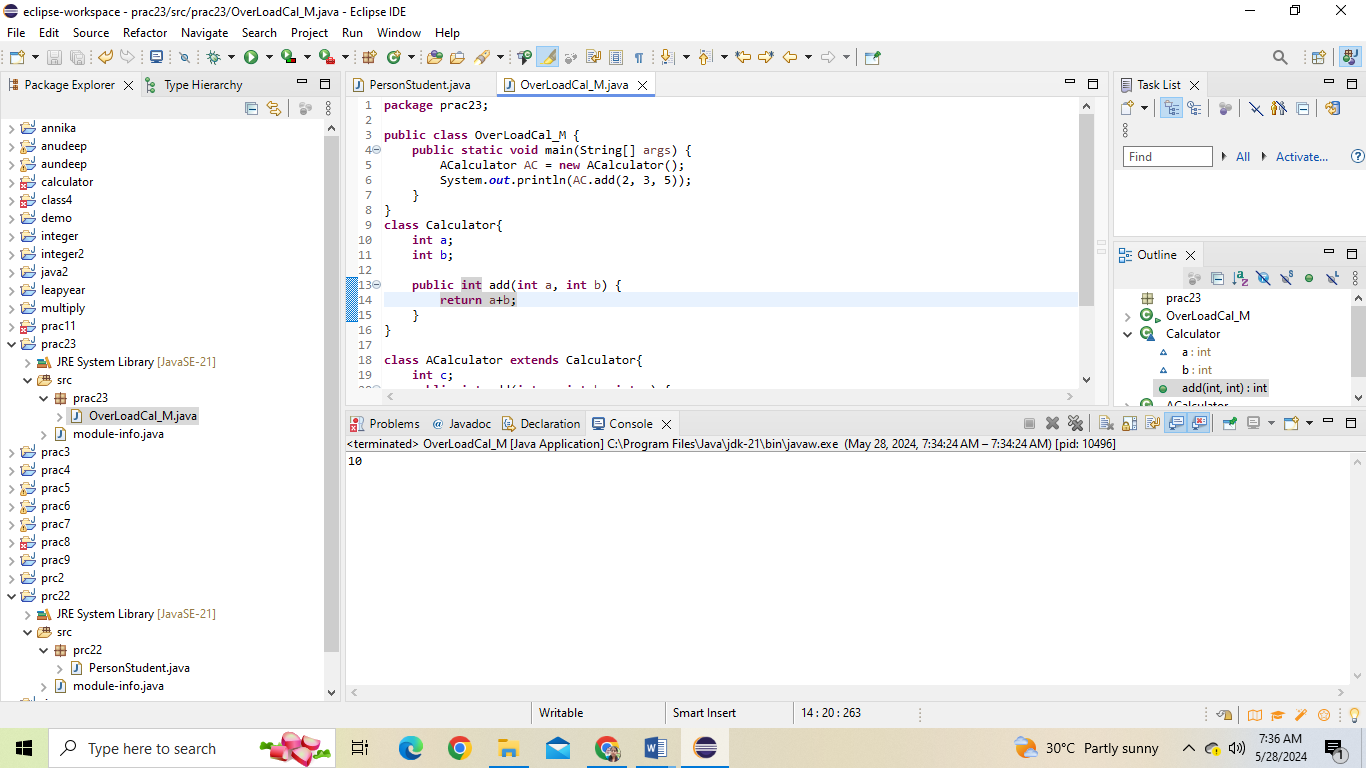
// Calls the parent class add method for the first two integers

public int add(int a, int b, int c) {

return super.add(a, b) + c;

}

}

Output: 

**Prac\_3:- Create a superclass Vehicle with a method move(). Create subclasses Car and Bike that inherit from Vehicle. Write a program to create objects of Car and Bike and call the move() method on each.**

Program: package prac24;

// Main class to test the Car and Bike classes

public class HirericalCar {

public static void main(String[] args) {

// Create an instance of Car

Car c = new Car();

// Call the move method on the Car instance

c.move();

System.out.println();

// Create an instance of Bike

Bike b = new Bike();

// Call the move method on the Bike instance

b.move();

}

}

// Base class representing a generic vehicle

class Vehicle {

// Method to indicate the vehicle is moving

public void move() {

System.out.println("A vehicle is moving");

}

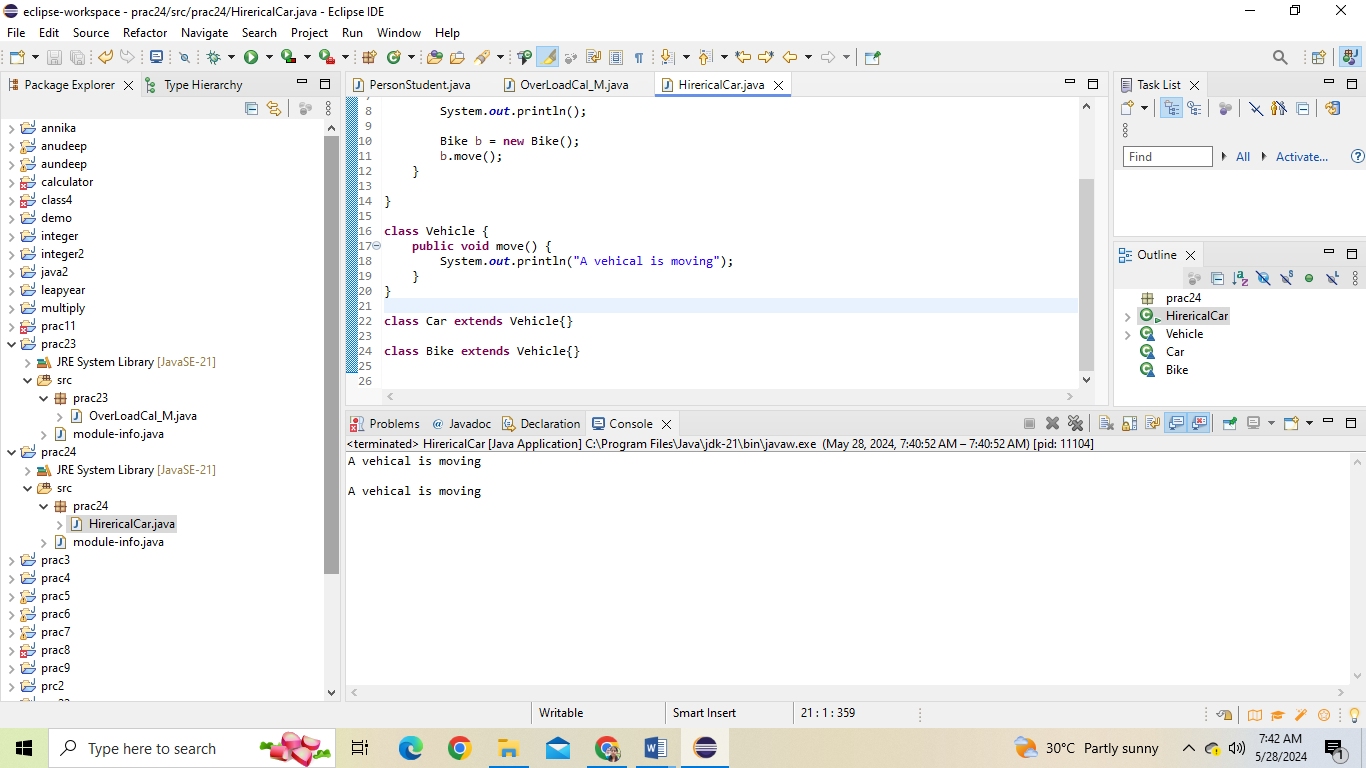
}

// Derived class representing a car, inheriting from Vehicle

class Car extends Vehicle {}

// Derived class representing a bike, inheriting from Vehicle

class Bike extends Vehicle {}

output 

**Prac\_4:- :- Create an class Document with an method void open(). Implement subclasses WordDocument, PDFDocument, and SpreadsheetDocument that extend Document and provide implementations for open(). Write a main class to demonstrate opening different types of documents.(implement complile time- polymorphism).**

**Program: package** prac24;

**public** **class** DocumentDemo {

// Abstract Document class

**abstract** **static** **class** Document {

**public** **abstract** **void** open();

}

// WordDocument class extending Document

**static** **class** WordDocument **extends** Document {

@Override

**public** **void** open() {

System.***out***.println("Opening a Word document.");

}

}

// PDFDocument class extending Document

**static** **class** PDFDocument **extends** Document {

@Override

**public** **void** open() {

System.***out***.println("Opening a PDF document.");

}

}

// SpreadsheetDocument class extending Document

**static** **class** SpreadsheetDocument **extends** Document {

@Override

**public** **void** open() {

System.***out***.println("Opening a Spreadsheet document.");

}

}

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

// Create instances of different document types

Document wordDoc = **new** WordDocument();

Document pdfDoc = **new** PDFDocument();

Document spreadsheetDoc = **new** SpreadsheetDocument();

// Open each document

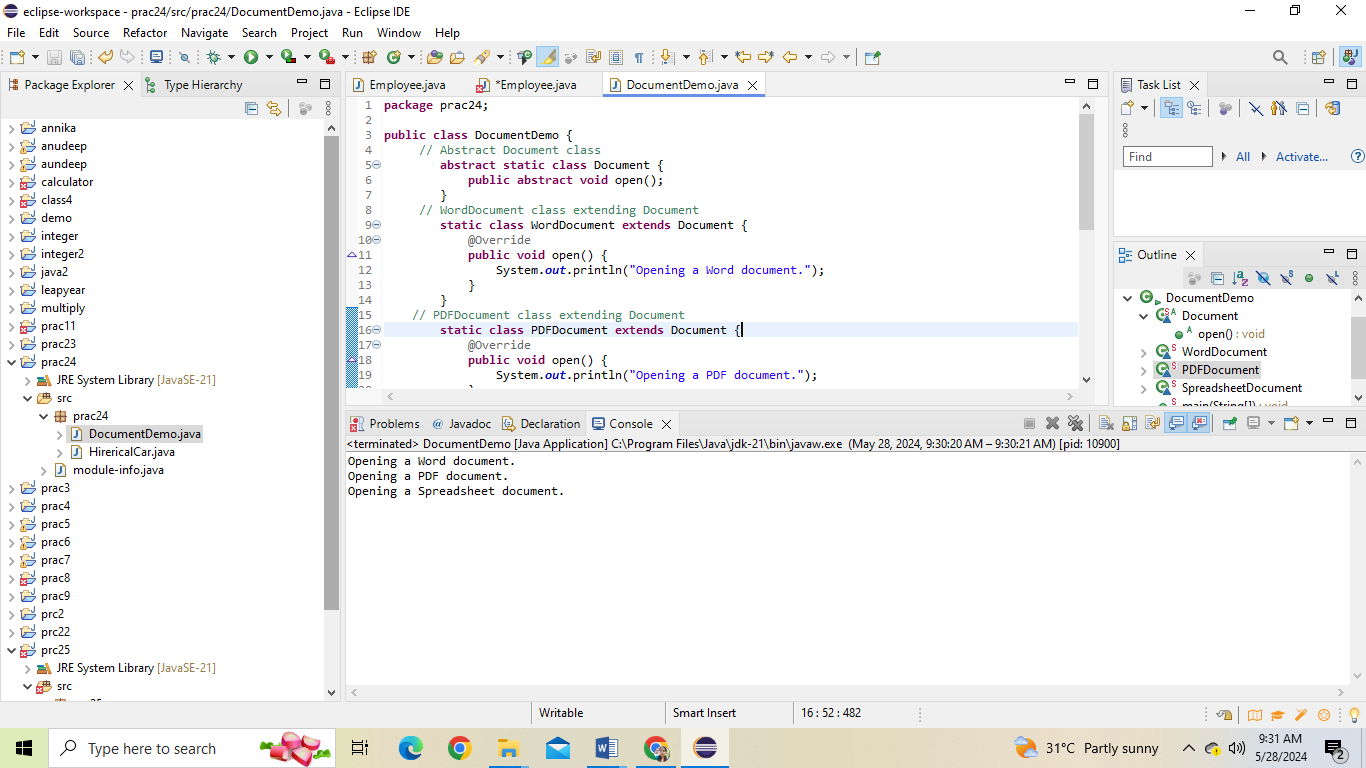
wordDoc.open();

pdfDoc.open();

spreadsheetDoc.open();

}

}

**Output:** 

**Prac\_5:- Create an class Employee with an abstract method calculatePay(). Create subclasses SalariedEmployee and HourlyEmployee that implement the calculatePay() method. Write a program to create objects of both subclasses and call the calculatePay() method.**

Program: **package** prc25;

**abstract** **class** Employe {

String name;

**int** employeeId;

**public** Employe(String name, **int** employeId) {

**this**.name = name;

**this**.employeeId = employeId;

}

// Abstract method to calculate pay

**abstract** **double** calculatePay();

**public** **void** display() {

System.***out***.println("Employee Name is: " + name);

System.***out***.println("Employee Id is: " + employeeId);

System.***out***.println("Employee pay is: " + calculatePay());

}

}

**class** SalariedEmployee **extends** Employe {

**double** annualSalary;

**public** SalariedEmployee(String name, **int** employeeId, **double** annualSalary) {

**super**(name, employeeId);

**this**.annualSalary = annualSalary;

}

@Override

**double** calculatePay() {

**return** annualSalary / 12;

}

}

**class** HourlyEmployee **extends** Employe {

**double** hourlyRate;

**int** hoursWorked;

**public** HourlyEmployee(String name, **int** employeeId, **double** hourlyRate, **int** hoursWorked) {

**super**(name, employeeId);

**this**.hourlyRate = hourlyRate;

**this**.hoursWorked = hoursWorked;

}

@Override

**double** calculatePay() {

**return** hourlyRate \* hoursWorked;

}

}

**public** **class** Employee{

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

// Creating a salaried employee

SalariedEmployee salariedEmployee = **new** SalariedEmployee("sejal", 123, 60000);

salariedEmployee.display();

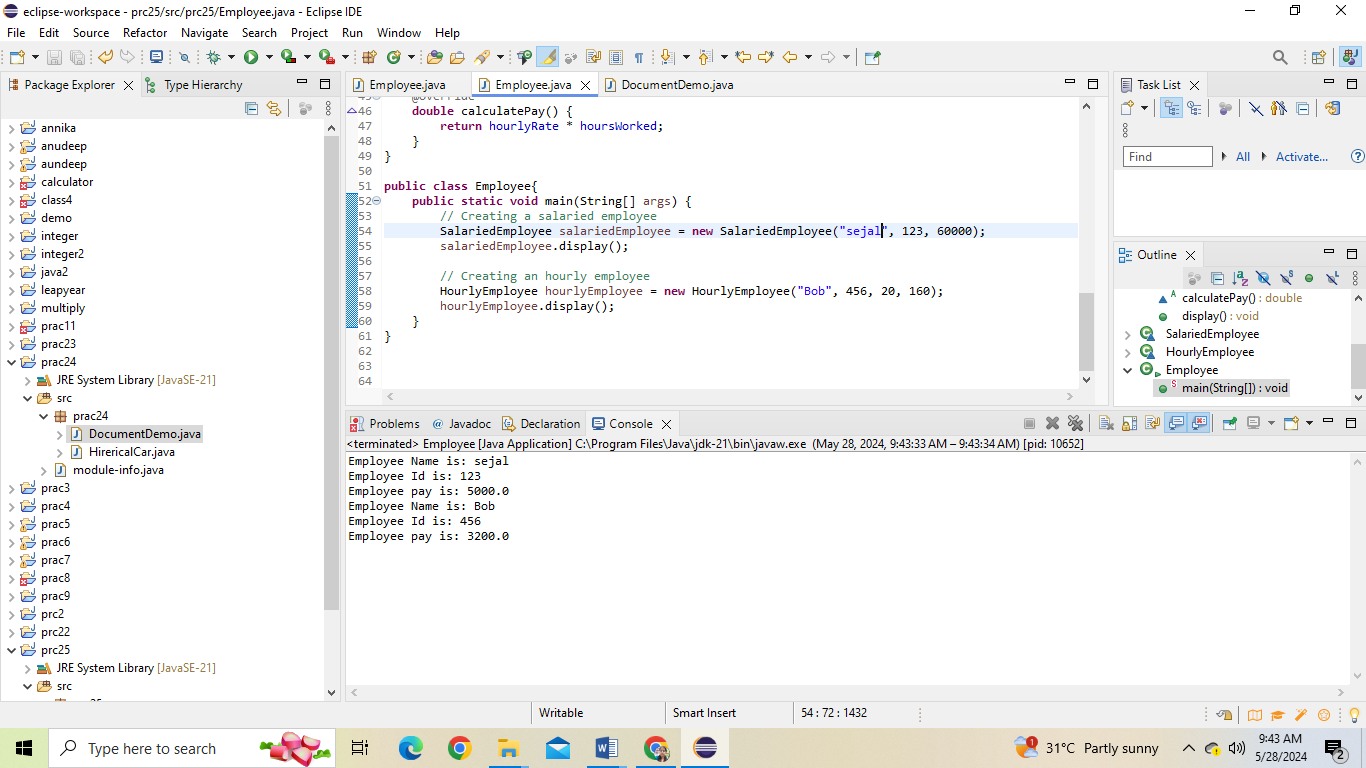
// Creating an hourly employee

HourlyEmployee hourlyEmployee = **new** HourlyEmployee("Bob", 456, 20, 160);

hourlyEmployee.display();

}

}

Output: 

**Prac\_6:- Create a class Calculator with overloaded methods add() that take different numbers and types of parameters: int add(int a, int b) double add(double a, double b) int add(int a, int b, int c) Write a main class to demonstrate the usage of these methods**

**Program: package** prac23;

**public** **class** Overload\_Cal {

// Method to add two integers

**public** **int** add(**int** a, **int** b) {

**return** a + b;

}

// Method to add two doubles

**public** **double** add(**double** a, **double** b) {

**return** a + b;

}

// Method to add three integers

**public** **int** add(**int** a, **int** b, **int** c) {

**return** a + b + c;

}

// Method to add four doubles

**public** **double** add(**double** a, **double** b, **double** c, **double** d) {

**return** a + b + c + d;

}

// Method to add an array of integers

**public** **int** add(**int**[] numbers) {

**int** sum = 0;

**for** (**int** num : numbers) {

sum += num;

}

**return** sum;

}

// Method to add an array of doubles

**public** **double** add(**double**[] numbers) {

**double** sum = 0.0;

**for** (**double** num : numbers) {

sum += num;

}

**return** sum;

}

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

Overload\_Cal calc = **new** Overload\_Cal();

// Adding two integers

**int** sumInt = calc.add(5, 10);

System.***out***.println("Sum of integers: " + sumInt);

// Adding two doubles

**double** sumDouble = calc.add(3.5, 2.5);

System.***out***.println("Sum of doubles: " + sumDouble);

// Adding three integers

**int** sumThreeInt = calc.add(2, 4, 6);

System.***out***.println("Sum of three integers: " + sumThreeInt);

// Adding four doubles

**double** sumFourDouble = calc.add(1.5, 2.5, 3.5, 4.5);

System.***out***.println("Sum of four doubles: " + sumFourDouble);

// Adding an array of integers

**int**[] intArray = {1, 2, 3, 4, 5};

**int** sumIntArray = calc.add(intArray);

System.***out***.println("Sum of array of integers: " + sumIntArray);

// Adding an array of doubles

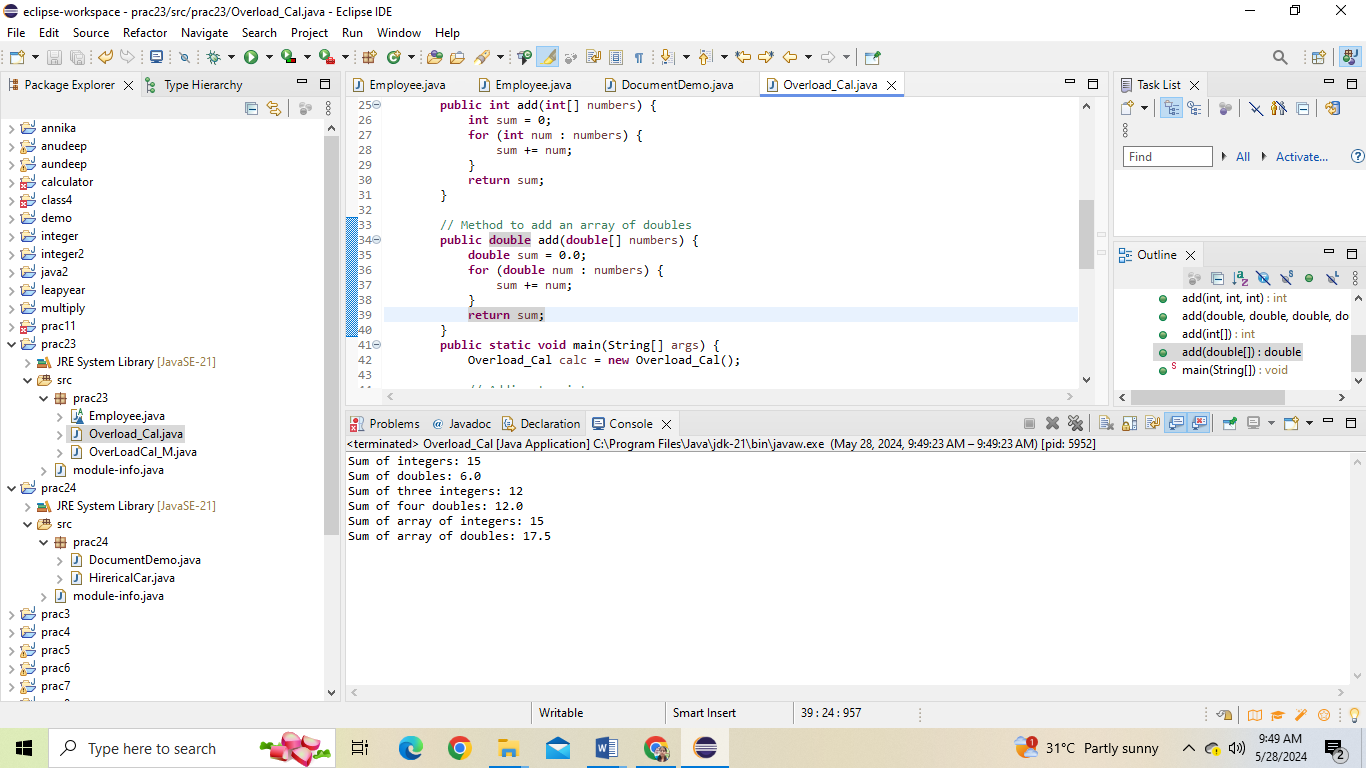
**double**[] doubleArray = {1.5, 2.5, 3.5, 4.5, 5.5};

**double** sumDoubleArray = calc.add(doubleArray);

System.***out***.println("Sum of array of doubles: " + sumDoubleArray);

}

}

Output: 

**Prac\_7:- Create a** [**JavaBean**](https://aln.anudip.org/mod/resource/view.php?id=12692) **class Person with properties firstName, lastName, age, and email. Implement the required no-argument constructor, getter and setter methods for each property. Write a main class to create an instance of Person, set its properties, and print them out**

**Program:** package java2;

public class Main\_person {

// Creating an instance of the inner class Person

private Person p = new Person();

// Inner class Person

public class Person {

// Private member variables

private String firstName;

private String lastName;

private int age;

private String email;

// Getter and Setter methods for firstName

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

// Getter and Setter methods for lastName

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

// Getter and Setter methods for age

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

// Getter and Setter methods for email

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

}

public static void main(String[] args) {

// Creating an instance of Main\_person to access the inner class instance

Main\_person MP = new Main\_person();

// Setting values for the person instance

MP.p.setFirstName("Riya");

MP.p.setLastName("Yadav");

MP.p.setAge(21);

MP.p.setEmail("Yriya@gmail.com");

// Getting and printing values from the person instance

System.out.println(MP.p.getFirstName());

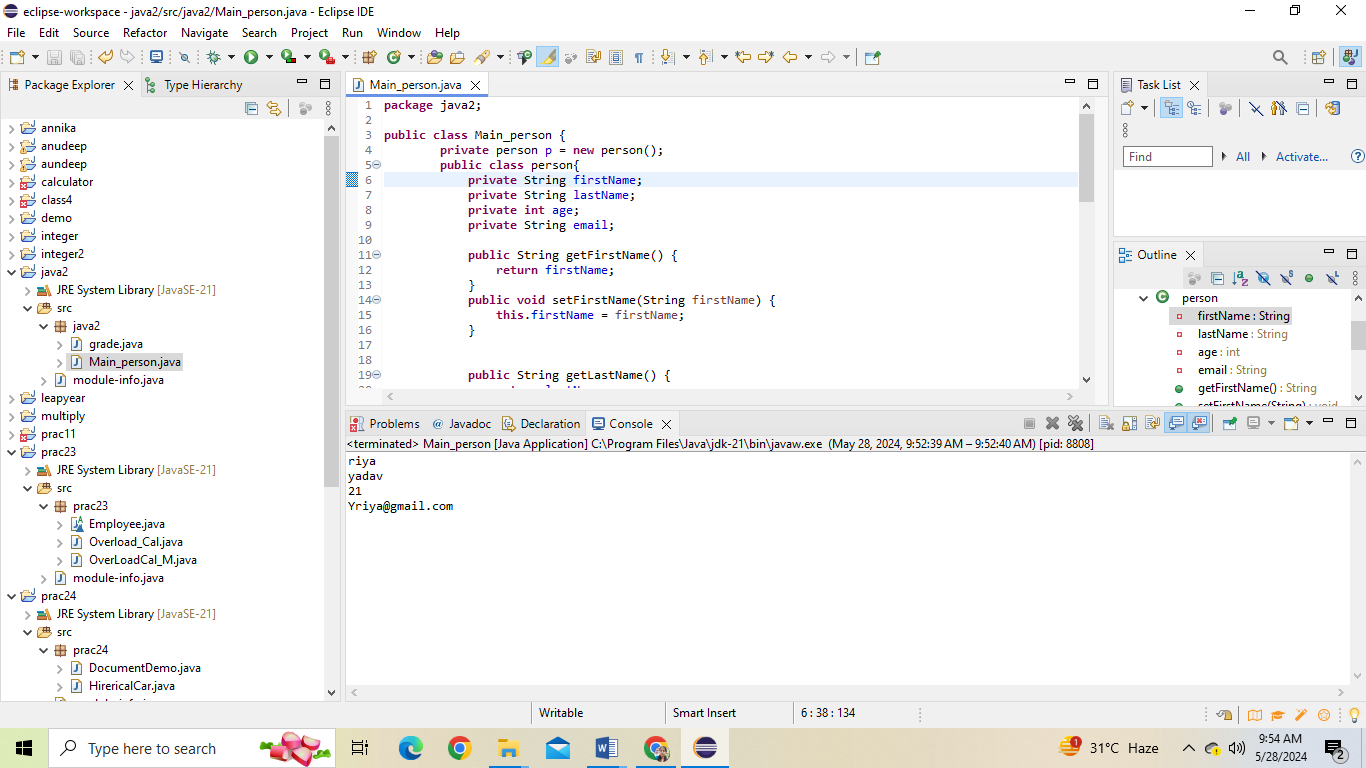
System.out.println(MP.p.getLastName());

System.out.println(MP.p.getAge());

System.out.println(MP.p.getEmail());

}

}

Output: 

**Prac\_8:- Create a** [**JavaBean**](https://aln.anudip.org/mod/resource/view.php?id=12692) **class Car with properties make, model, year, and color. Implement the required no-argument constructor, getter and setter methods for each property. Write a main class to create an instance of Car, set its properties, and print the car details**

**Program: package** anudeep;

**class** cars {

**private** String make;

**private** String model;

**private** **int** year;

**private** String color;

// Getter and setter methods for make

**public** String getMake() {

**return** make;

}

**public** **void** setMake(String make) {

**this**.make = make;

}

// Getter and setter methods for model

**public** String getModel() {

**return** model;

}

**public** **void** setModel(String model) {

**this**.model = model;

}

// Getter and setter methods for year

**public** **int** getYear() {

**return** year;

}

**public** **void** setYear(**int** year) {

**this**.year = year;

}

// Getter and setter methods for color

**public** String getColor() {

**return** color;

}

**public** **void** setColor(String color) {

**this**.color = color;

}

}

**public** **class** Main\_car {

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

cars myCar = **new** cars();

// Set car properties

myCar.setMake("Toyota");

myCar.setModel("Camry");

myCar.setYear(2022);

myCar.setColor("Red");

// Print car details

System.***out***.println("Car Details:");

System.***out***.println("Make: " + myCar.getMake());

System.***out***.println("Model: " + myCar.getModel());

System.***out***.println("Year: " + myCar.getYear());

System.***out***.println("Color: " + myCar.getColor());

}

}

Output: 