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.**

package day10;

class Person {

protected String name;

protected int age;

public void Main(String name, int age) {

this.name = name; // to call the current object

this. age = age;

}

public void display() {

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

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

}

}

class Students extends Person {

@Override

public void Main(String name, int age) {

// TODO Auto-generated method stub

super. Main(name, age);

}

@Override

public void display() {

// TODO Auto-generated method stub

super. Display();

}

public Students(String studentID) {

}

public Students(String string, int i, String string2) {

}

}

public class Main {

public static void main(String[] args) {

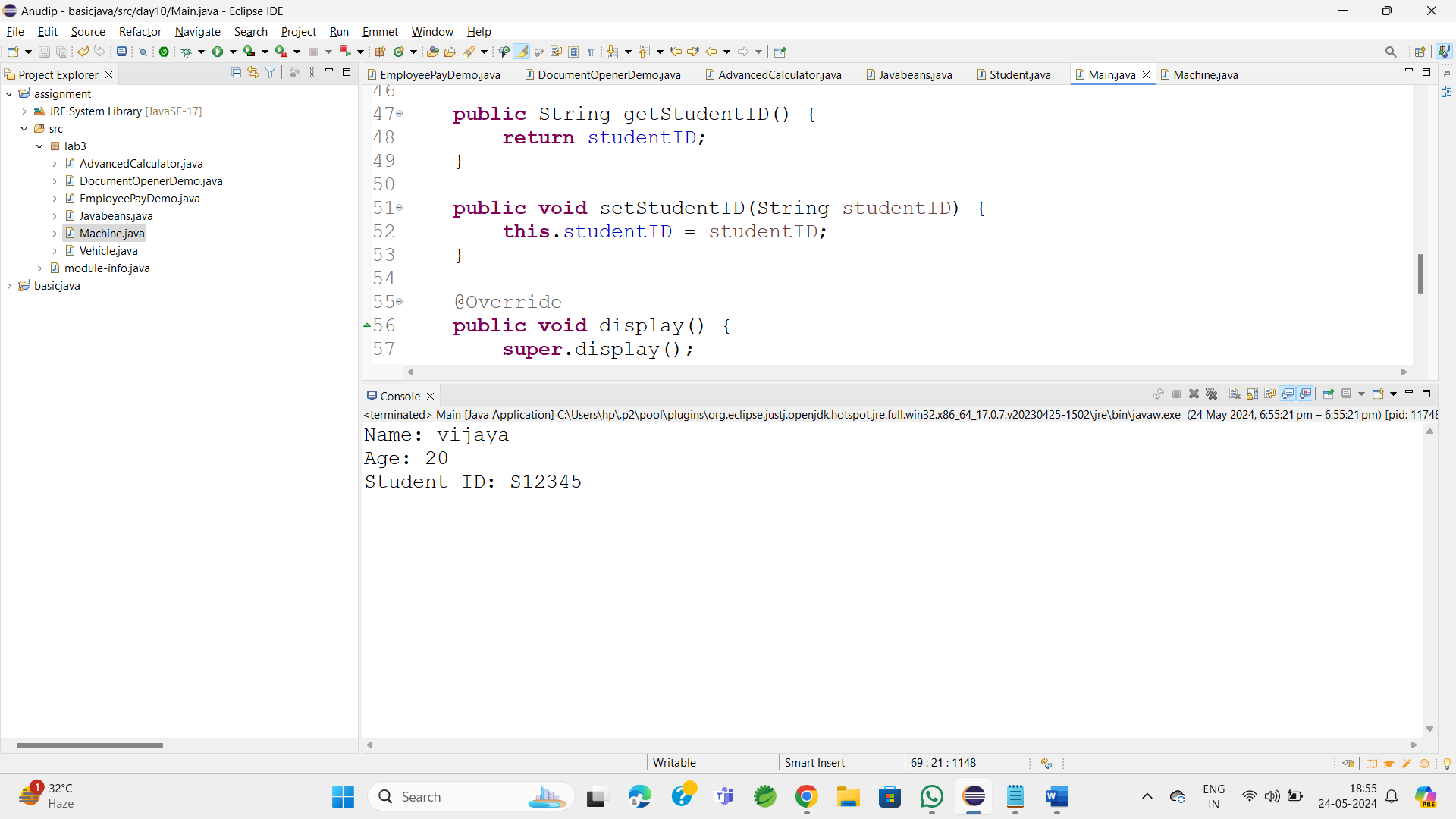
// Create a Student object

Students student = new Students("hhhhh", 21, "s123456");

student. display();

}}

**output:**

****

1. **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.**

package lab3;

class Calculator {

public int add(int a, int b) {

return a + b;

}

}

public class AdvancedCalculator extends Calculator {

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

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

}

public static void main(String[] args) {

AdvancedCalculator calc = new AdvancedCalculator();

int result1 = calc.add(5, 10); // Using the superclass method

int result2 = calc.add(5, 10, 15); // Using the subclass method

// Output the results

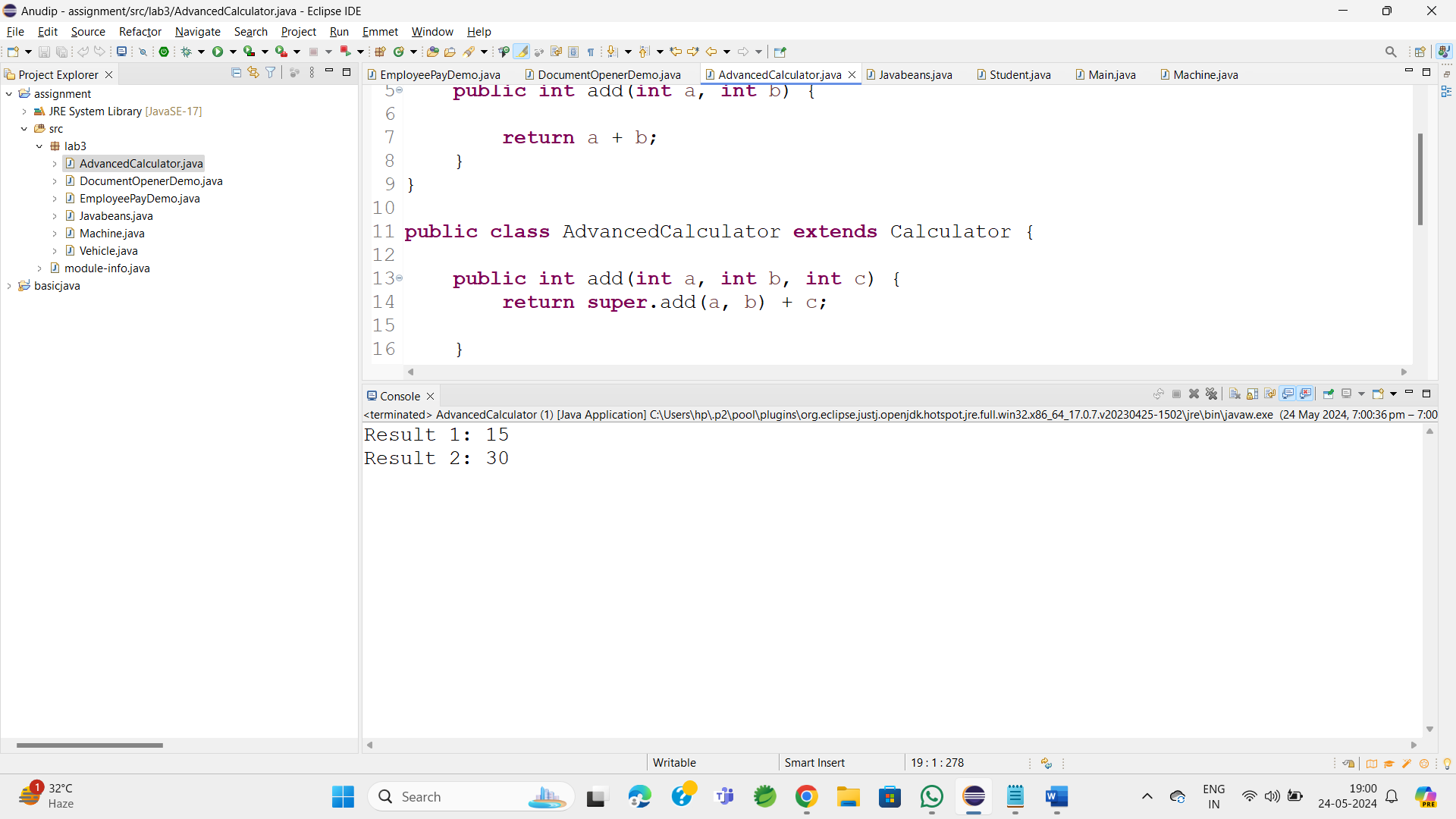
System.out.println("Result 1: " + result1);

System.out.println("Result 2: " + result2);

}

}

**Output:**



1. **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**.

package lab3;

class Car extends Vehicle {

@Override

public void move() {

super.move();

System.out.println("car is moving");

}

}

class Bike extends Vehicle {

@Override

public void move() {

System.out.println("Bike is moving");

}

}

public class Vehicle {

public void move() {

System.out.println("Vehicle is moving");

}

public static void main(String[] args) {

Vehicle myCar = new Car();

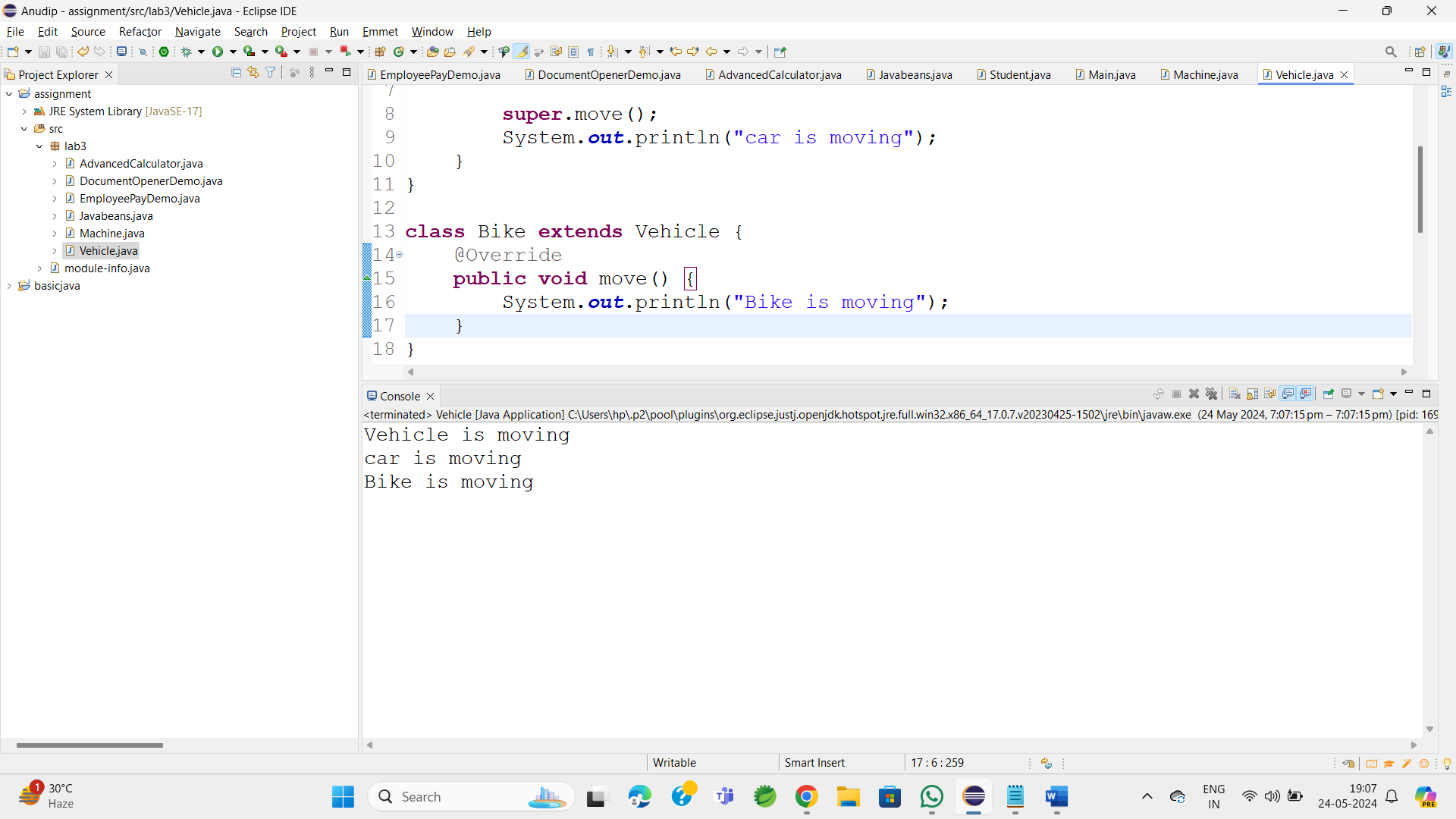
Vehicle myBike = new Bike();

myCar.move();

myBike.move();

}}

**OutPut:**



1. **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.**

package lab3;

//Abstract class Employee

abstract class Employee {

String name;

int employeeId;

// Constructor

public Employee(String name, int employeeId) {

super();

this.name = name;

this.employeeId = employeeId;

}

abstract double calculatePay();

}

class SalaryEmployee extends Employee {

double annualSalary;

// Constructor

public SalaryEmployee(String name, int employeeId, double annualSalary) {

super(name, employeeId);

this.annualSalary = annualSalary;

}

// Implementing the calculatePay method

@Override

double calculatePay() {

return annualSalary / 12;

}

}

class HourlyEmployee extends Employee {

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 EmployeePayDemo {

public static void main(String[] args) {

// Creating an object of SalariedEmployee

Employee salariedEmployee = new SalaryEmployee("vijaya", 10, 60000);

System.out.println("Monthly Pay for Salaried Employee: " + salariedEmployee.calculatePay());

// Creating an object of HourlyEmployee

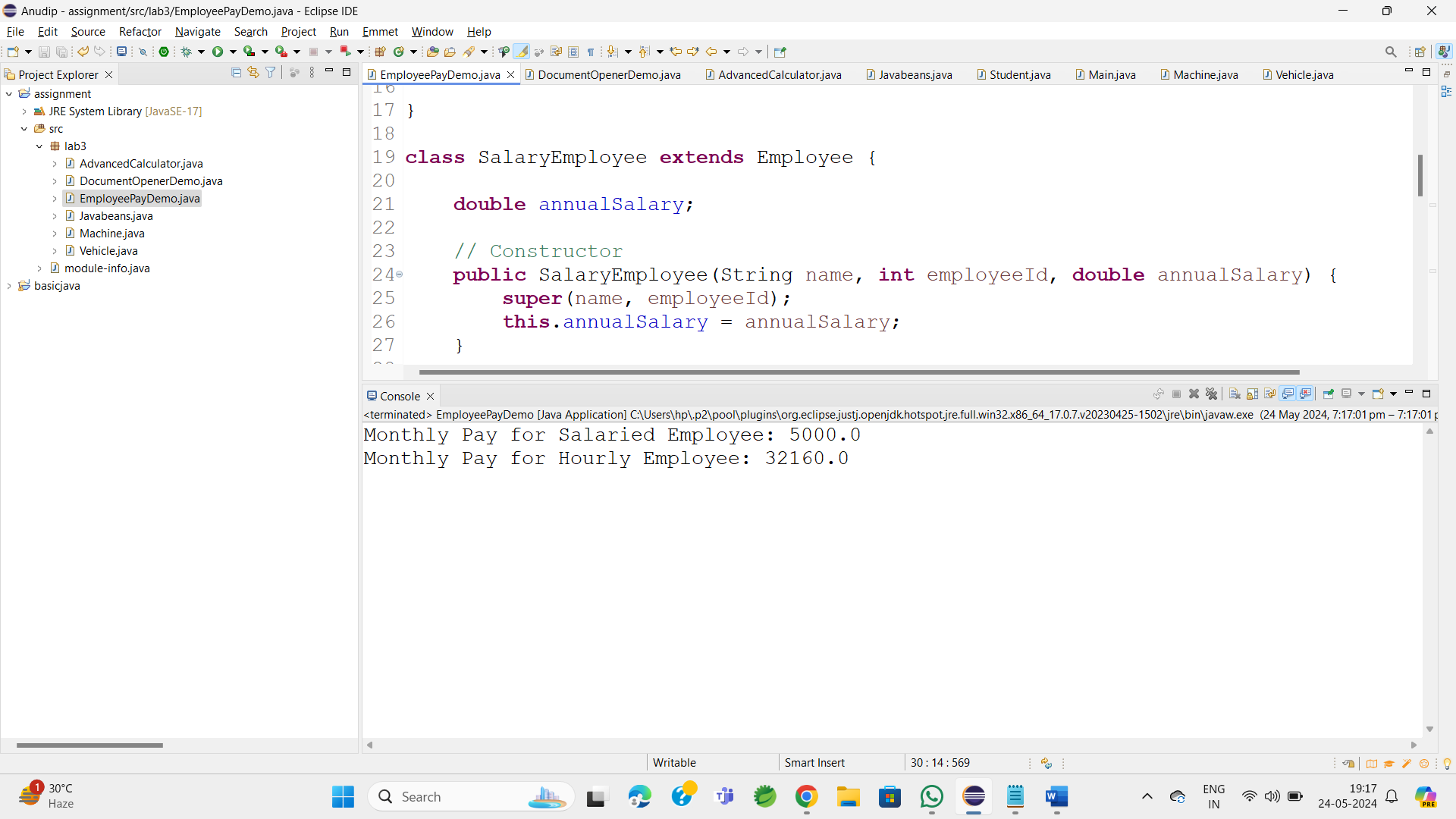
Employee hourlyEmployee = new HourlyEmployee("soni", 102, 201, 160);

System.out.println("Monthly Pay for Hourly Employee: " + hourlyEmployee.calculatePay());

}

}

**OutPut:**

****

1. **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).**

package lab3;

abstract class Document {

String fileName;

// Constructor

Document(String fileName) {

this.fileName = fileName;

}

// Abstract method to open a document

abstract void open();

}

class WordDocument extends Document {

// Constructor

WordDocument(String fileName) {

super(fileName);

}

// Implementing the open method for WordDocument

@Override

void open() {

System.out.println("Opening Word document: " + fileName);

}

}

//Subclass PDFDocument

class PDFDocument extends Document {

// Constructor

PDFDocument(String fileName) {

super(fileName);

}

// Implementing the open method for PDFDocument

@Override

void open() {

System.out.println("Opening PDF document: " + fileName);

}

}

//Subclass SpreadsheetDocument

class SpreadsheetDocument extends Document {

// Constructor

SpreadsheetDocument(String fileName) {

super(fileName);

}

// Implementing the open method for SpreadsheetDocument

@Override

void open() {

System.out.println("Opening Spreadsheet document: " + fileName);

}

}

public class DocumentOpenerDemo {

public static void main(String[] args) {

// Creating objects of different document types

Document wordDoc = new WordDocument("report.docx");

Document pdfDoc = new PDFDocument("sample.pdf");

Document spreadsheetDoc = new SpreadsheetDocument("financials.xlsx");

// Opening different types of documents

wordDoc.open();

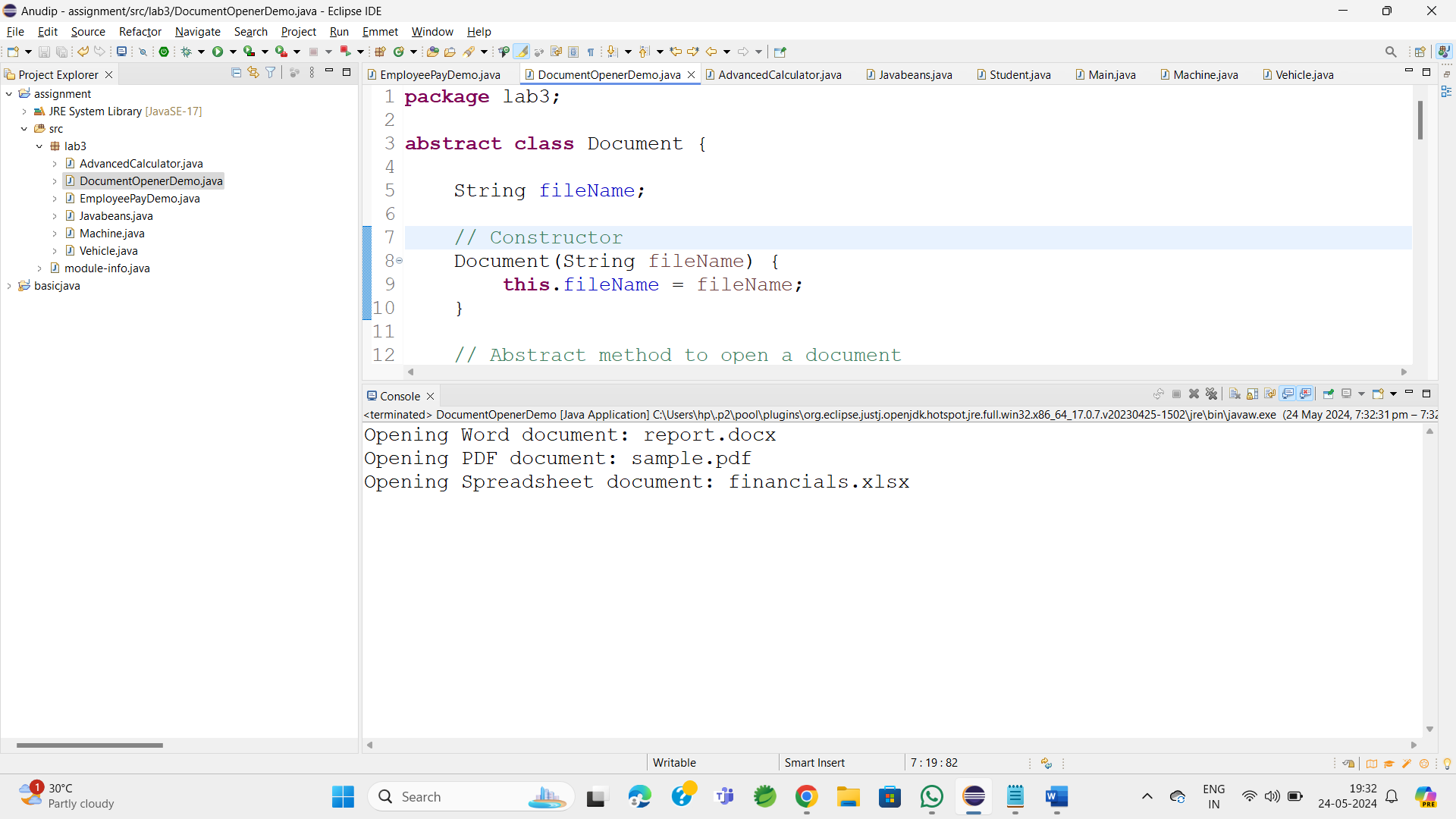
pdfDoc.open();

spreadsheetDoc.open();

}

}

OutPut:



1. **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)**

package lab3;

class 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;

}

}

public class CalculatorwithOverloaded {

public static void main(String[] args) {

cal c = new cal();

int sum = c.add(78, 35);

double sum1 = c.add(5.5, 4.7);

int sumThreeInt = c.add(5,10,15);

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

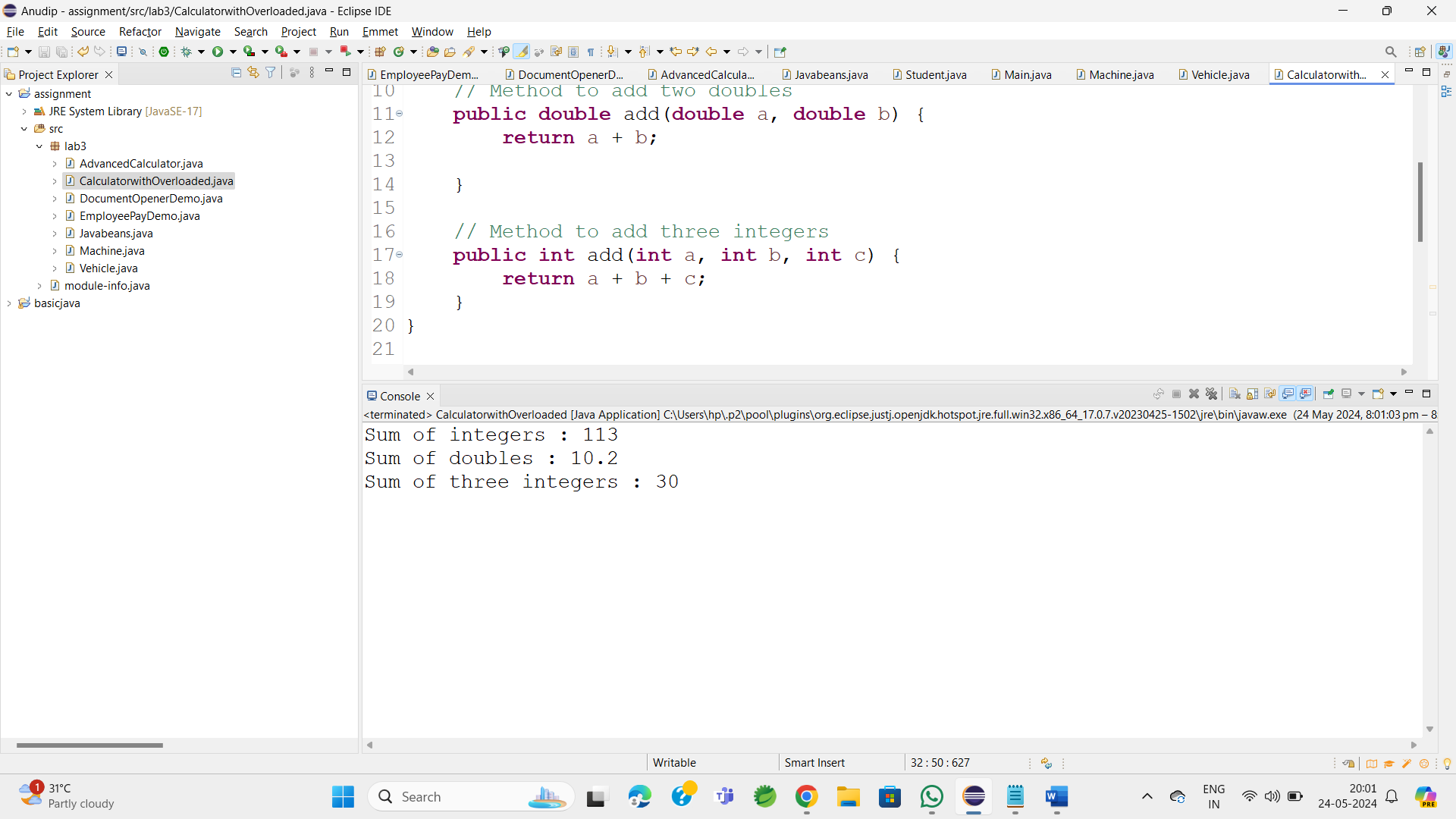
System.out.println("Sum of doubles : " + sum1);

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

}

}

**OutPut:**

****

**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.**

package lab3;

class Person {

private String firstName;

private String lastName;

private int age;

private String email;

// No-argument constructor

public Person() {

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

}

public class Javabeans {

public static void main(String[] args) {

Person person = new Person();

// Set properties

person.setFirstName("vijaya");

person.setLastName("ullendula");

person.setAge(22);

person.setEmail("sushilvijaya@gmail.com.com");

// Print properties

System.out.println("First Name: " + person.getFirstName());

System.out.println("Last Name: " + person.getLastName());

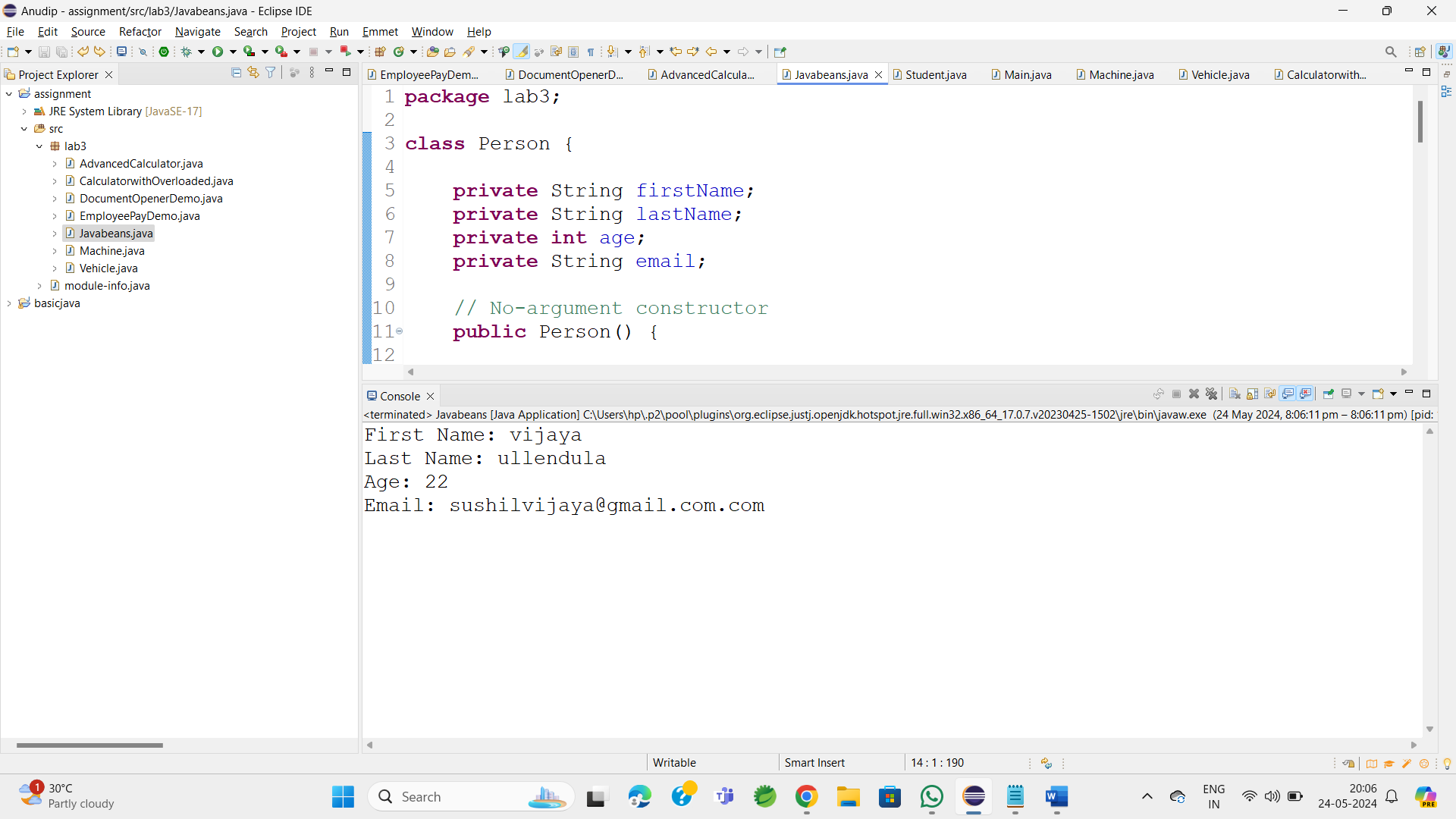
System.out.println("Age: " + person.getAge());

System.out.println("Email: " + person.getEmail());

}

}

**Output:**

****

**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.**

**package** lab3;

**class** Car1 {

**private** String make;

**private** String model;

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

**private** String color;

// No-argument constructor

**public** Car1() {

}

**public** String getMake() {

**return** make;

}

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

**this**.make = make;

}

**public** String getModel() {

**return** model;

}

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

**this**.model = model;

}

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

**return** year;

}

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

**this**.year = year;

}

**public** String getColor() {

**return** color;

}

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

**this**.color = color;

}

}

**public** **class** Machine {

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

Car1 car = **new** Car1();

// Set properties

car.setMake("Toyota");

car.setModel("Corolla");

car.setYear(2021);

car.setColor("Red");

// Print car details

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

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

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

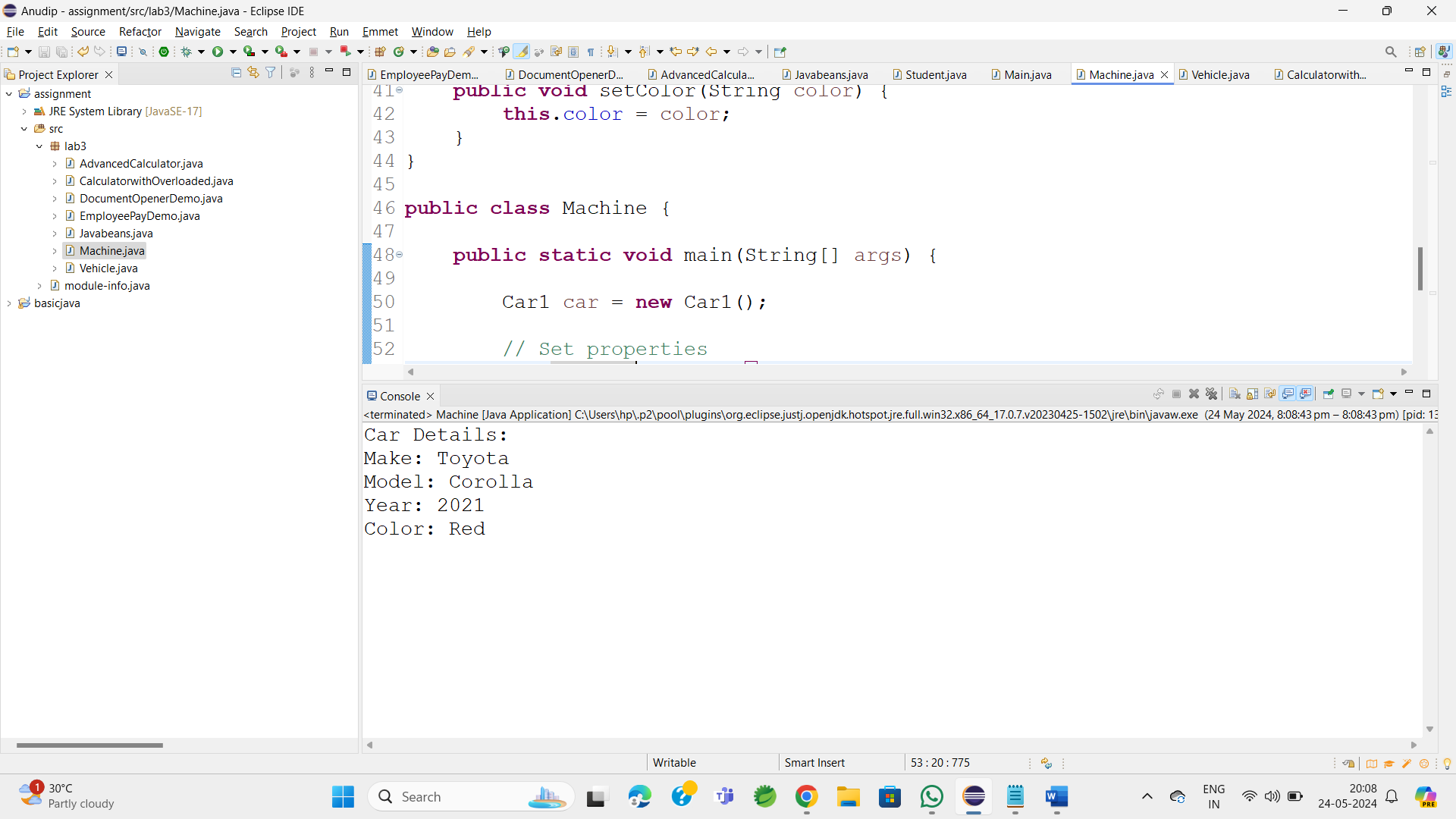
System.***out***.println("Year: " + car.getYear());

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

}

}

**Output:**

****