* Lab\_3

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.

**Code:**

**package** lab\_3; // Define the package name

// Superclass Person

**class** Person {

// Private attributes for name and age

**private** String name;

**private** **int** age;

// Constructor to initialize name and age

**public** Person(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

// Method to display name and age

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

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

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

}

}

// Subclass Student extends Person

**class** Student **extends** Person {

// Private attribute for studentID

**private** String studentID;

// Constructor to initialize name, age, and studentID

**public** Student(String name, **int** age, String studentID) {

**super**(name, age); // Call the superclass constructor

**this**.studentID = studentID;

}

// Override the display method to include studentID

@Override

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

**super**.display(); // Call the superclass display method

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

}

}

// Main class to test the Person and Student classes

**public** **class** main\_person {

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

// Create a Student object with name, age, and studentID

Student student = **new** Student("Sailee", 22, "AF0403143");

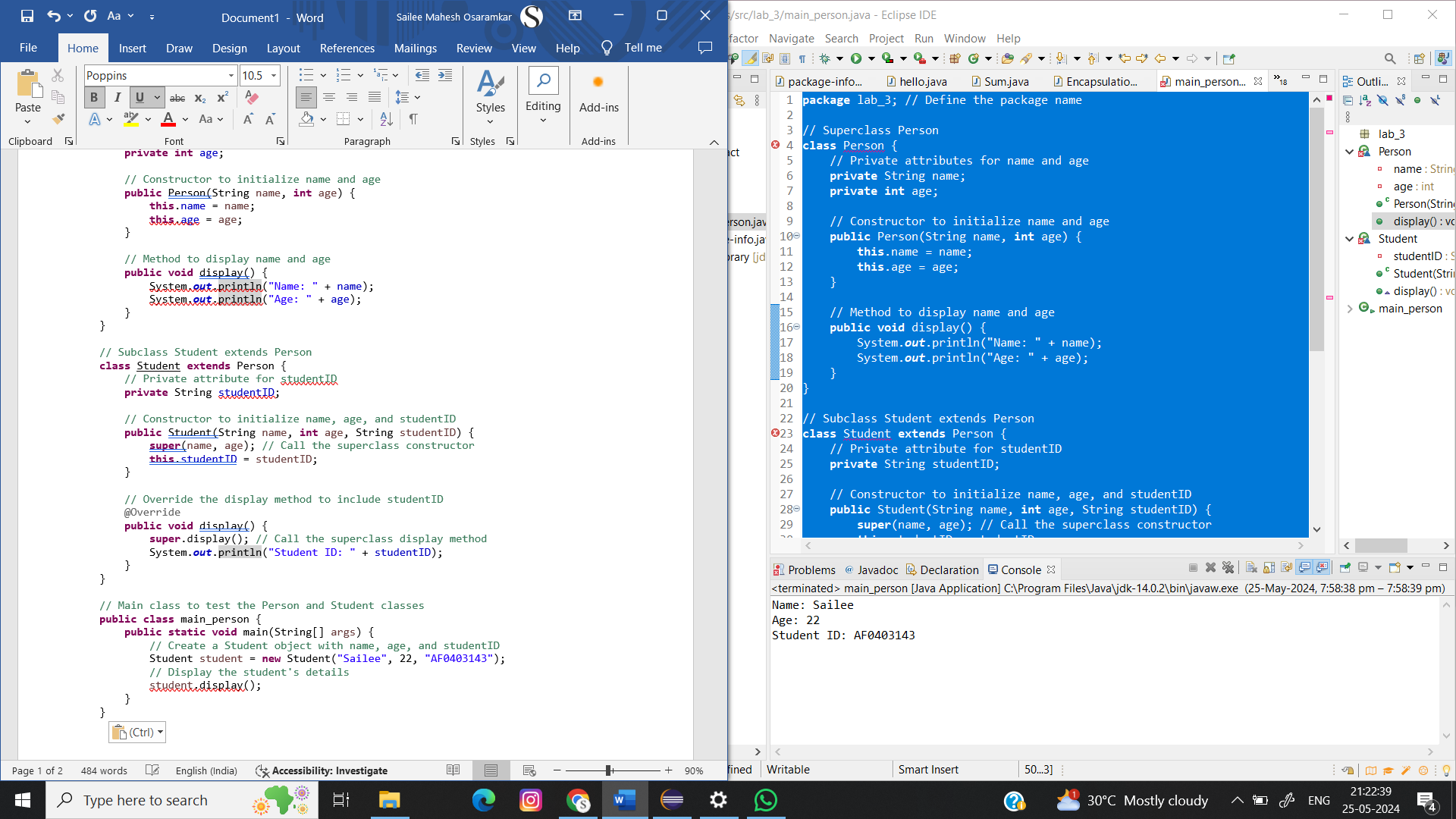
// Display the student's details

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.

**Code:**

**package** lab\_3;

//Superclass Calculator

**class** Calculator {

// Method to add two integers

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

**return** a + b;

}

}

//Subclass AdvancedCalculator

**class** AdvancedCalculator **extends** Calculator {

// Overloaded method to add three integers

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

**return** a + b + c;

}

}

//Main class to test the Calculator and AdvancedCalculator classes

**public** **class** main\_cal {

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

// Creating an instance of Calculator

Calculator calc = **new** Calculator();

**int** sum2 = calc.add(5, 10);

System.***out***.println("Sum of 5 and 10: " + sum2);

// Creating an instance of AdvancedCalculator

AdvancedCalculator advCalc = **new** AdvancedCalculator();

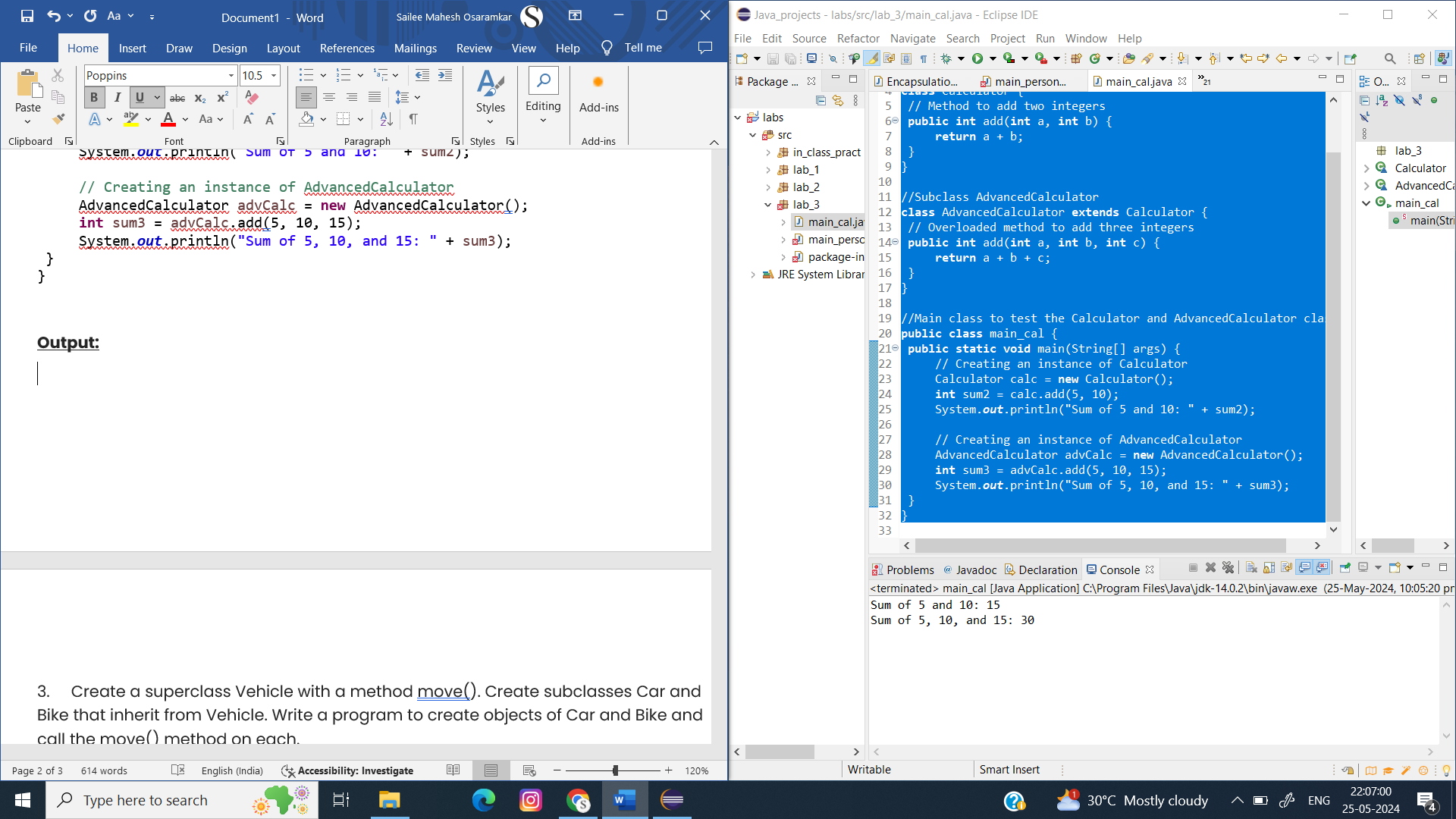
**int** sum3 = advCalc.add(5, 10, 15);

System.***out***.println("Sum of 5, 10, and 15: " + sum3);

}

}

**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.

**Code:**

**package** lab\_3;

//Superclass Vehicle

**class** Vehicle {

// Method to be overridden by subclasses

**public** **void** move() {

System.***out***.println("The vehicle is moving");

}

}

//Subclass Car

**class** Car **extends** Vehicle {

// Override the move method for Car

**public** **void** move() {

System.***out***.println("The car is driving");

}

}

//Subclass Bike

**class** Bike **extends** Vehicle {

// Override the move method for Bike

**public** **void** move() {

System.***out***.println("The bike is riding");

}

}

//Main class to test the Vehicle, Car, and Bike classes

**public** **class** main\_veh {

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

// Creating an instance of Car

Vehicle car = **new** Car();

car.move(); // Output: The car is driving

// Creating an instance of Bike

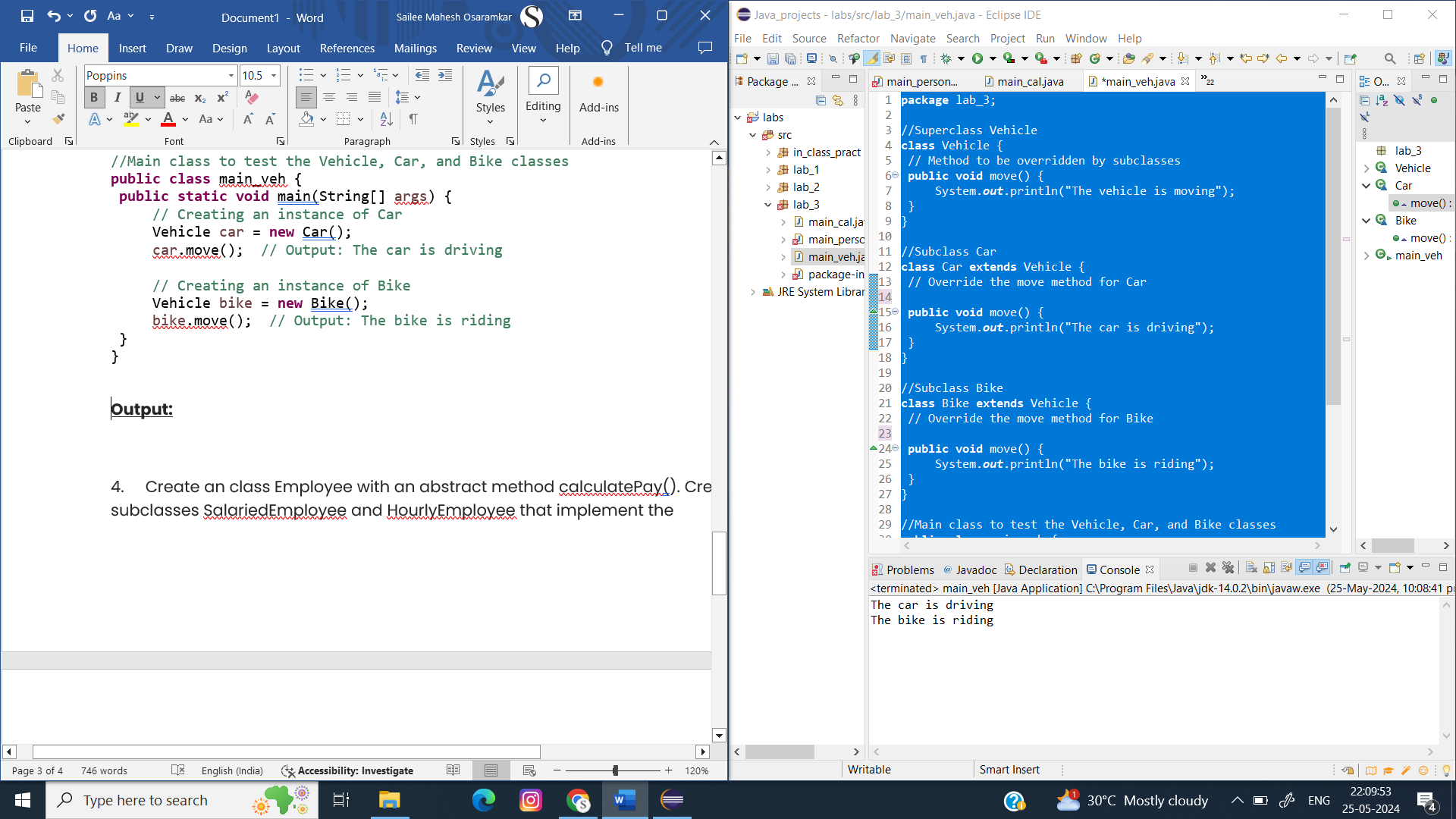
Vehicle bike = **new** Bike();

bike.move(); // Output: The bike is riding

}

}

**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.

**Code:**

**package** lab\_3;

//Abstract class Employee

**abstract** **class** Employee {

**protected** String name;

// Constructor to initialize name

**public** Employee(String name) {

**this**.name = name;

}

// Abstract method to calculate pay

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

}

//Subclass SalariedEmployee

**class** SalariedEmployee **extends** Employee {

**private** **double** annualSalary;

// Constructor to initialize name and annualSalary

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

**super**(name);

**this**.annualSalary = annualSalary;

}

// Implement the calculatePay method for salaried employees

**public** **double** calculatePay() {

**return** annualSalary / 12; // Monthly pay

}

}

//Subclass HourlyEmployee

**class** HourlyEmployee **extends** Employee {

**private** **double** hourlyRate;

**private** **int** hoursWorked;

// Constructor to initialize name, hourlyRate, and hoursWorked

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

**super**(name);

**this**.hourlyRate = hourlyRate;

**this**.hoursWorked = hoursWorked;

}

// Implement the calculatePay method for hourly employees

**public** **double** calculatePay() {

**return** hourlyRate \* hoursWorked; // Total pay based on hours worked

}

}

//Main class to test the Employee, SalariedEmployee, and HourlyEmployee classes

**public** **class** main\_emp {

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

// Creating an instance of SalariedEmployee

Employee salariedEmployee = **new** SalariedEmployee("Alice", 60000);

System.***out***.println("Monthly pay for " + salariedEmployee.name + ": $" + salariedEmployee.calculatePay());

// Creating an instance of HourlyEmployee

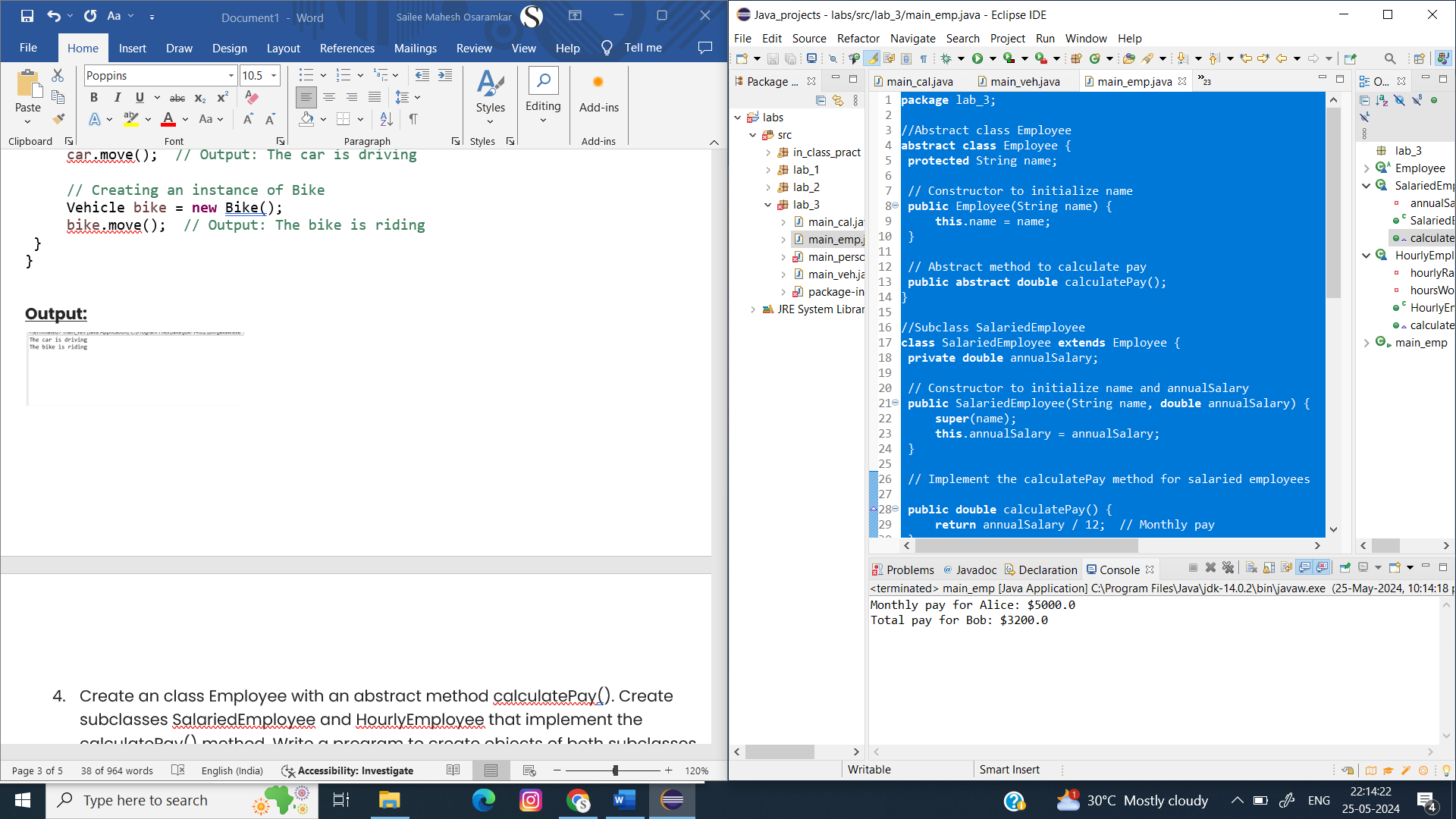
Employee hourlyEmployee = **new** HourlyEmployee("Bob", 20, 160);

System.***out***.println("Total pay for " + hourlyEmployee.name + ": $" + 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).

**Code:**

**package** lab\_3;

//Superclass Document

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

// Abstract method open

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

}

//Subclass WordDocument

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

// Implement the open method for WordDocument

@Override

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

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

}

}

//Subclass PDFDocument

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

// Implement the open method for PDFDocument

@Override

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

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

}

}

//Subclass SpreadsheetDocument

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

// Implement the open method for SpreadsheetDocument

@Override

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

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

}

}

//Main class to demonstrate opening different types of documents

**public** **class** main\_doc {

// Method demonstrating compile-time polymorphism

**public** **static** **void** openDocument(Document doc) {

doc.open();

}

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

// Create instances of different document types

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

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

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

// Demonstrate opening different types of documents

*openDocument*(wordDoc); // Output: Opening Word document...

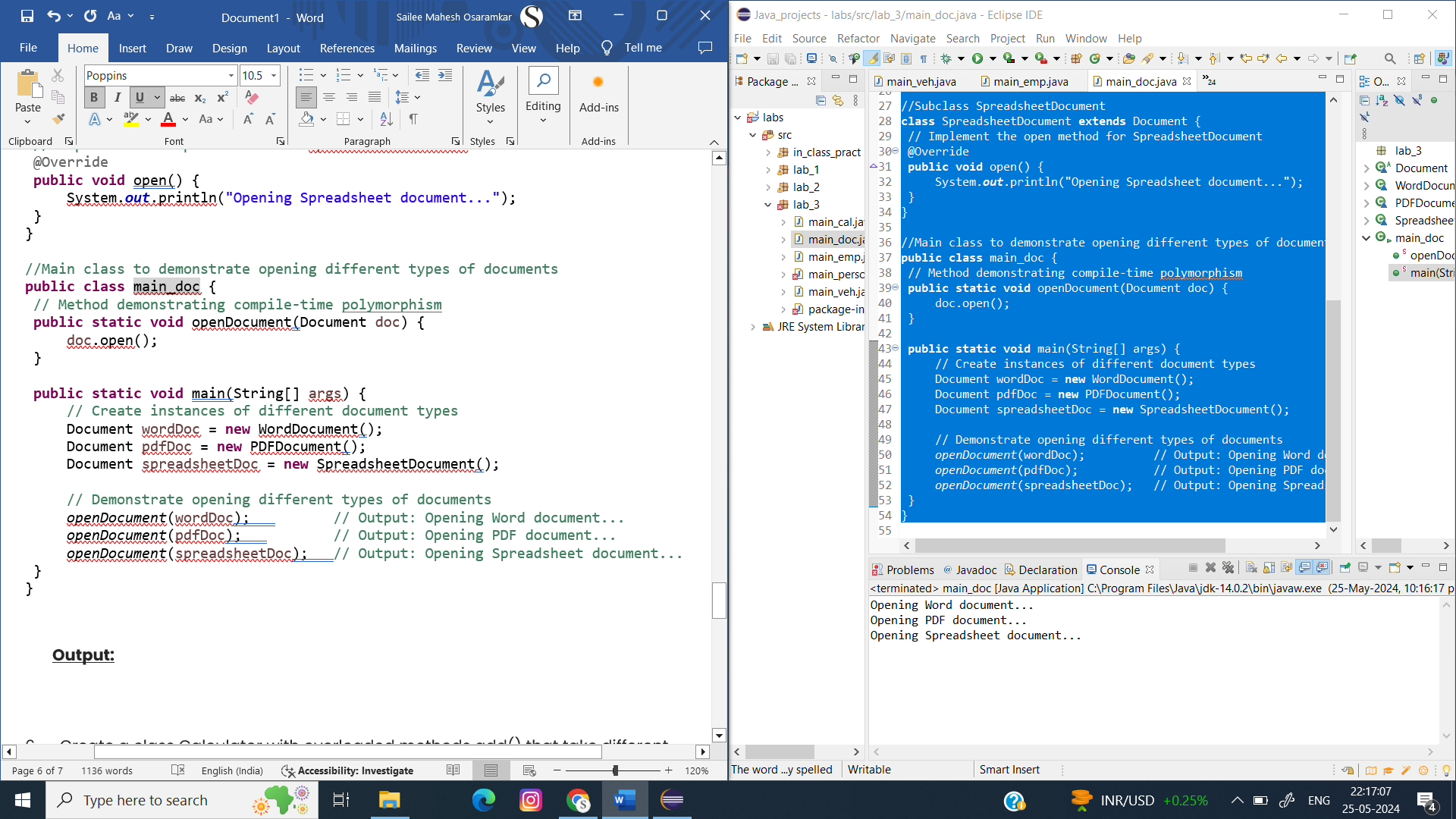
*openDocument*(pdfDoc); // Output: Opening PDF document...

*openDocument*(spreadsheetDoc); // Output: Opening Spreadsheet document...

}

}

**Output:**



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.

**Code:**

**package** lab\_3; // Ensure this matches your package declaration

// Class Calculator

**class** Calculator {

// Method to add two integers

**public** **int** add(**int** a, **int** 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 two double values

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

**return** a + b;

}

// Method to add three double values

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

**return** a + b + c;

}

// 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 double values

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

**double** sum = 0;

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

sum += num;

}

**return** sum;

}

}

// Main class to test the Calculator class

**public** **class** main {

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

Calculator calc = **new** Calculator();

// Test the overloaded add methods

System.***out***.println("Sum of 5 and 10: " + calc.add(5, 10)); // int add(int, int)

System.***out***.println("Sum of 5, 10 and 15: " + calc.add(5, 10, 15)); // int add(int, int, int)

System.***out***.println("Sum of 5.5 and 10.5: " + calc.add(5.5, 10.5)); // double add(double, double)

System.***out***.println("Sum of 5.5, 10.5 and 15.5: " + calc.add(5.5, 10.5, 15.5)); // double add(double, double, double)

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

System.***out***.println("Sum of int array: " + calc.add(intArray)); // int add(int[])

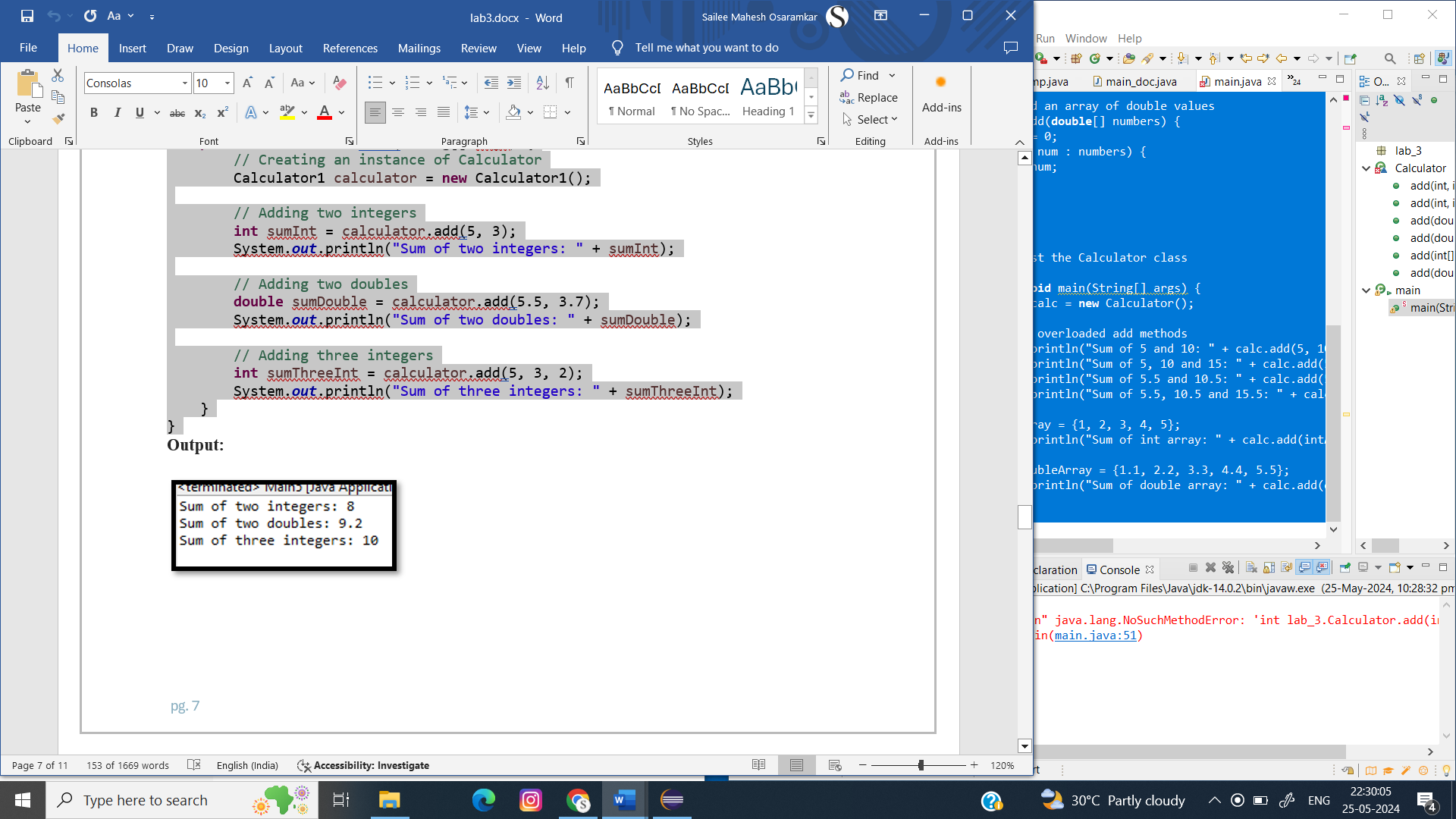
**double**[] doubleArray = {1.1, 2.2, 3.3, 4.4, 5.5};

System.***out***.println("Sum of double array: " + calc.add(doubleArray)); // double add(double[])

}

}

**Output:**



1. 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.

**Code:**

**package** in\_class\_pract;

**import** java.io.Serializable;

**public** **class** Person **implements** Serializable {

**private** String firstName;

**private** String lastName;

**private** **int** age;

**private** String email;

// No-argument constructor

**public** Person() {

}

// Getter for firstName

**public** String getFirstName() {

**return** firstName;

}

// Setter for firstName

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

// Getter for lastName

**public** String getLastName() {

**return** lastName;

}

// Setter for lastName

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

// Getter for age

**public** **int** getAge() {

**return** age;

}

// Setter for age

**public** **void** setAge(**int** age) {

**this**.age = age;

}

// Getter for email

**public** String getEmail() {

**return** email;

}

// Setter for email

**public** **void** setEmail(String email) {

**this**.email = email;

}

// Override toString() method to print Person details

@Override

**public** String toString() {

**return** "Person [firstName=" + firstName + ", lastName=" + lastName + ", age=" + age + ", email=" + email + "]";

}

}

**package** in\_class\_pract;

**public** **class** Main {

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

// Create an instance of Person

Person person = **new** Person();

// Set properties

person.setFirstName("Sailee");

person.setLastName("Osaramkar");

person.setAge(22);

person.setEmail("osaramkarsailee14@gmail.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());

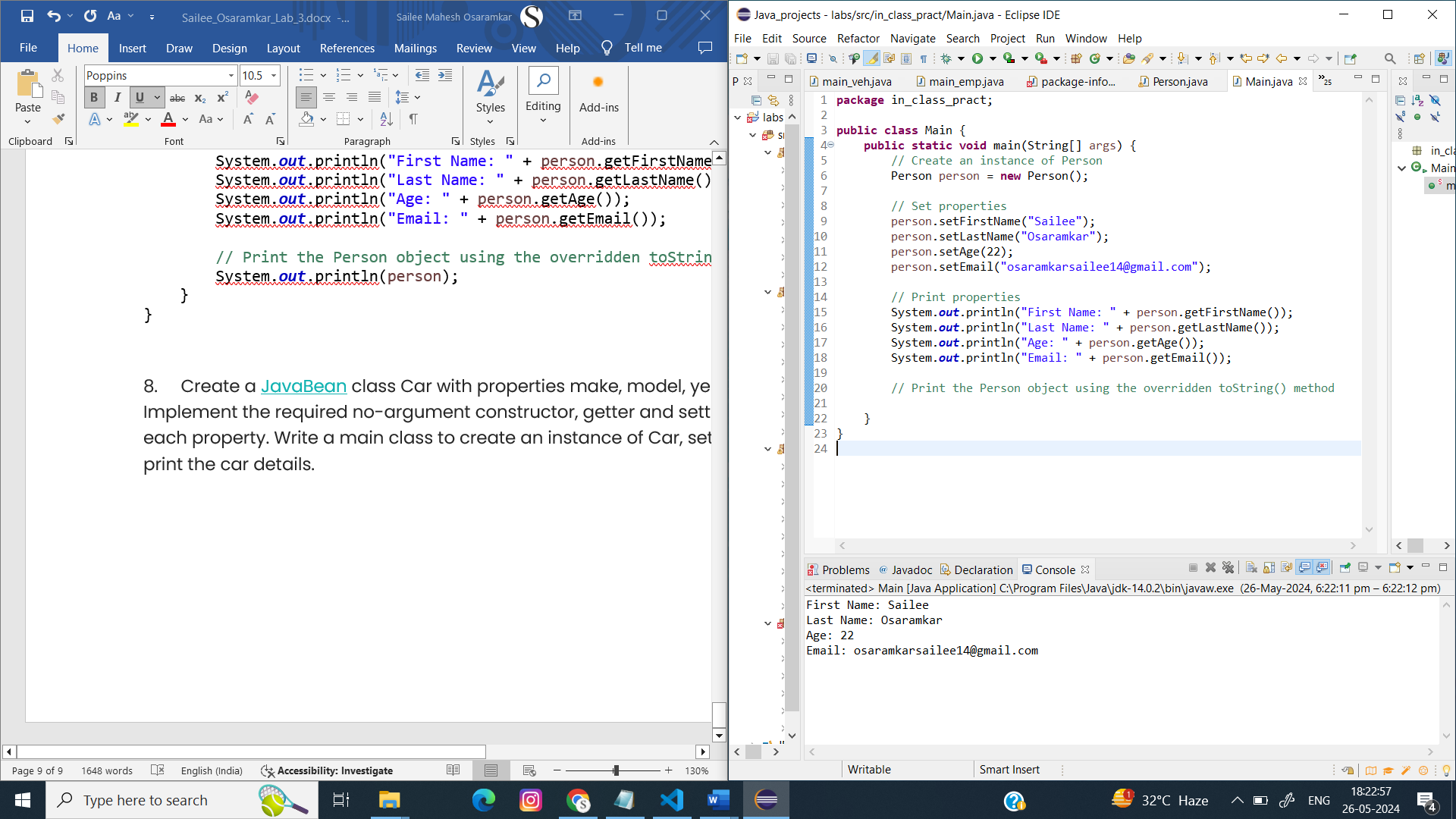
// Print the Person object using the overridden toString() method

System.***out***.println(person);

}

}

**Output:**



1. 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.

**Code:**

**package** in\_class\_pract;

**import** java.io.Serializable;

**public** **class** car **implements** Serializable {

**private** String make;

**private** String model;

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

**private** String color;

// No-argument constructor

**public** car() {

}

// Getter for make

**public** String getMake() {

**return** make;

}

// Setter for make

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

**this**.make = make;

}

// Getter for model

**public** String getModel() {

**return** model;

}

// Setter for model

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

**this**.model = model;

}

// Getter for year

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

**return** year;

}

// Setter for year

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

**this**.year = year;

}

// Getter for color

**public** String getColor() {

**return** color;

}

// Setter for color

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

**this**.color = color;

}

// Override toString() method to print Car details

@Override

**public** String toString() {

**return** "Car [make=" + make + ", model=" + model + ", year=" + year + ", color=" + color + "]";

}

}

**package** in\_class\_pract;

**public** **class** main\_ser {

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

// Create an instance of Car

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

// Set properties

car.setMake("Toyota");

car.setModel("Corolla");

car.setYear(2020);

car.setColor("Blue");

// Print properties

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:**

