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** Demo;

**class** Person {

String name;

**int** age;

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

**this**.name = name;

**this**.age = age;

}

// Method to display attributes

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

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

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

}

}

// Subclass Student inheriting from Person

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

**int** studentID;

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

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

**this**.studentID = studentID;

}

// Method to display all attributes including studentID

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

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

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

}

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

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

// **TODO** Auto-generated method stub

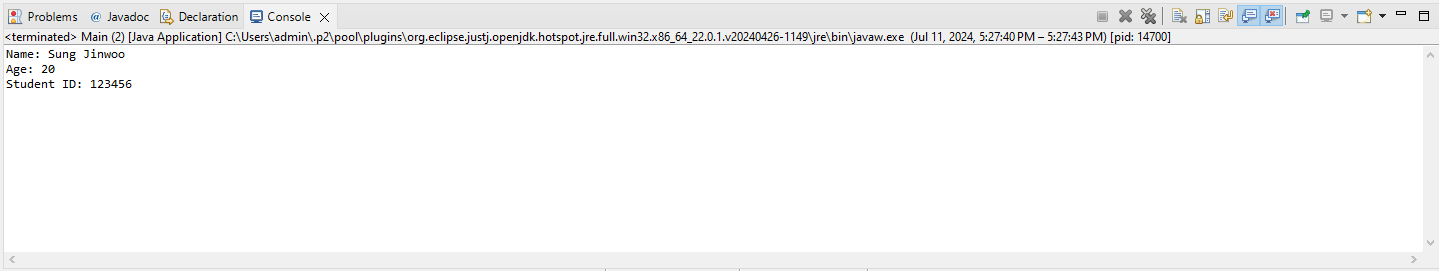
Student student1 = **new** Student("Sung Jinwoo", 20, 123456);

student1.display();

}

}

}

Output:-

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** cal\_demo;

**class** Calculator {

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

**return** a + b;

}

}

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

// Overloaded method to add three integers

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

**return** a + b + c;

}

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

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

// **TODO** Auto-generated method stub

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

// Use the add method from Calculator (adding two numbers)

**int** result1 = calc.add(10, 20);

System.***out***.println("Result of adding two numbers: " + result1);

// Create an AdvancedCalculator object

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

// Use the add method from AdvancedCalculator (adding three numbers)

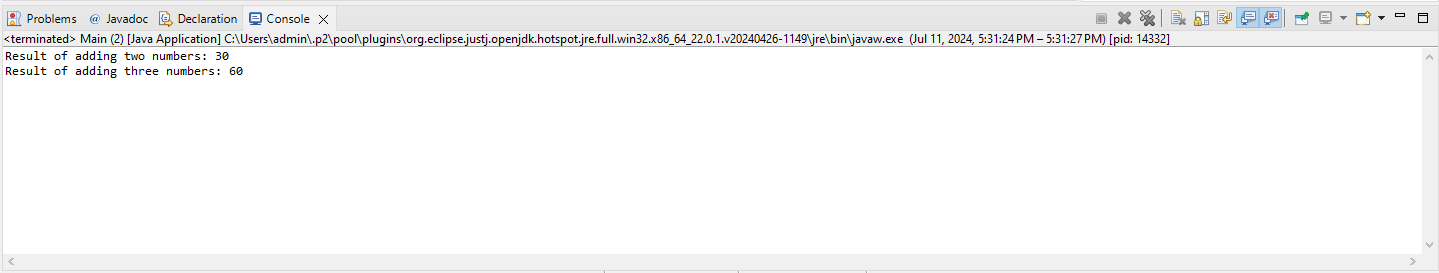
**int** result2 = advCalc.add(10, 20, 30);

System.***out***.println("Result of adding three numbers: " + result2);

}

}

}

Output:-

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** Demo;

**class** Vehicle {

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

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

}

}

// Subclass Car inheriting from Vehicle

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

}

// Subclass Bike inheriting from Vehicle

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

}

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

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

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

// **TODO** Auto-generated method stub

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

car.move();

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

bike.move();

}

}

Output:-

4.     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** cal\_demo;

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

**public** **abstract** **void** calculatePay();

}

// Subclass SalariedEmployee inheriting from Employee

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

**private** **double** salary;

**public** SalariedEmployee(**double** salary) {

**this**.salary = salary;

}

**public** **void** calculatePay() {

System.***out***.println("Salaried Employee's pay: " + salary);

}

}

// Subclass HourlyEmployee inheriting from Employee

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

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

**private** **double** hoursWorked;

**public** HourlyEmployee(**double** hourlyRate, **double** hoursWorked) {

**this**.hourlyRate = hourlyRate;

**this**.hoursWorked = hoursWorked;

}

**public** **void** calculatePay() {

**double** pay = hourlyRate \* hoursWorked;

System.***out***.println("Hourly Employee's pay: " + pay);

}

}

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

**public** **class** Main1 {

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

// **TODO** Auto-generated method stub

SalariedEmployee salariedEmployee = **new** SalariedEmployee(50000.0);

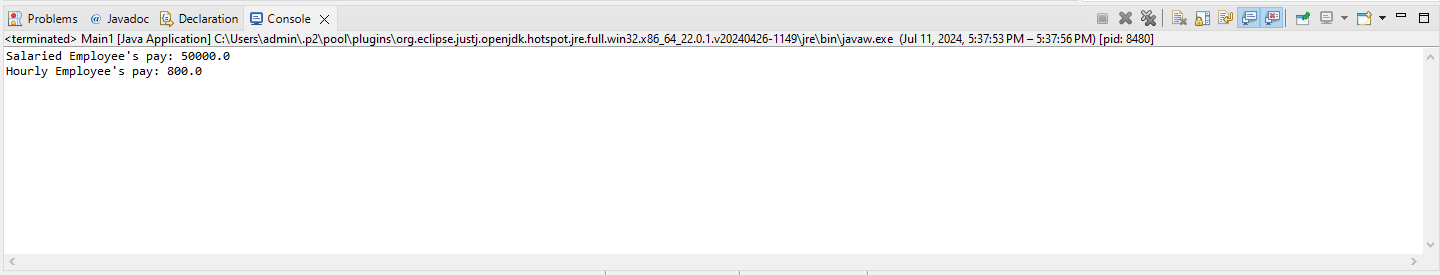
salariedEmployee.calculatePay();

HourlyEmployee hourlyEmployee = **new** HourlyEmployee(20.0, 40.0);

hourlyEmployee.calculatePay();

}

}

Output:-

5.     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** numbers\_demo;

**class** Document {

// Method to open document

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

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

}

}

// Subclass WordDocument inheriting from Document

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

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

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

}

}

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

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

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

}

}

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

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

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

}

}

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

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

// **TODO** Auto-generated method stub

Document doc1 = **new** WordDocument();

Document doc2 = **new** PDFDocument();

Document doc3 = **new** SpreadsheetDocument();

// Call open method on each document

doc1.open();

doc2.open();

doc3.open();

}

}

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.

Program:-

**package** sum\_demo;

**class** Calculator {

// 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** Main {

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

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

// **TODO** Auto-generated method stub

**int** result1 = calc.add(10, 20);

**double** result2 = calc.add(10.5, 20.7);

**int** result3 = calc.add(10, 20, 30);

System.***out***.println("Result of adding two integers: " + result1);

System.***out***.println("Result of adding two doubles: " + result2);

System.***out***.println("Result of adding three integers: " + result3);

}

}

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.

Program:-

**package** bean\_demo;

**class** Person {

**private** String firstName;

**private** String lastName;

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

**private** String email;

**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** Main {

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

// **TODO** Auto-generated method stub

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

// Set properties using setter methods

person.setFirstName("Sung");

person.setLastName("Jinwoo");

person.setAge(20);

person.setEmail("sung.jinwoo@619gmail.com");

// Print out the person's details using getter methods

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

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

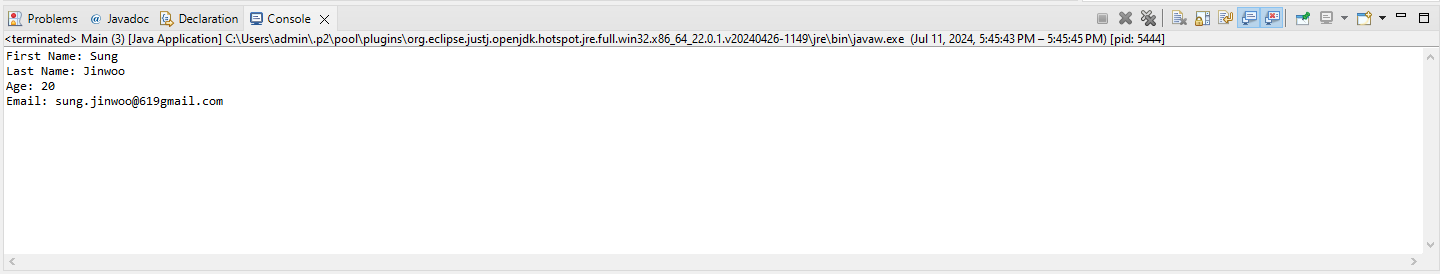
System.***out***.println("Age: " + person.getAge());

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.

Program:-

**package** constructor\_demo;

**class** Car {

**private** String make;

**private** String model;

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

**private** String color;

**public** Car() {

}

**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** Main {

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

// **TODO** Auto-generated method stub

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

// Set properties using setter methods

car.setMake("Toyota");

car.setModel("Supra");

car.setYear(2002);

car.setColor("Orange-Pearl");

// Print out the car's details using getter methods

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

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

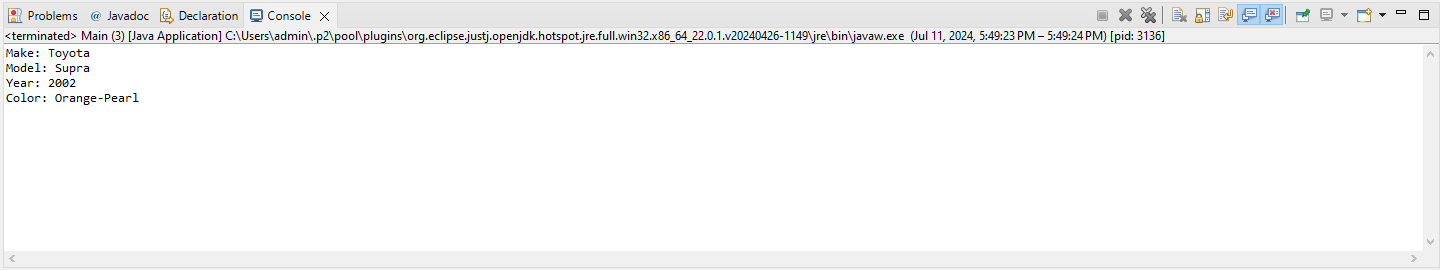
System.***out***.println("Year: " + car.getYear());

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

}

}

}

Output:-