**Q1 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** Assignments;

**class** Person {

String name;

**int** age;

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

**this**.name = name;

**this**.age = age;

}

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

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

}

}

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

String studentID;

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

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

**this**.studentID = studentID;

}

}

**class** Main {

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

Student student = **new** Student("Bob", 25, "ABC123");

student.display();

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

}

}

**O/P:- **

**Q2 Create a superclass Calculator with a method add(int a, int b). Create a subclass Advanced Calculator that overloads the add method to handle three integers.**

**Code:-**

**package** Assignments;

**class** Calculator {

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

**return** a + b;

}

}

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

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

**return** a + b + c;

}

}

**class** Main {

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

// Use Calculator class for two arguments

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

**int** sumTwo = calc.add(5, 3);

System.***out***.println("Sum of two numbers (using Calculator): " + sumTwo);

// Use AdvancedCalculator class for three arguments

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

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

System.***out***.println("Sum of three numbers (using AdvancedCalculator): " + sumThree);

}

}

**O/p:- **

## Q3) 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** Assignments;

**class** Vehicle {

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

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

}

}

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

@Override

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

System.***out***.println("Car is driving.");

}

}

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

@Override

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

System.***out***.println("Bike is riding.");

}

}

**class** Main {

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

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

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

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

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

}

}

## O/p:-

Q4) **Create an class Employee with an abstract method calculatePay(). Create subclasses Salaried Employee and Hourly Employee that implement the calculatePay() method. Write a program to create objects of both subclasses and call the calculatePay() method.**

**Code:-**

**package** Assignments;

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

**protected** String name;

**protected** **int** id;

**public** Employee(String name, **int** id) {

**this**.name = name;

**this**.id = id;

}

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

}

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

**private** **double** monthlySalary;

**public** SalariedEmployee(String name, **int** id, **double** monthlySalary) {

**super**(name, id);

**this**.monthlySalary = monthlySalary;

}

@Override

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

**return** monthlySalary;

}

}

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

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

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

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

**super**(name, id);

**this**.hourlyRate = hourlyRate;

**this**.hoursWorked = hoursWorked;

}

@Override

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

**return** hourlyRate \* hoursWorked;

}

}

**class** Main {

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

Employee salariedEmployee = **new** SalariedEmployee("Alice", 123, 4000.00);

Employee hourlyEmployee = **new** HourlyEmployee("Bob", 456, 20.00, 40);

System.***out***.println("Salary for " + salariedEmployee.name + ": **₹**" + salariedEmployee.calculatePay());

System.***out***.println("Pay for " + hourlyEmployee.name + ": **₹**" + hourlyEmployee.calculatePay());

}

}

O/p:- 

**Q5) Create an class Document with an method void open(). Implement subclasses Word Document, PDF Document, and Spreadsheet Document 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** Assignments;

**class** Document {

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

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

}

}

**class** Main {

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

Document doc;

doc = **new** WordDocument();

doc.open(); // Output: Opening a Word document...

doc = **new** PDFDocument();

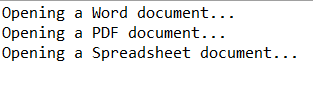
doc.open(); // Output: Opening a PDF document...

doc = **new** SpreadsheetDocument();

doc.open(); // Output: Opening a Spreadsheet document...

}

}

**O/p:- **

**Q6) 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** Assignments;

**class** Calculator {

// Method to add two integers

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

**return** a + b;

}

// Method to add two doubles

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

**return** a + b;

}

// Method to add three integers

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

**return** a + b + c;

}

}

**class** Main {

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

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

// Using add() for two integers

**int** sumInt = calc.add(5, 3);

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

// Using add() for two doubles

**double** sumDouble = calc.add(10.5, 20.75);

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

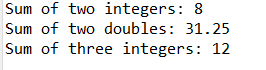
// Using add() for three integers

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

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

}

}

**O/p:- **

**Q7) Create a JavaBean 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** Assignments;

**public** **class** Person {

**private** String firstName;

**private** String lastName;

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

**private** String email;

// No-argument constructor (required for JavaBeans)

**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;

}

@Override

**public** String toString() {

**return** "Person{" +

"firstName='" + firstName + '\'' +

", lastName='" + lastName + '\'' +

", age=" + age +

", email='" + email + '\'' +

'}';

}

}

**class** Main {

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

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

person.setFirstName("Omkar");

person.setLastName("Powar");

person.setAge(24);

person.setEmail("omkar.powar@example.com");

System.***out***.println(person); // Prints all properties using the toString() method

}

}

**O/p:- **

**Q8.** **Create a JavaBean 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** Assignments;

**public** **class** Car {

**private** String make;

**private** String model;

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

**private** String color;

// No-argument constructor (required for JavaBeans)

**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;

}

@Override

**public** String toString() {

**return** "Car{" +

"make='" + make + '\'' +

", model='" + model + '\'' +

", year=" + year +

", color='" + color + '\'' +

'}';

}

}

**class** Main {

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

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

car.setMake("Toyota");

car.setModel("Camry");

car.setYear(2023);

car.setColor("Red");

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

System.***out***.println(car); // Prints all properties using the toString() method

}

}

**O/P:- **