**LAB3**

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 demo1;

// Superclass Person

class Person {

// Attributes

protected String name;

protected int age;

// Constructor

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

class Student extends Person {

// Additional attribute

private String studentID;

// Constructor

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

// Call the constructor of the superclass

super(name, age);

this.studentID = studentID;

}

// Override the display method to include studentID

*@Override*

public void display() {

// Call the display method of the superclass

super.display();

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

}

}

// Main class to create a Student object and display its attributes

public class Main {

public static void main(String[] args) {

// Create a Student object

Student student = new Student("pratham jagdhane", 21, "anp-12345");

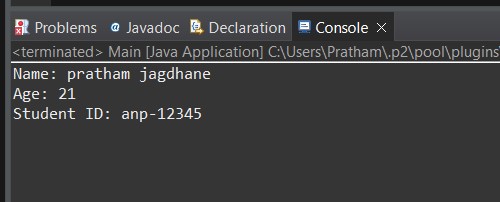
// Display all attributes

student.display();

}

}

**Output-**



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 demo1;

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 functionality

public class Main12 {

public static void main(String[] args) {

Calculator calculator = new Calculator();

// Use the add method to add two integers

System.***out***.println("Sum of 10 and 20: " + calculator.add(10, 20));

// Create an AdvancedCalculator object

AdvancedCalculator advancedCalculator = new AdvancedCalculator();

// Use the overloaded add method to add two integers

System.***out***.println("Sum of 30 and 40: " + advancedCalculator.add(30, 40));

// Use the overloaded add method to add three integers

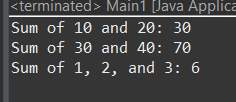
System.***out***.println("Sum of 1, 2, and 3: " + advancedCalculator.add(1, 2, 3));

}

}

}

**Output-**



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 demo1;

//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 {

// Overriding the move method

*@Override*

public void move() {

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

}

}

//Subclass Bike

class Bike extends Vehicle {

// Overriding the move method

*@Override*

public void move() {

System.***out***.println("The bike is cycling on the path");

}

}

//Main class to test the functionality

public class Main2 {

public static void main(String[] args) {

// Create a Car object

Car car = new Car();

// Call the move method on the Car object

car.move();

// Create a Bike object

Bike bike = new Bike();

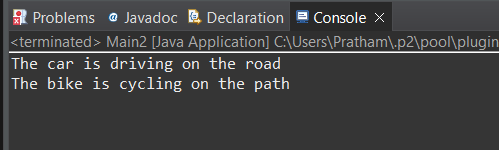
// Call the move method on the Bike object

bike.move();

}

}

**Output-**

****

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 demo1;

//Abstract class Employee

abstract class Employee {

// Abstract method to be implemented by subclasses

public abstract double calculatePay();

}

//Subclass SalariedEmployee

class SalariedEmployee extends Employee {

// Attributes for a salaried employee

private double annualSalary;

// Constructor

public SalariedEmployee(double annualSalary) {

this.annualSalary = annualSalary;

}

// Implementation of calculatePay method for salaried employee

*@Override*

public double calculatePay() {

return annualSalary / 12;

}

}

//Subclass HourlyEmployee

class HourlyEmployee extends Employee {

// Attributes for an hourly employee

private double hourlyRate;

private int hoursWorked;

// Constructor

public HourlyEmployee(double hourlyRate, int hoursWorked) {

this.hourlyRate = hourlyRate;

this.hoursWorked = hoursWorked;

}

// Implementation of calculatePay method for hourly employee

*@Override*

public double calculatePay() {

// Payment is calculated based on hours worked

return hourlyRate \* hoursWorked;

}

}

//Main class to test the functionality

public class Main7 {

public static void main(String[] args) {

// Create a SalariedEmployee object

SalariedEmployee salariedEmployee = new SalariedEmployee(60000);

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

// Create an HourlyEmployee object

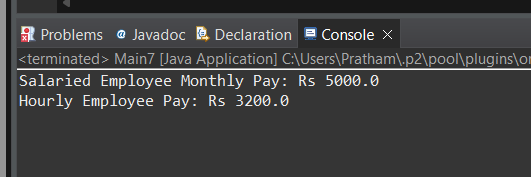
HourlyEmployee hourlyEmployee = new HourlyEmployee(20, 160);

System.***out***.println("Hourly Employee Pay: Rs " + hourlyEmployee.calculatePay());

}

}

**Output-**

****

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 demo1;

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...");

}

}

public class Main6 {

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...

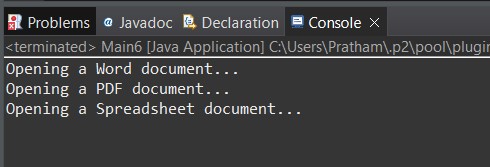
doc = new SpreadsheetDocument();

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

}

}

**Output-**



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 demo1;

class Calculator1 {

// 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;

}

}

public class Main5 {

public static void main(String[] args) {

Calculator1 calc = new Calculator1();

// 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

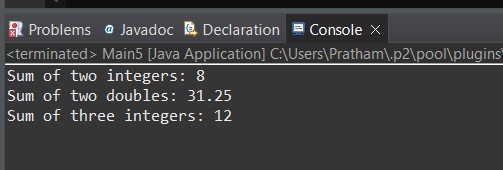
int sumThreeInt = calc.add(2, 4, 6);

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

}

}

**Output-**



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 demo1;

class Person1 {

private String firstName;

private String lastName;

private int age;

private String email;

// No-argument constructor (required for JavaBeans)

public Person1() {

}

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 + '\'' +

'}';

}

}

public class Main4 {

public static void main(String[] args) {

Person1 person = new Person1();

person.setFirstName("Pratham");

person.setLastName("Jagdhane");

person.setAge(21);

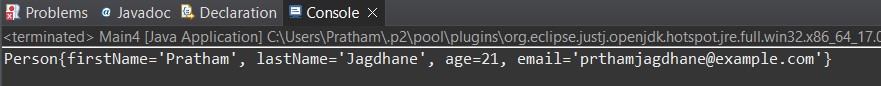
person.setEmail("prthamjagdhane@example.com");

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

}

}

**Output-**



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 demo1;

class Car1 {

private String make;

private String model;

private int year;

private String color;

// Constructor

public Car1() {

}

// Getter and Setter for make

public void setMake(String make) {

this.make = make;

}

public String getMake() {

return make;

}

// Getter and Setter for model

public void setModel(String model) {

this.model = model;

}

public String getModel() {

return model;

}

// Getter and Setter for year

public void setYear(int year) {

this.year = year;

}

public int getYear() {

return year;

}

// Getter and Setter for color

public void setColor(String color) {

this.color = color;

}

public String getColor() {

return color;

}

*@Override*

public String toString() {

return "Car{" +

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

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

", year=" + year +

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

'}';

}

}

public class Main3 {

public static void main(String[] args) {

Car1 car = new Car1();

car.setMake("Toyota");

car.setModel("Camry");

car.setYear(2023);

car.setColor("Red");

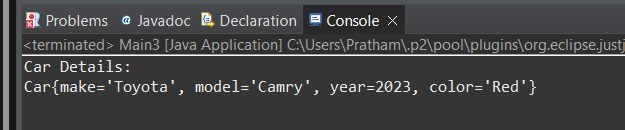
System.***out***.println("Car Details:");

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

}

}

**Output-**

****