LAB3 ASSIGNMENT

Q 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 Code :

**package** Lab3;

//Superclass Person

**class** Person {

String name;

**int** age;

// Constructor

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

**this**.name = name;

**this**.age = age;

}

// Method to display details

**void** display() {

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

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

}

}

//Subclass Student extending Person

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

**int** studentID;

// Constructor

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

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

**this**.studentID = studentID;

}

// Method to display all attributes including studentID

**void** display() {

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

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

}

}

//Main class to create Student object and display attributes

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

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

// Create a Student object

Student student = **new** Student("kajal", 21, 123456);

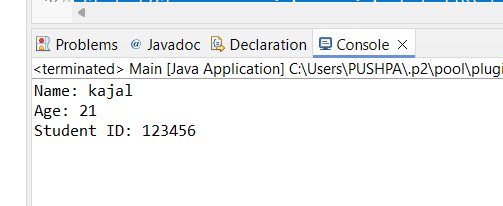
// Display student details

student.display();

}

}

OUTPUT:



Q 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 Code :

**package** Lab3;

//Superclass Calculator

**class** Calculator {

// Method to add two integers

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

**int** sum = a + b;

System.***out***.println("Sum of " + a + " and " + b + " is: " + sum);

}

}

//Subclass AdvancedCalculator extending Calculator

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

// Method overloading to add three integers

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

**int** sum = a + b + c;

System.***out***.println("Sum of " + a + ", " + b + " and " + c + " is: " + sum);

}

}

//Main class to test Calculator classes

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

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

Calculator basicCalc = **new** Calculator();

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

// Using superclass method

basicCalc.add(10, 20);

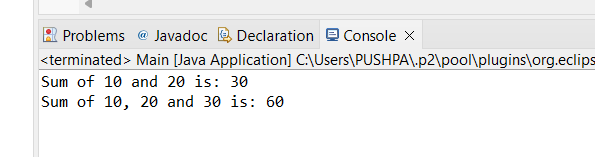
// Using subclass method with three parameters

advCalc.add(10, 20, 30);

}

}

OUTPUT :



Q 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 code:

// Superclass Vehicle

class Vehicle {

// Method to move

void move() {

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

}

}

// Subclass Car extending Vehicle

class Car extends Vehicle {

// Method overriding move() in Vehicle

@Override

void move() {

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

}

}

// Subclass Bike extending Vehicle

class Bike extends Vehicle {

// Method overriding move() in Vehicle

@Override

void move() {

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

}

}

// Main class to create objects of Car and Bike

public class Main {

public static void main(String[] args) {

// Create objects of Car and Bike

Car car = new Car();

Bike bike = new Bike();

// Call move() method on Car and Bike objects

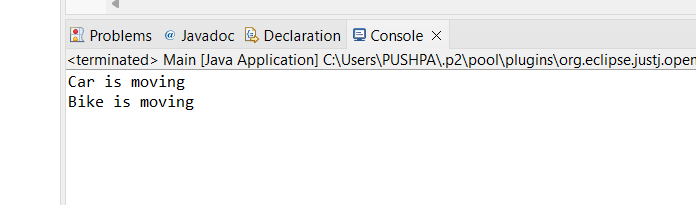
car.move();

bike.move();

}

}

OUTPUT :



Q 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 code:

**package** Lab3;

//Abstract superclass Employee

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

// Abstract method calculatePay()

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

}

//Subclass SalariedEmployee extending Employee

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

// Implementation of calculatePay() for SalariedEmployee

@Override

**void** calculatePay() {

System.***out***.println("Calculating salary for a salaried employee...");

// Add logic specific to salaried employee pay calculation

}

}

//Subclass HourlyEmployee extending Employee

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

// Implementation of calculatePay() for HourlyEmployee

@Override

**void** calculatePay() {

System.***out***.println("Calculating wages for an hourly employee...");

// Add logic specific to hourly employee pay calculation

}

}

//Main class to create objects of both subclasses and call calculatePay()

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

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

// Create objects of SalariedEmployee and HourlyEmployee

Employee salariedEmp = **new** SalariedEmployee();

Employee hourlyEmp = **new** HourlyEmployee();

// Call calculatePay() method on each object

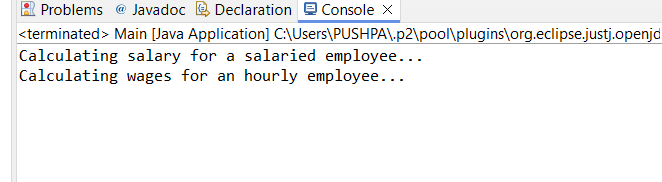
salariedEmp.calculatePay();

hourlyEmp.calculatePay();

}

}

OUTPUT :



Q. 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 code :

**package** Lab3;

//Superclass Document

**class** Document {

// Method to open document (to be overridden by subclasses)

**void** open() {

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

}

}

//Subclass WordDocument extending Document

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

// Override open() method for WordDocument

@Override

**void** open() {

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

}

}

//Subclass PDFDocument extending Document

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

// Override open() method for PDFDocument

@Override

**void** open() {

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

}

}

//Subclass SpreadsheetDocument extending Document

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

// Override open() method for SpreadsheetDocument

@Override

**void** open() {

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

}

}

//Main class to demonstrate polymorphism

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

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

// Create instances of different document types

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

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

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

// Demonstrate opening each document type

doc1.open(); // Calls WordDocument's open()

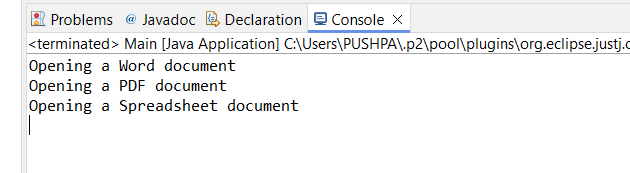
doc2.open(); // Calls PDFDocument's open()

doc3.open(); // Calls SpreadsheetDocument's open()

}

}

OUTPUT :



Q .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 code :

**package** Lab3;

//Calculator class with overloaded add() methods

**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;

}

}

//Main class to demonstrate Calculator class

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

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

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

// Call and display results of each add() method

**int** sum1 = calc.add(5, 10);

**double** sum2 = calc.add(3.5, 2.5);

**int** sum3 = calc.add(7, 8, 9);

System.***out***.println("Sum of integers (5 + 10): " + sum1);

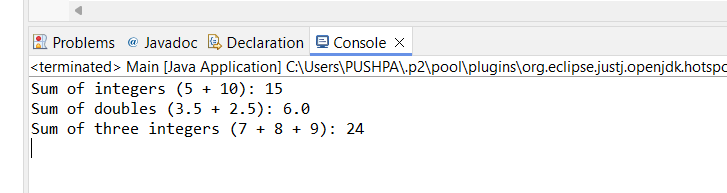
System.***out***.println("Sum of doubles (3.5 + 2.5): " + sum2);

System.***out***.println("Sum of three integers (7 + 8 + 9): " + sum3);

}

}

OUTPUT :



Q. 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 code :

**package** Lab3;

//JavaBean class Person

**class** Person {

**private** String firstName;

**private** String lastName;

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

**private** String email;

// Required no-argument constructor

**public** Person() {

}

// Getter and Setter methods for firstName

**public** String getFirstName() {

**return** firstName;

}

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

**this**.firstName = firstName;

}

// Getter and Setter methods for lastName

**public** String getLastName() {

**return** lastName;

}

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

**this**.lastName = lastName;

}

// Getter and Setter methods for age

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

**return** age;

}

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

**this**.age = age;

}

// Getter and Setter methods for email

**public** String getEmail() {

**return** email;

}

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

**this**.email = email;

}

}

//Main class to demonstrate Person class

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

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

// Create an instance of Person

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

// Set properties of Person

person.setFirstName("kajal");

person.setLastName("prasad");

person.setAge(21);

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

// Print out the properties of Person

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

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

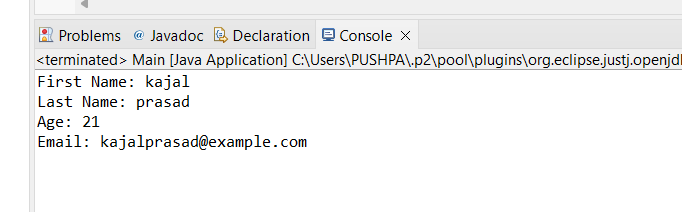
System.***out***.println("Age: " + person.getAge());

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

}

}

OUTPUT:



Q.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 code :

**package** Lab3;

//JavaBean class Car

**class** Car {

**private** String make;

**private** String model;

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

**private** String color;

// Required no-argument constructor

**public** Car() {

}

// Getter and Setter methods for make

**public** String getMake() {

**return** make;

}

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

**this**.make = make;

}

// Getter and Setter methods for model

**public** String getModel() {

**return** model;

}

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

**this**.model = model;

}

// Getter and Setter methods for year

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

**return** year;

}

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

**this**.year = year;

}

// Getter and Setter methods for color

**public** String getColor() {

**return** color;

}

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

**this**.color = color;

}

}

//Main class to demonstrate Car class

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

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

// Create an instance of Car

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

// Set properties of Car

car.setMake("Toyota");

car.setModel("Camry");

car.setYear(2022);

car.setColor("Silver");

// Print out the details of the car

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

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

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

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

}

}

OUTPUT :

