**Lab3**

**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** lab;

//Superclass Person with attributes name and age

**class** Person {

// Attributes

**private** String name;

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

// Constructor

**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);

}

}

**package** lab;

//Subclass Student that inherits from Person

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

// Additional attribute

**private** **int** studentID;

// Constructor

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

// Calling superclass constructor

**super**(name, age);

// Assigning studentID

**this**.studentID = studentID;

}

// Overriding display method to include studentID

@Override

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

// Calling superclass display method

**super**.display();

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

}

}

**package** lab;

//Main class to demonstrate the creation of Student object and display its attributes

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

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

// Creating a Student object

Student student = **new** Student("Saylee", 20, 12345);

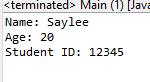
// Displaying all attributes of the Student

student.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** lab;

//Superclass Calculator with a method add(int a, int b)

**class** Calculator {

// Method to add two integers

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

**return** a + b;

}

}

**package** lab;

//Subclass AdvancedCalculator that inherits from Calculator

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

// Overloading the add method to handle three integers

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

**return** a + b + c;

}

}

**package** lab;

//Main class to demonstrate the use of the Calculator classes

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

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

// Creating an instance of AdvancedCalculator

AdvancedCalculator advancedCalculator = **new** AdvancedCalculator();

// Using the add method of Calculator to add two integers

**int** result1 = advancedCalculator.add(5, 3);

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

// Using the overloaded add method of AdvancedCalculator to add three integers

**int** result2 = advancedCalculator.add(5, 3, 2);

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

}

}

**Output:**

**A screenshot of a computer

Description automatically generated**

**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** lab;

//Superclass Vehicle with a method move()

**class** Vehicle {

// Method to move

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

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

}

}

**package** lab;

//Subclass Car that inherits from Vehicle

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

// Additional attributes and methods specific to Car can be added here

}

**package** lab;

//Subclass Bike that inherits from Vehicle

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

// Additional attributes and methods specific to Bike can be added here

}

**package** lab;

//Main class to demonstrate the use of Vehicle, Car, and Bike classes

**public** **class** Main2{

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

// Creating objects of Car and Bike

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

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

// Calling the move method on Car object

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

car.move();

// Calling the move method on Bike object

System.***out***.println("\nBike:");

bike.move();

}

}

**Output:**

**A screenshot of a computer

Description automatically generated**

**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** lab;

//Superclass Employee with an abstract method calculatePay()

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

// Abstract method to calculate pay

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

}

**package** lab;

//Subclass SalariedEmployee that inherits from Employee

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

**private** **double** salary; // Monthly salary

// Constructor

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

**this**.salary = salary;

}

// Implementing the calculatePay method for SalariedEmployee

@Override

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

**return** salary;

}

}

**package** lab;

//Subclass HourlyEmployee that inherits from Employee

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

**private** **double** hourlyRate; // Hourly rate

**private** **int** hoursWorked; // Hours worked

// Constructor

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

**this**.hourlyRate = hourlyRate;

**this**.hoursWorked = hoursWorked;

}

// Implementing the calculatePay method for HourlyEmployee

@Override

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

**return** hourlyRate \* hoursWorked;

}

}

**package** lab;

//Main class to demonstrate the use of SalariedEmployee and HourlyEmployee classes

**public** **class** Main3{

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

// Creating objects of both subclasses

SalariedEmployee salariedEmployee = **new** SalariedEmployee(3000);

HourlyEmployee hourlyEmployee = **new** HourlyEmployee(15, 40);

// Calling the calculatePay method on objects

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

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

}

}

**Output:**

**A screenshot of a computer

Description automatically generated**

**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** lab;

//Superclass Document with a method open()

**class** Document {

// Method to open a document

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

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

}

}

**package** lab;

//Subclass WordDocument that extends Document

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

// Implementing open method for Word documents

@Override

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

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

}

}

**package** lab;

//Subclass PDFDocument that extends Document

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

// Implementing open method for PDF documents

@Override

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

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

}

}

**package** lab;

//Subclass SpreadsheetDocument that extends Document

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

// Implementing open method for Spreadsheet documents

@Override

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

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

}

}

**package** lab;

//Main class to demonstrate compile-time polymorphism

**public** **class** Main4{

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

// Creating instances of different types of documents

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

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

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

// Demonstrating opening different types of documents using compile-time polymorphism

doc1.open(); // Opens a Word document

doc2.open(); // Opens a PDF document

doc3.open(); // Opens a Spreadsheet document

}

}

**Output:**

**A screenshot of a computer

Description automatically generated**

**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** lab;

//Calculator class with overloaded add() methods

**public** **class** Calculator1{

// 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;

}

}

**package** lab;

// Main class to demonstrate the usage of Calculator methods

**public** **class** Main5{

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

// Creating an instance of Calculator

Calculator1 calculator = **new** Calculator1();

// Adding two integers

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

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

// Adding two doubles

**double** sumDouble = calculator.add(5.5, 3.7);

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

// Adding three integers

**int** sumThreeInt = calculator.add(5, 3, 2);

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

}

}

**Output:**

**A screenshot of a computer

Description automatically generated**

**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** lab;

**public** **class** Person1{

**private** String firstName;

**private** String lastName;

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

**private** String email;

// No-argument constructor

**public** Person1() {

}

// 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;

}

}

**package** lab;

**public** **class** Main6{

// Main method to demonstrate usage

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

// Creating an instance of Person

Person1 person = **new** Person1();

// Setting properties

person.setFirstName("Saylee");

person.setLastName("Pawar");

person.setAge(23);

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

// Printing out the properties

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

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

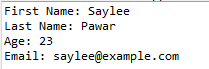
System.***out***.println("Age: " + person.getAge());

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

}

}

**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** lab;

//Car JavaBean class with properties make, model, year, and color

**public** **class** Car1{

**private** String make;

**private** String model;

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

**private** String color;

// No-argument constructor

**public** Car1() {

}

// 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;

}

}

**package** lab;

//Main class to demonstrate the usage of Car class

**public** **class** Main7{

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

// Creating an instance of Car

Car1 car = **new** Car1();

// Setting properties

car.setMake("Toyota");

car.setModel("Camry");

car.setYear(2021);

car.setColor("Red");

// Printing out the car details

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

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:**

**A screenshot of a computer

Description automatically generated**