**LAB 3**

**Solve below questions:**

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** Lab3;

// Superclass Person

**class** Person {

// Person class attributes

String name = "Ashi";

**int** age = 20;

// Method to display the attributes in Person class

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

System.***out***.println("The name of the person is " + name + " and the age is " + age);

}

}

// Subclass Student

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

// Additional attribute of Student class

**int** studentID = 1;

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

// Student class object

Student s1 = **new** Student();

// Printing attributes

System.***out***.println("Student details: ");

System.***out***.println("Name : " + s1.name);

System.***out***.println("Age : " + s1.age);

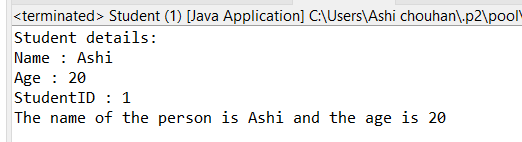
System.***out***.println("StudentID : " + s1.studentID);

s1.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** Lab3;

// Superclass calculator

**class** Calculator {

// Method to add 2 integers a and b

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

**return** a+b;

}

}

// Subclass of Calculator class

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

// Overriding add(int a, int b) method from superclass to add 3 integers

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

**return** a+b+c ;

}

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

**int** a = 2;

**int** b = 3;

**int** c = 5;

// object Of AdvancedCalculator class

AdvancedCalculator ad = **new** AdvancedCalculator();

// Printing output

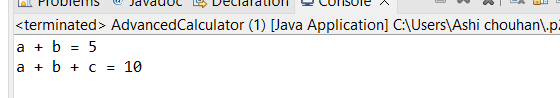
System.***out***.println("a + b = " + ad.add(a, b));

System.***out***.println("a + b + c = " + ad.add(a, b, c));

}

}

**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** nDemo;

// Superclass Vehicle

**class** Vehicle {

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

System.***out***.println("The vehicle is moving !!!");

}

}

// Subclass of Vehicle class

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

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

System.***out***.println("The car is moving !!!");

}

}

//Subclass of Vehicle class

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

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

System.***out***.println("The Bike is moving !!!");

}

}

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

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

// Object of Vehicle class

Vehicle v1 = **new** Vehicle();

// Object of Car class

Car c1 = **new** Car();

// Object of Bike class

Bike b1 = **new** Bike();

// Calling move() method on each

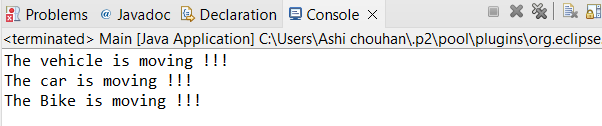
v1.move();

c1.move();

b1.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** Lab3;

//Abstract class Employee

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

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

}

//Subclass of Employee

**class** SalarizedEmployee **extends** Employee {

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

// Parameterized constructor

**public** SalarizedEmployee(**double** salary) {

**this**.salary = salary;

}

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

**return** salary;

}

}

// Subclass of Employee

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

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

**private** **int** hours;

// Constructor

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

**this**.hourlyRate = hourlyRate;

**this**.hours = hoursWorked;

}

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

**return** hourlyRate \* hours;

}

}

**public** **class** EmployeePay {

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

// Creating objects of SalarizedEmployee and HourlyEmployee

SalarizedEmployee sE = **new** SalarizedEmployee(5000);

HourlyEmployee hE = **new** HourlyEmployee(25.0, 160);

// To call calculatePay by both subclass object

**double** salariedPay = sE.calculatePay();

System.***out***.println("Salaried Employee Pay: Rs " + salariedPay);

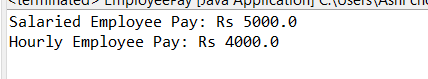
**double** hourlyPay = hE.calculatePay();

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

}

}

**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** Lab3;

// Superclass Document

**class** Document {

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

System.***out***.println("Opening Document !!!");

}

}

// Subclass of Document class

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

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

System.***out***.println("Opening Word Document !!!");

}

}

//Subclass of Document class

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

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

System.***out***.println("Opening PDF Document !!!");

}

}

//Subclass of Document class

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

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

System.***out***.println("Opening Spreadsheet Document !!!");

}

}

**public** **class** DocumentMain {

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

// Creating objects of all subclasses

WordDocument WD = **new** WordDocument();

PDFDocument PDF = **new** PDFDocument();

SpreadsheetDocument SD = **new** SpreadsheetDocument();

// Calling open() method using all objects

WD.open();

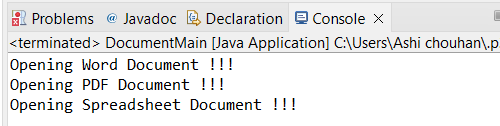
PDF.open();

SD.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** Lab3;

**class** Calculator2 {

// 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** CalculatorMain {

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

// Object of Calculator2 class

Calculator2 c1 = **new** Calculator2();

// Function call and printing outputs

System.***out***.println("2 + 3 = " + c1.add(2, 3));

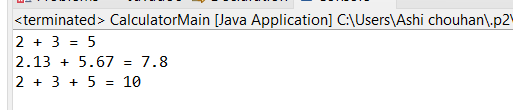
System.***out***.println("2.13 + 5.67 = " + c1.add(2.13, 5.67));

System.***out***.println("2 + 3 + 5 = " + c1.add(2, 3, 5));

}

}

**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.

**OUTPUT:**

**package** Lab3;

**import** java.io.Serializable;

**class** Person1 **implements** Serializable {

// Private attribute declaration

**private** String firstName;

**private** String lastName;

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

**private** String email;

Person1() {

// Default constructor

}

// Getter setter Method for firstName

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

**this**.firstName = firstName;

}

**public** String getFName() {

**return** firstName;

}

// Getter setter Method for lastName

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

**this**.lastName = lastName;

}

**public** String getLName() {

**return** lastName;

}

// Getter setter Method for age

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

**this**.age = age;

}

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

**return** age;

}

// Getter setter Method for email

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

**this**.email = email;

}

**public** String getEmail() {

**return** email;

}

}

**public** **class** PersonMain {

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

// Object of Person1 class

Person1 p1 = **new** Person1();

// Setting values of all attributes

p1.setFName("Ashi");

p1.setLName("Chouhan");

p1.setAge(20);

p1.setEmail("chouhancrusador@gmail.com");

// Retrieving or getting all attributes

System.***out***.println("First Name : " + p1.getFName());

System.***out***.println("Last Name : " + p1.getLName());

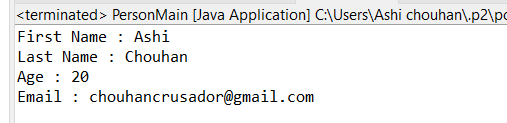
System.***out***.println("Age : " + p1.getAge());

System.***out***.println("Email : " + p1.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** Lab3;

**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;

}

}

**public** **class** CarMain {

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

// Object of Car1 class

Car1 c1 = **new** Car1();

// Set properties using setter methods

c1.setMake("Toyota");

c1.setModel("Camry");

c1.setYear(2022);

c1.setColor("Silver");

// Getting or retrieving all properties

System.***out***.println("\nIndividual Properties:");

System.***out***.println("Make: " + c1.getMake());

System.***out***.println("Model: " + c1.getModel());

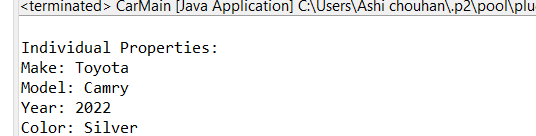
System.***out***.println("Year: " + c1.getYear());

System.***out***.println("Color: " + c1.getColor());

}

}

**OUTPUT:**

****