**LAB 3**

**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.**

**package** Session;

//Superclass Person

**class** Person {

**int** age ;

String name;

**public** **void** Display()

{

System.***out***.println("age is "+age);

System.***out***.println("name is "+name);

}

}

//Subclass Student extending Person

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

{

**int** studentid;

// Constructor for Student class

**public** Student (**int** age,**int** id,String name)

{

**this**.age=age;

**this**.studentid=id;

**this**.name=name;

}

**public** **void** Display()

{

**super**.Display();

System.***out***.println("id is "+ studentid);

}

}

**public** **class** InherLab {

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

// Creating an object of Student

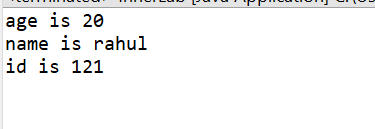
Person s = **new** Student(20,121,"rahul");

s.Display();// Calling the Display method of Student

}

}

**Output :-**

****

**2. 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.**

**package** Session;

//Superclass Vehicle

**class** Vehicle{

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

{

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

}

}

//Subclass Car extending Vehicle

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

**public** **void** go()

{

System.***out***.println("car is going");

}

}

//Subclass Bike extending Vehicle

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

**public** **void** stop()

{

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

}

}

**public** **class** InherVechile {

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

// Create instances of Vehicle, Car, and Bike

Vehicle v = **new** Vehicle();

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

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

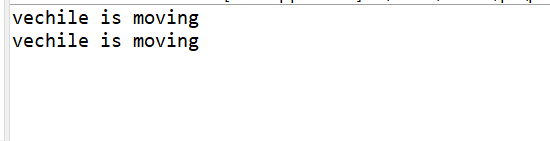
car.move();//calling move() method

bike.move();//calling move() method

}

}

**Output :-**

****

**3. 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.**

**package** Session;

//Abstract superclass Employee

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

// Abstract method to calculate pay()

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

}

//Subclass SalariedEmployee extending Employee

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

{

**public** **void** calculatePay() {

System.***out***.println("paying SalariedEmployee");

}

}

//Subclass HourlyEmployee extending Employee

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

{

**public** **void** calculatePay() {

System.***out***.println("paying HourlyEmployee");

}

}

**public** **class** InherSalary {

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

// Create objects of both subclasses and call calculatePay()

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

Employee sal2=**new** HourlyEmployee();

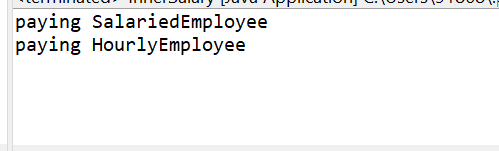
sal.calculatePay();// Calls calculatePay() method of SalariedEmployee

sal2.calculatePay();// Calls calculatePay() method of HourlyEmployee

}

}

**Output :-**

****

**4. 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).**

**package** Session;

// super class document

**class** Document {

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

{

System.***out***.println("opening a document");

}

}

//Subclass WordDocument extending Document

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

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

{

System.***out***.println("opening a Worddocument");

}

}

//Subclass PDFDocument extending Document

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

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

{

System.***out***.println("opening a PDFdocument");

}

}

//Subclass SpreadsheetDocument extending Document

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

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

{

System.***out***.println("opening a SpreadsheetDocument ");

}

}

**public** **class** InherDocument {

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

// Creating objects of different document type

Document d = **new** Document();

Document d1 = **new** WordDocument();

Document d2 = **new** PDFDocument();

Document d3 = **new** SpreadsheetDocument();

// Demonstrating compile-time polymorphism through method overriding

d.open();// Calls Document's open() method

d1.open();// Calls WordDocument's open() method

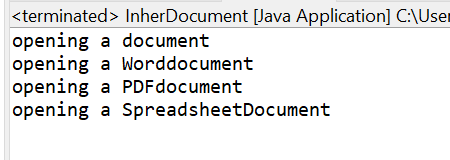
d2.open();// Calls pdfDocument's open() method

d3.open();// Calls SpreadsheetDocument's open() method

}

}

**Output :-**

****

**5. 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**.

**package** Session;

//Calculator class with overloaded methods

**class** Calc{

// Method to add two integers and print the result

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

{

**int** sum= a+b;

System.***out***.println(sum +" = add");

}

// Method to add two double and print the result

**public** **void** doubleadd(**double** a, **double** b)

{

**double** sum= a+b;

System.***out***.println(sum +" = doubleadd");

}

// Method to add three integers and print the result

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

{

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

System.***out***.println(sum +" = 3add");

}

}

**public** **class** CalcOverload {

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

Calc c = **new** Calc();

c.add(2, 3);// Calls add(int, int)

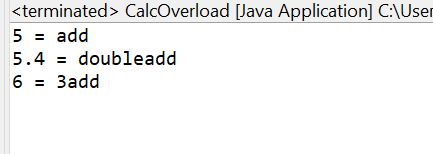
c.doubleadd(2.2, 3.2);// Calls doubleAdd(double, double)

c.add(1, 2, 3);// Calls add(int, int, int)

}

}

**Output :-**

****

**6. 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.**

**package** Session;

**class** Per {

**private** String firstName;

**private** String lastName;

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

**private** String email;

// Required no-argument constructor

**public** Per() {

}

// 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;

}

}

**public** **class** GetSetBean {

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

Per person = **new** Per();

// Set properties using setter methods

person.setFirstName("Rahul");

person.setLastName("yadav");

person.setAge(21);

person.setEmail("rahulyaduvanshi2001.com");

// Print out the properties using getter methods

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

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

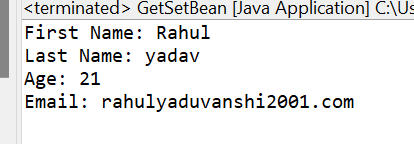
System.***out***.println("Age: " + person.getAge());

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

}

}

**Output :-**

****

**7. 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.**

**package** Session;

**class** Ca {

**private** **int** make;

**private** String model;

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

**private** String colour;

// Required no-argument constructor

**public** Ca() {

}

// Getter and Setter methods for Make

**public** **int** getMake() {

**return** make;

}

**public** **void** setMake(**int** 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 colour

**public** String getcolour() {

**return** colour;

}

**public** **void** setcolour(String colour) {

**this**.colour = colour;

}

}

**public** **class** GetSetBean {

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

Ca n = **new** Ca();

// Set properties using setter methods

n.setMake(2001);

n.setmodel("Toyota");

n.setyear(2012);

n.setcolour("blue");

// Print out the properties using getter methods

System.***out***.println("make: " + n.getMake());

System.***out***.println("model: " + n.getmodel());

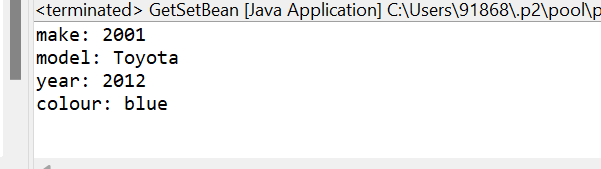
System.***out***.println("year: " + n.getyear());

System.***out***.println("colour: " + n.getcolour());

}

}

**Output :-**

****

**8. 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.**

**package** Session;

// Superclass Calculator

**class** Calcul {

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

**int** sum = a + b;

System.***out***.println("Sum : " + sum);

}

}

// Subclass AdvancedCalculator extending Calculator

**class** AdvancedCalculator **extends** Calcul {

// Overloaded method to add three integers

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

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

System.***out***.println("Sum of advance : " + sum);

}

}

**public** **class** InherlabCalc {

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

// Create instances of Calculator and AdvancedCalculator

AdvancedCalculator adCalc = **new** AdvancedCalculator();

// Call methods on AdvancedCalculator instance

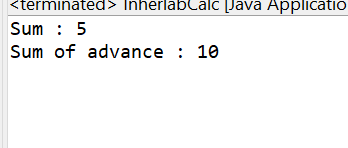
adCalc.add(2, 3); // Calls add(int a, int b) from Calcul

adCalc.add(2, 3, 5); // Calls add(int a, int b, int c) from AdvancedCalculator

}

}

**Output :-**

****