**Candidate Id: 2380155**

**Name : Bharath Magesh**

**Assignment No : 03**

**Lab Exercise No:1**

**Exercise Objective(s): The concept of inheritance**

**Exercise: Create a class called Vehicle. Create subclasses like Truck, Bus, Car etc. Add common methods in the base class and specific methods in the corresponding class. Create a class called Road and create objects for the Truck, Car, Bus etc and display the appropriate message.**

**Answer:**

**package** assignment.day3.proj;

//Base class **class** Vehicle {

String name;

// Method to set the name **void** setName(String vehicleName) { name = vehicleName;

}

// Common method

**void** displayInfo() {

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

}

}

//Subclass Truck

**class** Truck **extends** Vehicle { // Specific method for Truck **void** loadCargo() {

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

}

}

//Subclass Bus **class** Bus **extends** Vehicle { // Specific method for Bus **void** pickPassengers() {

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

}

}

//Subclass Car **class** Car **extends** Vehicle { // Specific method for Car **void** driveFast() {

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

}

}

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

// Create objects for Truck, Bus, and Car

Truck truck = **new** Truck();

truck.setName("Truck"); truck.displayInfo(); truck.loadCargo();

Bus bus = **new** Bus(); bus.setName("Bus"); bus.displayInfo(); bus.pickPassengers();

Car car = **new** Car(); car.setName("Car"); car.displayInfo(); car.driveFast();

}

}

**Output:**

**This is a Truck**

**Truck is loading cargo.**

**This is a Bus**

**Bus is picking up passengers.**

**This is a Car**

**Car is driving fast.**

**Lab Exercise No:2**

**Write a Java program to Implement single inheritance Answer: package** assignment.day3.proj;

//Base class **class** Animal { **void** eat() {

System.***out***.println("This animal eats food.");

}

}

//Derived class **class** Dog **extends** Animal { **void** bark() {

System.***out***.println("The dog barks.");

}

}

//Main class to test single inheritance **public** **class** SingleInheritanceDemo { **public** **static** **void** main(String[] args) { Dog dog = **new** Dog(); dog.eat(); // Inherited method dog.bark(); // Method of Dog class

}

}

**Output:**

**This animal eats food.**

**The dog barks.**

**Lab Exercise No:3**

**Write a Java program to based on the multilevel inheritance in Java Answer:**

**package** assignment.day3.proj;

//Base class **class** Animaln { **void** eat() {

System.***out***.println("This animal eats food.");

}

}

//Derived class **class** Dogn **extends** Animaln { **void** bark() {

System.***out***.println("The dog barks.");

}

}

//Further derived class **class** Puppy **extends** Dogn { **void** weep() {

System.***out***.println("The puppy weeps.");

}

}

//Main class to test multilevel inheritance **public** **class** MultilevelInheritanceDemo { **public** **static** **void** main(String[] args) { Puppy puppy = **new** Puppy(); puppy.eat(); // Inherited from Animal class puppy.bark(); // Inherited from Dog class puppy.weep(); // Method of Puppy class

}

}

**Output:**

**This animal eats food.**

**The dog barks.**

**The puppy weeps.**

**Lab Exercise No:4**

**Create a class named 'Member' having the following members:**

**Data members**

1. **- Name**
2. **- Age**
3. **- Phone number**
4. **- Address 5 - Salary**

**It also has a method named 'printSalary' which prints the salary of the members.**

**Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same. Answer:**

**package** assignment.day3.proj;

//Base class Member **class** Member { String name; **int** age;

String phoneNumber; String address; **double** salary;

// Method to print the salary **void** printSalary() {

System.***out***.println("Salary: " + salary);

}

// Method to display member details **void** displayDetails() {

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

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

System.***out***.println("Phone Number: " + phoneNumber); System.***out***.println("Address: " + address); printSalary();

}

}

//Subclass Employee **class** Employee **extends** Member {

String specialization;

// Method to display Employee details **void** displayEmployeeDetails() {

System.***out***.println("Specialization: " + specialization); displayDetails();

}

}

//Subclass Manager **class** Manager **extends** Member {

String department;

// Method to display Manager details **void** displayManagerDetails() {

System.***out***.println("Department: " + department); displayDetails();

}

}

//Main class **public** **class** Main { **public** **static** **void** main(String[] args) { // Creating Employee object Employee employee = **new** Employee(); employee.name = "John";

employee.age = 30; employee.phoneNumber = "1234567890"; employee.address = "123 Street, City"; employee.salary = 50000; employee.specialization = "Software Development";

// Display Employee details System.***out***.println("Employee Details:"); employee.displayEmployeeDetails();

// Creating Manager object Manager manager = **new** Manager(); manager.name = "Alice"; manager.age = 40; manager.phoneNumber = "0987654321"; manager.address = "456 Avenue, City"; manager.salary = 80000; manager.department = "Human Resources";

// Display Manager details

System.***out***.println("\nManager Details:"); manager.displayManagerDetails();

}

}

**Output:**

**Employee Details:**

**Specialization: Software Development**

**Name: John**

**Age: 30**

**Phone Number: 1234567890**

**Address: 123 Street, City**

**Salary: 50000.0**

**Manager Details:**

**Department: Human Resources**

**Name: Alice**

**Age: 40**

**Phone Number: 0987654321**

**Address: 456 Avenue, City**

**Salary: 80000.0**