by Kunal Sir

INHERITANCE Assignment

---

1. SINGLE INHERITANCE (Parent → Child)

Q1. Vehicle → Car Management System

A transport company maintains basic vehicle information.
`Vehicle` has: registration number, manufacturer, base price.
`Car` adds: fuel type and number of seats.

Task:
Create classes `Vehicle` and `Car`. The Car class should print complete details including on-road price = base price + ₹15,000 road tax.

✓ Sample Input

```
Car
MH12AB1234
Tata Motors
550000
Petrol
5
```

✓ Sample Output

```
Registration: MH12AB1234
Manufacturer: Tata Motors
Base Price: 550000
Fuel Type: Petrol
Seats: 5
On-Road Price: 565000
```

Q2. BankAccount → SavingsAccount

`BankAccount`: account number, holder name, balance
`SavingsAccount`: interestRate, method applyInterest()

Apply interest:
```
newBalance = balance + (balance * rate / 100)
```

✓ Sample Input
```
SavingsAccount
Account No: 12345
Name: Rohan
Balance: 10000
Interest Rate: 5
```
✓ Sample Output
```
Before Interest: 10000
After Interest: 10500
```

Q3. Employee → Manager Salary Calculation

`Employee`: id, name, baseSalary
`Manager`: teamAllowance, overridden calculateSalary()

✓ Sample Input
```
Manager
ID: 101
Name: Meera
Base Salary: 50000
Team Allowance: 8000
```
✓ Sample Output
```
Employee ID: 101
Name: Meera
Final Salary: 58000
```

---

by Kunal Sir

Q4. Product → ElectronicProduct Billing

`Product`: brand, price
`ElectronicProduct`: warrantyYears, finalPrice = price + warrantyYears*500

✓ Sample Input

```
ElectronicProduct
Brand: Samsung
Price: 20000
Warranty: 2 years
```

✓ Sample Output

```
Brand: Samsung
Base Price: 20000
Warranty Years: 2
Final Price: 21000
```

---

2. MULTILEVEL INHERITANCE (Grandparent → Parent → Child)

Q1. Person → Employee → Manager

`Person`: name, age
`Employee`: empId, department
`Manager`: team size

✓ Sample Input

```
Manager
Name: Aarav
Age: 32
Emp ID: E102
Department: IT
Team Size: 12
```

✓ Sample Output

```
Name: Aarav
Age: 32
Employee ID: E102
Department: IT
Team Size: 12
```

Q2. Animal → Mammal → Dog

`Animal`: eat(), sleep()
`Mammal`: warm-blooded info
`Dog`: bark()

✓ Sample Input

```
Dog
Action: Details
```

✓ Sample Output

```
Animal: Eats food
Animal: Sleeps
Mammal: Warm-blooded creature
Dog: Barks loudly
```

Q3. Device → Computer → Laptop

`Device`: serialNo
`Computer`: processor, RAM
`Laptop`: batteryBackup

✓ Sample Input

```
Laptop
Serial: D1001
Processor: i5
RAM: 8GB
Battery: 6 Hours
```

✓ Sample Output

```
Serial No: D1001
Processor: i5
RAM: 8GB
Battery Backup: 6 Hours
```

Q4. Course → OnlineCourse → SelfPacedCourse

`Course`: title, duration
`OnlineCourse`: platform
`SelfPacedCourse`: access validity

✓ Sample Input

```
SelfPacedCourse
Title: Java Basics
Duration: 30 Days
Platform: Udemy
Access Validity: 1 Year
```

✓ Sample Output

```
Course: Java Basics
Duration: 30 Days
Platform: Udemy
Access Validity: 1 Year
```

---

3. HIERARCHICAL INHERITANCE (One Parent → Many Children)

Q1. Shape → Circle, Rectangle, Triangle

`Shape`: color
Each child writes its own draw() method to show shape-specific details.

✓ Sample Input

```
Circle
Color: Red
Radius: 5
```

✓ Sample Output

```
Drawing Circle
Color: Red
Radius: 5
Area: 78.5
```

Q2. Vehicle → Car, Bike, Truck

`Vehicle`: brand
`Car`: seats
`Bike`: mileage
`Truck`: loadCapacity

✓ Sample Input

```
Truck
Brand: Ashok Leyland
Load Capacity: 12000 kg
```

✓ Sample Output

```
Vehicle Type: Truck
Brand: Ashok Leyland
Load Capacity: 12000 kg
```

Q3. Media → Book, Movie, Song

`Media`: title, year
`Book, Movie, Song`: Child adds its own fields.

✓ Sample Input

```
Movie
Title: Inception
Year: 2010
Director: Christopher Nolan
```

✓ Sample Output

```
Media Type: Movie
Title: Inception
Year: 2010
Director: Christopher Nolan
```

Q4. Payment → UPI, CreditCard, Cash

`Payment`: amount
UPI, CreditCard, Cash: Each child class writes its own processPayment() method according to its payment type.

✓ Sample Input

```
UPI
Amount: 250
UPI ID: rohan@oksbi
```

✓ Sample Output

```
Processing UPI Payment
Amount: 250
Paid via: rohan@oksbi
Payment Successful
```

Q1. University Management System (Multilevel + Hierarchical)

A university needs a class structure to manage different roles of people.

Person (base class)

- name, age

Employee extends Person (single inheritance)

- employeeId, department

TeachingStaff extends Employee (multilevel inheritance)

- subject, timetable

NonTeachingStaff extends Employee (hierarchical inheritance)

- duty, shift

Student extends Person (hierarchical inheritance)

- rollNo, course

Problem:

Design the inheritance structure so that:

- Person → Employee → TeachingStaff forms **multilevel inheritance**
- Employee → TeachingStaff and Employee → NonTeachingStaff form **hierarchical inheritance**
- Person → Student is **single inheritance**

Create a method **getDetails()** in every class and show how inheritance reduces code duplication.

Q2. Hospital Management (Single + Multilevel + Hierarchical)

A hospital wants to classify its working structure.

HospitalMember

- id, name

Doctor extends HospitalMember (single inheritance)

- specialization

Surgeon extends Doctor (multilevel inheritance)

- surgeryType

Nurse extends HospitalMember (hierarchical inheritance)

- wardAssigned

Receptionist extends HospitalMember (hierarchical inheritance)

- deskNumber

Problem:

Build the above hierarchy where:

- HospitalMember → Doctor → Surgeon forms a **multilevel** chain
- HospitalMember → Nurse and HospitalMember → Receptionist form **hierarchical inheritance**

Demonstrate method overriding of **performDuty()** for each role.

Q3. Automobile Manufacturing (Single + Hierarchical + Multilevel)

An automobile company needs to model its vehicle categories.

Vehicle

- brand, model

Car extends Vehicle (single inheritance)

- numberOfDoors

SUV extends Car (multilevel inheritance)

- groundClearance

Truck extends Vehicle (hierarchical inheritance)

- loadCapacity

SportsCar extends Car (hierarchical inheritance)

- topSpeed

Problem:

Create the class model such that:

- Vehicle → Car → SUV forms **multilevel inheritance**
- Vehicle → Truck and Vehicle → Car form **hierarchical inheritance**
- Car extends Vehicle is **single inheritance**

Implement a method **vehicleInfo()** in all classes to show specialization increasing at each level.

Q4. E-Commerce Product Catalog (Multilevel + Hierarchical + Single)

An e-commerce platform wants to organize products.

Product

- id, name

Electronics extends Product (single inheritance)

- warranty

MobilePhone extends Electronics (multilevel inheritance)

- cameraQuality

Laptop extends Electronics (hierarchical inheritance)

- processor

Clothing extends Product (hierarchical inheritance)

- size, fabric

Problem:

Model the platform so that:

- Product → Electronics → MobilePhone forms **multilevel inheritance**
- Product → Electronics and Product → Clothing form **hierarchical inheritance**
- Electronics extends Product is **single inheritance**

Create a method `display()` in every class.
Each subclass should show its own extra information along with the details it inherits from its parent class.
