Day – 2

Revisiting the basics of Java

- **Inheritance**
- **Method overloading**

Note: Prepare the theory concept first, then implement the below requirement

*****AI is injurious to Developers*****
This task is not for LAZY BUGS

Note: You are supposed to use only concepts such as constructors, Inheritance, constructor overloading, method ooverloading(if req).

Requirement 1

Enhance your movie booking system by categorizing movies into:

- **Now Showing**
- **Coming Soon**

Create a Movie class to store and display movie info by using displayMovieInfo().

- **Add the below class inside the Movie class.**
  - **name - String**
  - **language - String**
  - **genre - double**
  - **rating - Strong**

Create a class NowShowing which extends Movie class. Override displayMovieInfo()

- **Add the below class inside the Movie class.**
  - **showTime - String**

Create a class ComingSoon which extends Movie class. Override displayMovieInfo()

- **Add the below class inside the Movie class.**
  - **releaseDate - String**

Parent Class : Movie
Child Class : NowShowing, ComingSoon

Method to override in child classes : displayMovieInfo().

Expected Output format

NowShowing

```
Movies in Theatre
-----------------
Movie: KGF
Language: Kannada
Genre: Action
Rating: 4.9/5
Show Time: 10:00AM
```

ComingSoon

```
Upcoming Movies
-----------------
Movie: Fantasic4
Language: English
Genre: Action
Rating: NA
Release Date: 25-July-2025
```

Requirement 2

Enhance your ShoppingCart class by categorizing products into:

- **ElectronicProducts**
- **ClothProducts**

Create a ShoppingCart class to store and display products info stored in cart by using displayItems().

- **Add the below class inside the ShoppingCart class.**
  - **name – String**
  - **quantity - int**
  - **price - double**
  - **total - double**

Create a class ElectronicProducts which extends ShoppingCart class. Override displayItems()

- **Add the below class inside the ElectronicProducts class.**
  - **warrenty - String**

Create a class ClothProducts which extends ShoppingCart class. Override displayItems()

- **Add the below class inside the ClothProduct class.**
  - **size - String**

Parent Class : ShoppingCart
Child Class : ElectronicProducts, ClothProducts
Method to override in child classes : displayItems().

Expected Output format

ElectronicProducts

```
Product: iPhone
Quantity: 1
Price: 150000
Total: 150000
Warranty: 4 years
```

ClothProducts

```
Product: Shirt
Quantity: 2
Price: 1000
Total: 2000
Size: Medium
```

Scenario:
A user may add an item by:

- **Only name**
- **Only name and quantity (price fetched from DB)**

Expected Output format

ClothProduct

```
Product: Shoes
Size: 8
```

ElectronicProduct

```
Product: Headphones
Quantity: 3
Warranty: 1 year
```

Requirement 3

Enhance your Employee class by categorizing them into:

- **DayShift**
- **NightShift**

Create a Employee class to store and display employee info by using displayEmplInfo().

- **Add the below class inside the Movie class.**
  - **name – String**
  - **age - int**
  - **job - String**
  - **salary - double**

Create a class NightShift which extends Employee class. Override displayEmplInfo()

- **Add the below class inside the Employee class.**
  - **timing – String**
  - **night-allowance – double [extra amount paid for night shift emps]**

Create a class DayShift which extends Employee class. Override displayEmplInfo()

- **Add the below class inside the Employee class.**
  - **timing – String**

Create a class Senior which extends DayShift class. Override displayEmplInfo()

- **Add the below class inside the Employee class.**
  - **experience – int [in years]**

Create a class Junior which extends DayShift class. Override displayEmplInfo()

- **Add the below class inside the Employee class.**
  - **manager – String [need to display the manager name from any of the DayShift employee data]**

Parent Class : Employee
Child Class : NightShift, DayShift
Child Class for DayShift : Senior, Junior
Method to override in child classes : displayEmplInfo().

Expected Output format

NightShift

```
Name: Roy
Age: 40
Job: Manager
Salary: 40000
Timing: 6 PM to 3 AM
Night-Allowance: 3000
```

DayShift

```
Name: Kumar
Age: 32
Job: Developer
Salary: 45000
Timing: 9 AM to 6 PM
```

Senior

```
Name: Kumar
Age: 32
Job: Developer
Salary: 45000
Timing: 9 AM to 6 PM
Experience [in years]: 7
```

Junior

```
Name: Raghav
Age: 28
Job: Developer
Salary: 35000
Timing: 9 AM to 6 PM
Manager: Kumar
```