

# Abstract Classes & Interfaces

---

## 1. Shape Drawing System (Abstract Base Class)

- Abstract class Shape:
    - Abstract method double area().
    - Concrete method void printArea() → prints name + area.
  - Subclasses:
    - Circle (field: radius).
    - Rectangle (fields: length, width).
  - **Main Task:**
    - Create a Shape[ ] with different shapes and loop to print areas polymorphically.
- 

## 2. Payment System (Interface)

- Interface Payment:
    - Method String pay(double amount).
  - Implementations:
    - CreditCardPayment: add 2% fee.
    - PayPalPayment: add 3% + 10 flat fee.
    - CashPayment: no extra fee.
  - **Main Task:**
    - Store different Payment objects in an array, call pay(1000) on each, and print receipts.
-

### 3. Animal Hierarchy (Abstract + Interface)

- Abstract class Animal:
    - Abstract method String sound().
    - Abstract method String habitat().
  - Interface Pet:
    - Method void play().
  - Subclasses:
    - Dog extends Animal implements Pet.
    - Cat extends Animal implements Pet.
    - Lion extends Animal (not a pet).
  - **Main Task:**
    - Show sounds, habitats, and call play() only on pets.
- 

### 4. Vehicle System (Abstract Parent + Child Abstract Class)

- Abstract class Vehicle:
    - Abstract method double fuelEfficiency() (km/l).
    - Concrete method void printType().
  - Abstract subclass MotorVehicle (adds field: engineCapacity).
  - Concrete subclasses:
    - Car, Bike extend MotorVehicle.
  - **Main Task:**
    - Demonstrate abstract → abstract → concrete inheritance.
    - Print fuel efficiencies for multiple vehicles.
-

## 5. File System (Interface Inheritance)

- Interface Readable: method void `read()`.
  - Interface Writable extends Readable: method void `write(String data)`.
  - Class TextFile implements Writable:
    - Stores a String content.
  - **Main Task:**
    - Write and then read contents of TextFile.
- 

## 6. Employee Hierarchy (Abstract Class + Polymorphism)

- Abstract class Employee:
    - Abstract method double `calculateSalary()`.
    - Field: String name.
  - Subclasses:
    - FullTimeEmployee: salary = base + allowance.
    - PartTimeEmployee: salary = hourlyWage × hours.
  - **Main Task:**
    - Store employees in an array and print names + salaries.
- 

## 7. Sports Interface System (Multiple Interfaces)

- Interface Playable: method void `play()`.
  - Interface Scorable: method int `getScore()`.
  - Class Football implements both.
  - Class Cricket implements both.
  - **Main Task:**
    - Demonstrate multiple interface implementation by tracking play actions and scores.
-

## 8. Appliance Control (Interface + Abstract Class)

- Interface Switchable:
    - void turnOn(), void turnOff().
  - Abstract class Appliance implements Switchable:
    - Field: String brand.
    - Abstract method void run().
  - Subclasses:
    - Fan, WashingMachine.
  - **Main Task:**
    - Turn appliances on, call run(), then turn off.
- 

## 9. Online Media Player (Parent Interface + Implementations)

- Interface Media:
    - Method void play().
  - Interface Downloadable: extends Media:
    - Method void download().
  - Class Music implements Downloadable.
  - Class Video implements Downloadable.
  - **Main Task:**
    - Demonstrate polymorphism via Media[] and Downloadable[].
-

## 10. Robotics Control (Complex Integration)

- Abstract class Robot:
  - Field String id.
  - Abstract method void performTask().
- Interface Rechargeable:
  - Method void recharge().
- Interface Connectable:
  - Method void connect(String network).
- Subclasses:
  - CleaningRobot extends Robot implements Rechargeable, Connectable.
  - DeliveryRobot extends Robot implements Rechargeable.
- Main Task:
  - Create both robots.
  - Make them perform tasks, recharge, and (for CleaningRobot) connect to Wi-Fi.