DAY 7 PRACTICE QUESTIONS

Problem 1: Single Inheritance

Problem Statement:

Create two classes, `Person` and `Employee`. The `Person` class should have properties like `name` and `age`. The `Employee` class should inherit from the `Person` class and have additional properties like `employeeId` and `department`. Write a program to create an instance of `Employee`, set values for all properties, and print them.

Test Case:

- **Input:**

  Set `name` = "Alice", `age` = 30, `employeeId` = "E123", `department` = "HR"

- **Output:**

  ```

  Name: Alice

  Age: 30

  Employee ID: E123

  Department: HR

  ```

Problem 2: Multilevel Inheritance

Problem Statement:

Create three classes: `Animal`, `Mammal`, and `Dog`. The `Animal` class should have properties like `speciesName`. The `Mammal` class should inherit from `Animal` and have an additional property like `hasFur`. The `Dog` class should inherit from `Mammal` and have a property `breed`. Write a program to create an instance of `Dog`, set values for all properties, and print them.

Test Case:

- **Input:**

  Set `speciesName` = "Canine", `hasFur` = true, `breed` = "Golden Retriever"

- **Output:**

  ```

  Species: Canine

  Has Fur: true

  Breed: Golden Retriever

  ```

Problem 3: Hierarchical Inheritance

Problem Statement:

Create a base class `Vehicle` with properties like `make` and `model`. Create two derived classes, `Car` and `Bike`, that inherit from `Vehicle`. The `Car` class should have an additional property `numberOfDoors`, and the `Bike` class should have an additional property `type` (e.g., "Sport", "Cruiser"). Write a program to create instances of both `Car` and `Bike`, set values for all properties, and print them.

Test Case:

- **Input:**

  For `Car`: Set `make` = "Toyota", `model` = "Corolla", `numberOfDoors` = 4

  For `Bike`: Set `make` = "Yamaha", `model` = "R1", `type` = "Sport"

- **Output:**

  ```

  Car Make: Toyota

  Car Model: Corolla

  Number of Doors: 4

  Bike Make: Yamaha

  Bike Model: R1

  Bike Type: Sport

  ```

Problem 4: Multiple Inheritance (Using Abstract Class)

Problem Statement:

Java does not support multiple inheritance directly, but it can be achieved using abstract classes. Create two abstract classes, `Person` and `Employee`, where `Person` has properties `name` and `age`, and `Employee` has properties `employeeId` and `salary`. Create a class `Manager` that inherits from both `Person` and `Employee`. Implement a method in the `Manager` class that prints the manager's details.

Test Case:

- **Input:**

  Set `name` = "Bob", `age` = 45, `employeeId` = "M456", `salary` = 95000.00

- **Output:**

  ```

  Name: Bob

  Age: 45

  Employee ID: M456

  Salary: 95000.0

  ```

Problem 5: Hybrid Inheritance (Using Abstract Classes)

Problem Statement:

Create three abstract classes, `Vehicle`, `FourWheeler`, and `TwoWheeler`. The `Vehicle` class has properties `make` and `model`, `FourWheeler` has a property `numberOfDoors`, and `TwoWheeler` has a property `hasCarrier`. Create two classes `Car` and `Bike` that inherit from `FourWheeler` and `TwoWheeler`, respectively. Write a program to set and print the properties of both `Car` and `Bike`.

Test Case:

- **Input:**

For `Car`: Set `make` = "Honda", `model` = "Civic", `numberOfDoors` = 4

For `Bike`: Set `make` = "Suzuki", `model` = "Hayabusa", `hasCarrier` = false

- **Output:**

```

Make: Honda

Model: Civic

Number of Doors: 4

Make: Suzuki

Model: Hayabusa

Has Carrier: false

```

Problem 6: Using `this`, `super`, and `abstract` Keywords

Problem Statement:

Create a class `Parent` with a constructor that takes a parameter `name` and assigns it to the instance variable using the `this` keyword. Create a class `Child` that inherits from `Parent` and uses the `super` keyword to call the parent constructor. Additionally, create an abstract class `AbstractClass` with a method `display()` that is implemented in the `Child` class. Write a program to create an instance of `Child` and call the `display()` method.

Test Case:

- **Input:**

Set `name` = "Charlie"

- **Output:**

```

Name: Charlie

```