VIGNAN'S NIRULA
INSTITUTE OF TECHNOLOGY & SCIENCE FOR WOMEN
(Approved by AICTE, New Delhi & Affiliated to JNTU Kakinada)
Vignan Avenue, Peda Palakaluru, Guntur - 522009
Department of Information Technology

Innovative Teaching Practice

Faculty Name : Dr. V. Pavani, Mrs.B.Aruna Kumari
Course Name : Java Programming
Class : II B. Tech II Semester
Academic Year : 2021-2022
Title of the Topic : Inheritance and its types
Activity Name : Mind Mapping

Objective

Inheritance in Java allows a class to inherit properties and methods from another class, promoting code reusability and easier maintenance. It supports different types, including single inheritance**,** multilevel inheritance, and hierarchical inheritance

Introduction

Inheritance in Java allows a class to inherit fields and methods from another class, promoting code reusability and a more organized structure. It supports types like single inheritance**,** where a class inherits from one superclass**,** multilevel inheritance**,** where a subclass inherits from another subclass, and hierarchical inheritance, where multiple subclasses inherit from a single superclass.

Mind Map Creation

Key concepts to include:

Inheritance in Java

- **Definition**: Mechanism where one class acquires the properties and behaviors of another class.

Types of Inheritance

1. **Single Inheritance**
   - A subclass inherits from one superclass.
   - Example: class Dog extends Animal
2. **Multilevel Inheritance**
   - A subclass inherits from another subclass, creating a chain.
   - Example: class Puppy extends Dog extends Animal
3. **Hierarchical Inheritance**
   - Multiple subclasses inherit from a single superclass.
   - Example: class Dog extends Animal, class Cat extends Animal
4. **Multiple Inheritance (through Interfaces)**
   - Achieved by implementing more than one interface.
   - Example: class Dog implements Animal, Pet

5. **Hybrid Inheritance**
    - Combination of multiple types of inheritance (class + interface).
    - Example: class Dog extends Mammal implements Animal

Key Concepts

- **Code Reusability**: Subclasses reuse code from their parent class.
- **Method Overriding**: Subclasses can provide specific implementations of superclass methods.
- **Polymorphism**: Subclass objects can be treated as instances of the superclass.

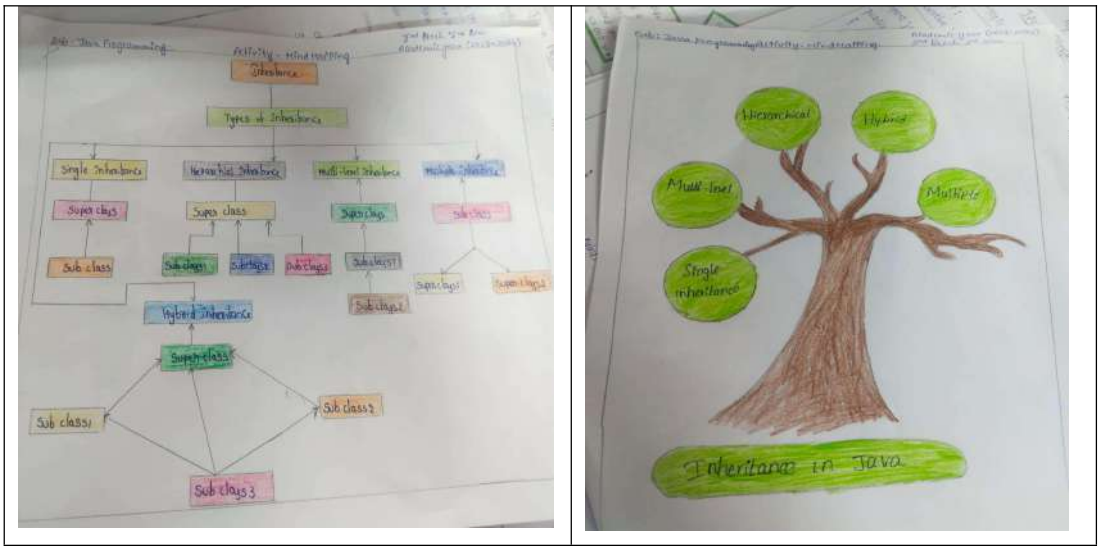
Features of Inheritance

- **Extensibility**: Add or modify functionality without changing the original class.
- **Hierarchical Structure**: Classes can be organized into a clear hierarchy.
- **Object Relationships**: Establishes "is-a" relationships (e.g., Dog is an Animal).

Constructor Inheritance

- **Super Constructor**: A subclass can call the parent class constructor using super().
- **Constructor Chaining**: Inherited constructors can call other constructors within the class or superclass.

Screenshot of the Practice



central Node: Inheritance in Java

Branch 1: Types of Inheritance

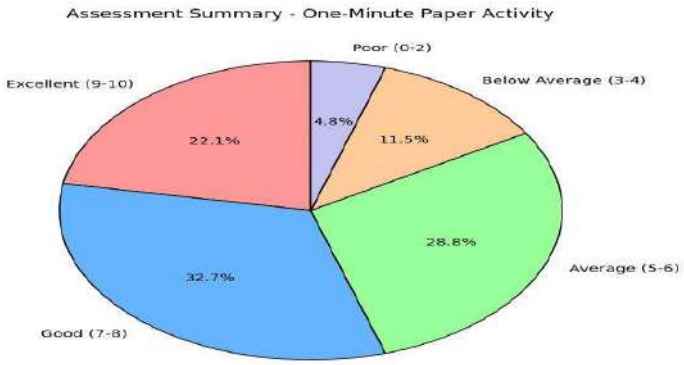
- **Sub-branches:**
    - **Single Inheritance**:
        - One subclass inherits from a single superclass.
        - Example: class Dog extends Animal

- **Multilevel Inheritance**:
  - A subclass inherits from another subclass (creating a chain).
  - Example: class Puppy extends Dog extends Animal
- **Hierarchical Inheritance**:
  - Multiple subclasses inherit from a single superclass.
  - Example: class Dog extends Animal, class Cat extends Animal

- **Multiple Inheritance (via Interfaces)**:
  - Achieved by implementing multiple interfaces (Java does not allow multiple inheritance through classes).
  - Example: class Dog implements Animal, Pet
- **Hybrid Inheritance**:
  - A combination of more than one type of inheritance (e.g., class inheritance and interface inheritance).
  - Example: class Dog extends Mammal implements Animal

Assessment Analysis

Marks Range	Number of Students	Percentage
9-10 (Excellent)	22	21.78%
7-8 (Good)	35	34.65%
5-6 (Average)	30	29.70%
3-4 (Below Avg)	8	7.92%
0-2 (Poor)	6	5.94%
Total	**101**	**100%**



Conclusion:

Inheritance in Java is a fundamental concept that allows classes to share and extend functionality, promoting code reusability and a more organized structure. It supports various types such as single, multilevel, hierarchical, and multiple inheritance (through interfaces), each offering unique ways to structure relationships between classes.

Signature of the Faculty **Head of the Department**