Java – Notes 02

Topic: Methods, Variables & Inheritance

Student Class with Methods

(From page 1 diagram – rollno, name, contact with setValues() and display())

🔹 Structure Explained

The diagram shows:

- Data Members:
  - `rollno`
  - `name`
  - `contact`
- Methods:
  - `setValues(int a, String b, long c)`
  - `display()`

Example Code with Explanation

```
class Student {
    int rollno;
    String name;
    long contact;

    void setValues(int a, String b, long c) {
        rollno = a;
        name = b;
        contact = c;
    }

    void display() {
        System.out.println("Roll No: " + rollno);
```

```
        System.out.println("Name: " + name);
        System.out.println("Contact: " + contact);
    }
}
```

Explanation

- Variables are **instance variables**
- `setValues()` assigns values to instance variables
- `display()` prints values
- This demonstrates **Encapsulation** (data + methods inside class)

Types of Inheritance (From Page 1 Diagrams)

The diagrams clearly show different inheritance structures:

---

1. Single Inheritance

A → B

One parent, one child.

```
class A {
    void show() {
        System.out.println("Class A");
    }
}

class B extends A {
}
```

---

2. Hierarchical Inheritance

```
  A
 / \
B   C
```

One parent, multiple children.

---

3. Multilevel Inheritance

A → B → C

Child becomes parent for another class.

---

4. Multiple Inheritance (❌ Not supported in Java using classes)

```
A   B
 \ /
  C
```

Java does **NOT** support multiple inheritance with classes
(It avoids ambiguity problem – Diamond Problem)

---

5. Hybrid Inheritance

Combination of more than one type
❌ Not supported directly with classes

---

extends Keyword

From page 1 bottom section.

`extends` is used to inherit properties of another class.

class A {

```
    int x;

    void setX(int x1) {
        x = x1;
    }
}

class B extends A {
}
```

Here:

- Class B inherits variable `x`
- Class B inherits method `setX()`

Method Execution & Object Creation (Page 2 Diagram)

The diagram shows:

```
Y y1 = new Y();
y1.setX(10);
y1.setY(20);
y1.displayX();
y1.displayY();
```

🔎 What Happens Internally?

1. Object `y1` is created in memory.
2. Memory allocated for:
    - x = 0 (default)
    - y = 0 (default)
3. After `setX(10)` → x becomes 10
4. After `setY(20)` → y becomes 20
5. `display()` prints values

Example Code

```
class X {
    int x;

    void setX(int x1) {
        x = x1;
    }

    void displayX() {
        System.out.println(x);
    }
}

class Y extends X {
    int y;

    void setY(int y1) {
        y = y1;
    }

    void displayY() {
        System.out.println(y);
    }
}
```

Types of Variables (From Page 2 Bottom Section)

The board clearly mentions three types:

---

1. Local Variable

✔ Defined inside a method or block
✔ Scope limited to that method

```
void method() {
```

```
    int a = 10;   // Local variable
}
```

---

2. Instance Variable (Member Variable)

✔ Defined inside class
✔ Outside method
✔ Each object gets its own copy

```
class Demo {
    int x;   // Instance variable
}
```

---

3. Class Variable (Static Variable)

✔ Defined using `static` keyword
✔ Shared among all objects

```
class Demo {
    static int count;
}
```

Key Differences

Type	Defined Where	Scope	Memory
Local	Inside method	Only method	Stack
Instance	Inside class	Per object	Heap
Static	Inside class (static)	Shared	Method Area

Important Concepts Covered in Session 2

✔ Class & Methods
✔ Object Creation
✔ Memory Allocation
✔ Inheritance Types
✔ extends keyword
✔ Instance vs Local vs Static Variables

Conceptual Understanding Summary

- Inheritance promotes **code reuse**
- `extends` creates an **IS-A relationship**
- Java supports:
    - Single
    - Multilevel
    - Hierarchical
- Java does NOT support:
    - Multiple (via classes)
    - Hybrid (directly)