# Inheritance and Interfaces

Date: 07-10-2025

Write a program to create a base class called Student and two subclasses named MTech and Btech. The Student class contains three attributes: name, dept and CGPA and a method called show(), which displays the values of the attributes. The Mtech class has an additional attribute known as Specialization. The Btech class has an additional attribute known as Placement. Create objects of Mtech and Btech class and display their details.

**Aim:**
To write a Java program that demonstrates inheritance using a base class Student and two subclasses MTech and BTech. Each subclass extends Student by adding an extra attribute and displaying all details using the show() method.

**Algorithm:**

1. Base Class:
   Student — contains attributes: name, dept, cgpa and method show() to display them.
2. Subclass 1:
   MTech — extends Student and adds an attribute specialization.
3. Subclass 2:
   BTech — extends Student and adds an attribute placement.
4. Method Used:
   show() — displays details of both parent and subclass attributes.
5. Main Class:
   Create objects of MTech and BTech classes and call their show() methods to display details.

**Program:**

```
/**
 * Base Class: Student
 * Subclasses: MTech, BTech
 * Demonstrates Inheritance and method overriding in Java.
 */
class Student {
    String name;
    String dept;
    double cgpa;
```

```java
    // Constructor for Student
    Student(String name, String dept, double cgpa) {
        this.name = name;
        this.dept = dept;
        this.cgpa = cgpa;
    }

    // Method to display details
    void show() {
        System.out.println("Name: " + name);
        System.out.println("Department: " + dept);
        System.out.println("CGPA: " + cgpa);
    }
}

// Subclass 1
class MTech extends Student {
    String specialization;

    MTech(String name, String dept, double cgpa, String specialization) {
        super(name, dept, cgpa);
        this.specialization = specialization;
    }

    @Override
    void show() {
        super.show();
        System.out.println("Specialization: " + specialization);
        System.out.println("------------------------------");
    }
}

// Subclass 2
class BTech extends Student {
    String placement;

    BTech(String name, String dept, double cgpa, String placement) {
        super(name, dept, cgpa);
        this.placement = placement;
    }
```

```java
        @Override
        void show() {
            super.show();
            System.out.println("Placement: " + placement);
            System.out.println("-------------------------------");
        }
    }

    // Main Class
    public class InheritanceDemo {
        public static void main(String[] args) {
            // Creating MTech object
            MTech m1 = new MTech("Kishor G", "CSE", 9.0, "Artificial Intelligence");
            // Creating BTech object
            BTech b1 = new BTech("Arun Kumar", "IT", 8.7, "Placed at TCS");

            // Displaying details
            System.out.println("M.Tech Student Details:");
            m1.show();

            System.out.println("B.Tech Student Details:");
            b1.show();
        }
    }
```

**Output:**

M.Tech Student Details:

Name: Kishor G

Department: CSE

CGPA: 9.0

Specialization: Artificial Intelligence

--------------------------------

B.Tech Student Details:

Name: Arun Kumar

Department: IT

CGPA: 8.7

Placement: Placed at TCS

--------------------------------

**Result:**

Thus, the Java program was successfully executed to demonstrate inheritance and method overriding by displaying details of both MTech and BTech students derived from the base class Student.