

1st July 2025 (Tuesday)

This session was dedicated to Object-Oriented Programming (OOP) fundamentals, which form the conceptual bridge between Java and Android. The mentor began with the real-world analogy of how classes and objects represent blueprints and instances. Using relatable examples like a “Car” class and its objects (“Honda”, “Ford”, “Maruti”), I learned how object-oriented principles help organize code efficiently.

We coded a Student class that contained attributes like name, roll number, and marks, and implemented methods to calculate grades. I also learned about constructors — their types (default and parameterized) — and how they initialize object data automatically. The instructor emphasized encapsulation and data hiding, showing how private access modifiers protect data from unauthorized access. We also used public getter and setter methods to access private variables safely.

Later, we briefly discussed the this keyword, static members, and method overloading. The mentor demonstrated how overloading methods allow us to define multiple versions of a method with different parameter types or counts. I wrote several small programs to solidify the concept. In the final few minutes, the instructor linked these principles to Android, explaining how Activities, Fragments, and Views in Android are actually classes and objects that communicate using event-driven methods. This helped me visualize how Java’s OOP structure naturally extends into Android’s architecture.