2nd July 2025 (Wednesday)

Building upon the previous class, today’s focus was on Inheritance, Polymorphism, and Abstraction. The instructor began with the concept of code reusability through inheritance, demonstrating how one class can extend another to gain access to its properties and methods. I practiced examples involving base and derived classes like Vehicle, Car, and Bike. This made me understand how general-to-specific relationships are implemented in code.

The mentor then explained method overriding, which allows subclasses to redefine parent class methods. We discussed the difference between compile-time (method overloading) and runtime (method overriding) polymorphism, and how they enhance flexibility in code design. I practiced a small example where different types of payment classes (e.g., CreditCardPayment, UPI) override a generic processPayment() method.

After that, the session moved to abstract classes and interfaces, which are essential for achieving abstraction and multiple inheritance. We created examples using abstract keywords and implemented interfaces like Runnable to understand real-world use cases. The instructor concluded by relating these OOP concepts to Android components, where abstraction and interfaces play a key role in handling events like onClick() or background threads. I ended the session with a much clearer picture of how Java’s modular design supports Android’s architecture.