Here are 6 detailed assignments that combine Java Static Polymorphism, Dynamic Polymorphism, and Interface Static & Default Methods, designed to learn concepts by doing.

Assignment 1 – Static Polymorphism (Method Overloading)

Scenario:

You are building a Math Utility Library for an engineering company. They want a single class to handle add() for integers, doubles, and arrays.

Tasks:

1. Create a class MathOperations.
2. Overload the method add() in three ways:

- Add two integers.
- Add two doubles.
- Add an array of integers.

3. Create a main() method to test all overloaded versions.

Expected Learning:

Understand compile-time binding — how the method signature determines which method runs.

Assignment 2 – Dynamic Polymorphism (Method Overriding)

Scenario:

You are creating a Payment Gateway System that supports multiple payment methods.

Tasks:

1. Create a base class Payment with a method processPayment(double amount).
2. Create subclasses CreditCardPayment and UPIPayment that override the method with their own logic.
3. In the main() method, store both payment objects in a Payment reference and process payments.

Expected Learning:

Understand runtime method binding and how the object type (not reference type) decides the method execution.

Assignment 3 – Combining Static & Dynamic Polymorphism

Scenario:

You are working on an Online Shopping System.

There's a need for calculating discounts differently for different customer types.

Tasks:

1. Create a base class Customer with two calculateDiscount() methods:

- One taking double purchaseAmount (overloaded)
- One taking double purchaseAmount and int loyaltyPoints (overloaded)

2. Create two subclasses RegularCustomer and PremiumCustomer that override calculateDiscount(double purchaseAmount).

3. In main(), show both overloading (static polymorphism) and overriding (dynamic polymorphism) in action.

Expected Learning:

See both compile-time and runtime polymorphism in one scenario.

Assignment 4 – Interface Static Method

Scenario:

You are developing a Data Validator for form submissions.

Tasks:

1. Create an interface DataValidator with:

- A static method isNotEmpty(String input) that returns true if input is not null or empty.
- An abstract method isValid(String input).

2. Create classes EmailValidator and PhoneValidator implementing the interface.

3. In main(), call the interface static method directly without creating an object.

Expected Learning:

Understand that interface static methods are utility-like and can only be called using the interface name.

Assignment 5 – Interface Default Method

Scenario:

You are making a Music Player Application that supports multiple formats.

Tasks:

1. Create an interface MusicPlayer with:
   - An abstract method play(String fileName).
   - A default method stop() that prints "Music stopped".
2. Create classes MP3Player and WAVPlayer implementing the interface.
3. In main(), call the default method from different player objects.

Expected Learning:

Understand that default methods allow adding new methods to interfaces without breaking existing implementations.

Assignment 6 – Industry Simulation: All-in-One

Scenario:

You are tasked to build a Banking System simulation.

Requirements:

1. Static Polymorphism:

Create a TransactionProcessor class with overloaded process() methods for:

- process(int accountNumber, double amount)
- process(int fromAccount, int toAccount, double amount)

2. Dynamic Polymorphism:

Create a BankAccount base class with an overridden calculateInterest() method in SavingsAccount and CurrentAccount.

3. Interface Static & Default Methods:

Create an interface SecurityCheck with:

- A static method logAttempt(String user)
- A default method showSecurityStatus() printing "Secure connection established".
- An abstract method verifyUser(String username, String password).

4. Use all concepts in a single main() method.

Expected Learning:

Apply all four concepts (static polymorphism, dynamic polymorphism, interface static methods, interface default methods) in a real-world-like project.