

# Model Paper – IT3030: Programming Applications and Frameworks

---

Duration: 90 Minutes

Total Marks: 80

## Question 1 – Framework Use in Application Development (Total: 25 marks)

1. Define the term “**software framework**” and explain its role in achieving modularity in modern software design. (3 marks)
2. Describe a case where a framework’s default architecture significantly influenced the design decisions in a software project. (6 marks)
3. Compare the use of software frameworks in developing mobile vs. web applications. Highlight three key differences, including any limitations or advantages. (10 marks)
4. Explain how using a framework with built-in dependency injection can impact testing and maintenance. Use a suitable example to justify. (6 marks)

## Question 2 – Version Control Practices with Git (Total: 25 marks)

1. Discuss three reasons why Git is widely adopted in modern software teams over older VCS systems. (5 marks)
2. Assume you are mentoring an intern who is new to Git. Outline the step-by-step process of branching, merging, and resolving a conflict using CLI commands. (8 marks)
3. Identify three good practices when managing a collaborative codebase using Git. Justify how each helps in reducing integration issues. (6 marks)
4. Analyze the implications of misusing “**force push**” in a shared repository. Provide two examples of real-world consequences. (6 marks)

### Question 3 – API Design and System Integration (Total: 30 marks)

Scenario:

You are designing a RESTful API for a Campus Resource Management System. The system includes managing lab schedules, booking equipment, and tracking usage statistics.

1. Define the concept of “**statelessness**” in REST. Illustrate how this constraint would apply when booking lab equipment. (5 marks)

2. Design three RESTful endpoints for the following:

- Fetching the weekly lab schedule for a department
- Booking a specific device for a date/time range
- Cancelling an existing booking

Provide method, URI structure, and a one-line purpose. (6 marks)

3. Your team decides to adopt token-based authentication for securing API endpoints.

- a. Explain how this approach supports REST constraints
- b. Describe a scenario where token expiry would be critical
- c. Suggest how to implement token refresh securely (6 marks)

4. In a design review, a teammate suggests including embedded metadata within every JSON response (e.g., processing time, request ID).

- a. Discuss two advantages of this suggestion
- b. Identify one potential downside
- c. Recommend a suitable format or convention for such metadata (7 marks)

5. A client app developer reports that they cannot distinguish between “**not found**” and “**unauthorized**” errors.

- a. Suggest three HTTP status codes that would resolve this confusion with context-specific use cases
- b. Explain the significance of semantic status codes for frontend-backend coordination (6 marks)