

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

PUBLICATION

# EXAM-TM40L-PRE



---

## PRE-TEST — SL 4L: SOFTWARE ENGINEER

---

*Maven Smart System (MSS) — USAREUR-AF*

HEADQUARTERS  
UNITED STATES ARMY EUROPE AND AFRICA  
(USAREUR-AF)  
Wiesbaden, Germany

DRAFT — NOT FOR OFFICIAL USE. FOR TRAINING PLANNING PURPOSES ONLY.

**26 MARCH 2026**

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

# PRE-TEST — SL 4L: SOFTWARE ENGINEER

## MAVEN SMART SYSTEM (MSS) — USAREUR-AF

Field	Detail
Course	SL 4L: Software Engineer
Form	Pre-Test
Level	SL 4L (Specialist)
Audience	Software engineers; prerequisite: SL 1+20+30 + TypeScript/Python + REST API familiarity
Time Allowed	30 minutes
Passing Score	N/A — diagnostic only

## INSTRUCTIONS

This diagnostic assessment establishes your baseline knowledge before training. Your score does not affect course eligibility. Answer honestly — results help the instructor tailor instruction to gaps.

## SECTION 1 — MULTIPLE CHOICE

Circle the letter of the best answer. (2 points each)

1. In TypeScript, an `async` function returns:

- A. The resolved value of the computation directly
- B. A callback function that must be invoked manually
- C. A `Promise` that resolves to the return value
- D. An `Observable` stream of values

2. "Pagination" in a REST API context means:

- A. Returning large result sets in smaller chunks (pages) with a cursor or offset to retrieve subsequent pages
- B. Breaking a web page into multiple sections for rendering performance
- C. Caching API responses to reduce server load
- D. Versioning API endpoints to maintain backward compatibility

**3. When a REST API call returns a `202 Accepted` status code, it typically means:**

A. The request was completed synchronously and the result is in the response body  
B. The request was received and accepted for processing, but the result is not yet available  
C. The request was rejected due to a validation error  
D. The request requires authentication before it can be processed

**4. The `try/catch/finally` pattern in TypeScript is used for:**

A. Iterating over arrays and handling empty results  
B. Validating function parameters before execution  
C. Structured error handling — catching exceptions, providing fallback logic, and ensuring cleanup code runs  
D. Typing asynchronous function return values

**5. "Rate limiting" in an API context means:**

A. The API enforces a maximum response time for each request  
B. The API restricts the number of requests a client can make in a given time period  
C. The client batches requests to reduce API call overhead  
D. The server limits the size of request payloads

**6. In TypeScript, `interface` declarations are used to:**

A. Define the implementation of a class  
B. Describe the shape (properties and types) of an object without providing implementation  
C. Import external libraries and modules  
D. Define enumerated constant values

**7. A WebSocket connection differs from a standard HTTP request in that:**

A. WebSocket maintains a persistent, bidirectional connection enabling real-time push notifications without polling  
B. WebSocket uses UDP instead of TCP for faster transmission  
C. WebSocket is only available on secure (HTTPS) connections  
D. WebSocket responses are automatically cached by the browser

**8. In a REST API, "input sanitization" before processing user-supplied data is important to prevent:**

A. Data type mismatches that cause TypeScript compilation errors  
B. Performance degradation from large payload sizes  
C. Injection attacks (SQL injection, command injection) where malicious input manipulates system behavior  
D. Unauthorized access to restricted API endpoints

**9. A "webhook" is best described as:**

A. A library for building REST APIs in TypeScript  
B. A type of persistent connection similar to WebSocket  
C. A tool for testing API endpoints during development  
D. An HTTP callback — when an event occurs, the server sends an HTTP POST to a pre-registered URL

**10. "Hardcoded credentials" in source code are a security risk because:**

A. They cause TypeScript compilation errors if the credential format is incorrect  
B. They are exposed to anyone with access to the code repository and cannot be rotated without a code change  
C. They slow down application startup by requiring credential validation on load  
D. They prevent the application from running in multiple environments

**11. In a JavaScript/TypeScript application, `Promise.all([...])` is used to:**

A. Execute an array of promises sequentially, waiting for each to complete before starting the next B. Convert an array of synchronous values into a resolved promise C. Retry a failed promise until it succeeds D. Execute multiple promises concurrently and wait for all of them to resolve (or one to reject)

**12. An API that returns the first 100 records but the full dataset contains 10,000 records requires the client to:**

A. Accept the first 100 records as a representative sample B. Request the full dataset in a single call with a `limit=10000` parameter C. Implement pagination logic to repeatedly fetch subsequent pages until all records are retrieved D. Export the data to a file and process it offline

**13. "Input validation at system boundaries" in a software engineering context means:**

A. Checking and sanitizing all external inputs (user input, API responses, file data) before processing them in the application B. Validating the database schema before any query is executed C. Running unit tests on all functions that accept input parameters D. Validating TypeScript type annotations at runtime

**14. In a client-server architecture, "state management" in a frontend application refers to:**

A. Managing and updating the application's in-memory representation of data as user interactions and API responses change it B. The persistence of data in the server-side database C. The storage of user session tokens on the client D. Synchronizing application state across multiple browser tabs

**15. A "code review" before merging a pull request serves which primary purpose?**

A. Verifying that the code compiles and all tests pass B. Ensuring the developer followed the correct branching workflow C. Documenting the changes made in the commit history D. A human review of logic correctness, security, standards compliance, and maintainability before changes reach production

---

## SECTION 2 — SHORT ANSWER

*Answer in 2–5 sentences. (6 points each)*

**SA-1. Explain the difference between synchronous and asynchronous programming in TypeScript. Give an example of when you would use `async/await` instead of a synchronous call, and why.**

**SA-2. Describe what an "N+1 query problem" is in a data access context. Give an example in a TypeScript application and explain why it is a performance anti-pattern.**

**SA-3. Explain what a "REST API" is and describe the four standard HTTP methods (GET, POST, PUT/PATCH, DELETE) and what each is typically used for.**

**SA-4. Describe why environment variables are the correct way to manage application credentials and configuration in a production software system, instead of hardcoding values in source code.**

**SA-5. You are performing a security code review of a TypeScript application. Describe four specific security issues you would look for in the code and explain why each is a risk.**

## SCORING SUMMARY

Section	Questions	Points Each	Total Points
Multiple Choice	15	2	30
Short Answer	5	6	30
<b>Total</b>	—	—	<b>60</b>

Passing: N/A — Pre-test is diagnostic only.

## ANSWER KEY — INSTRUCTOR USE ONLY

*Do not distribute to students.*

**Multiple Choice:** 1. C — `async` functions return a Promise that resolves to the return value. 2. A — Pagination returns large result sets in smaller pages with a cursor/offset. 3. B — 202 Accepted = request received for async processing, result not yet available. 4. C — try/catch/finally is for structured error handling with cleanup. 5. B — Rate limiting restricts the number of requests per time period. 6. B — TypeScript `interface` describes object shape without providing implementation. 7. A — WebSocket maintains persistent bidirectional connection for real-time communication. 8. C — Input sanitization prevents injection attacks. 9. D — Webhook = HTTP callback triggered by server-side event. 10. B — Hardcoded credentials are exposed in repos and cannot be rotated without code changes. 11. D — Promise.all executes concurrently and waits for all promises to resolve. 12. C — Pagination logic must be implemented to retrieve all pages. 13. A — Input validation at system boundaries checks all external inputs before processing. 14. A — State management maintains the application's in-memory data representation. 15. D — Code review verifies logic, security, standards compliance, and maintainability.

### Short Answer Guidance:

SA-1. Full credit: synchronous = sequential execution, blocking — one operation completes before the next starts; asynchronous = non-blocking — operations can run concurrently, with results handled via callbacks, promises, or async/await; use async/await when calling an external API or database that has latency — blocking the thread waiting for a network response would freeze the application; example:

`const data = await fetch(url)` instead of blocking the event loop. Partial credit (3 pts) for correct definition of one type without an example.

SA-2. Full credit: N+1 problem = one query to get a list of N records, then N additional queries (one per record) to get related data — total queries = N+1; example: fetch 100 vehicles, then for each vehicle make a separate API call to get its maintenance records = 101 queries instead of 1 or 2; anti-pattern because it causes O(N) API calls, degrading performance significantly as N grows; fix: batch-fetch all maintenance records for the vehicle list in a single query. Partial credit (3 pts) for correct definition without an example.

SA-3. Full credit: REST API = Representational State Transfer — a standard architecture for web APIs using HTTP; GET = retrieve data (read, no side effects); POST = create a new resource; PUT/PATCH = update an existing resource (PUT replaces entirely, PATCH updates partially); DELETE = remove a resource. All four methods correctly described for full credit.

SA-4. Full credit: environment variables externalize configuration from code — credentials can be rotated without code changes; the source code can be shared publicly or reviewed without exposing secrets; different environments (dev/test/prod) use different values without code changes; hardcoded values in

source code are exposed to anyone with repository access and appear in git history even after removal. Partial credit (3 pts) for two of four reasons.

SA-5. Full credit: any four from — hardcoded credentials or API keys; unsanitized user input (injection risk); unvalidated API response data used directly without type/schema check; missing error handling (unhandled promise rejection); secrets in log output; insecure HTTP instead of HTTPS; no rate limiting on outbound API calls; missing access control check before privileged operations. Each issue must include a brief risk explanation for full credit.

---

*USAREUR-AF Operational Data Team TM-40L Pre-Test | Version 1.0 | March 2026*

DRAFT