UNIVERSITY COMPUTER SCIENCE • EXERCISE SHEET

# 12.5 Testing and continuous integration

Name: _____ Class: _____ Date: _____ **Total: 24 marks**

**KEY WORDS** test pyramid

## Objective

Build the pyramid, read coverage honestly, and run the red-green loop.

## 1 Key knowledge and worked examples

■ **The levels**

Many unit tests, fewer integration tests, few end-to-end tests; CI runs them per push; red blocks ✓.

■ **Worked example: coverage**

92% line, 81% branch – the branch number is the honest one; the gate sits at 85% line ✓.

## 2 Practice

**2.1** Why do unit tests form the pyramid's base? [3]

**2.2** Line 92% but branch 81% – what does the gap mean? [3]

**2.3** What is a test without an assertion? [3]

## 3 Exam-style questions

**3.1** The renewal feature (FR-4) lands next sprint.

Design its tests at all three levels: name the unit cases (dates included), the integration case, and the end-to-end case. State which level catches a leap-February bug fastest and why. State the CI order and where the coverage gate sits. [8]

**3.2** A PR arrives with 93% line coverage on the new module and one test whose body is empty (no assertions).

Classify the coverage claim honestly. State what the empty test contributes and what it should contain. Give the red-green loop for the reservation-blocks-renewal rule from a failing test to merge. [7]

## Solutions

**2.1**

- Fast and isolated: they run constantly and pinpoint failures ✓.

**2.2**

- Decisions never exercised: each untested branch is an untested behaviour ✓.

**2.3**

- Coverage theatre – a number without a verdict ✓.

**3.1**

- Unit: due extension arithmetic – renew from D gives $D+14$, twice gives $D+28$, third refused; cases with leap February and year boundary dates ✓.
- Integration: the renewed due date persists and reads back ✓.
- End-to-end: a scripted member renews and the screen shows the new due date ✓.
- Leap-February: the unit level – pure date arithmetic, runs in milliseconds, no setup ✓.
- CI order: install → unit → integration → the 85% coverage gate → merge; red at any stage blocks ✓.

**3.2**

- 93% line with an empty test: line coverage counts executed lines, so the claim is true and worthless – the behaviour is unverified ✓.
- The empty test contributes nothing but coverage; it must assert the observable outcome (the due date, the refusal) ✓.
- Loop: write the failing test (reservation present, renew refused) → push, CI red → implement the reservation check → CI green, coverage rises → merge unblocked ✓.