# **Lean pre-coding sheet — 78/89**

**Component:** tests/determinism.rs (same-OS & cross-OS reproducibility)

## **1) Goal & success**

**Goal:** Prove byte-identical **Result** and **RunRecord** on repeat (same OS) and across **Windows/macOS/Linux**, per **VM-TST-019/020**.

**Success:** Matching SHA-256 for Result and RunRecord; RNG seed recorded if used; time/memory within published profile for the large synthetic.

## **2) Scope**

**In:** Canonical serialization (UTF-8, sorted JSON keys, LF, UTC), stable ordering, integer/rational comparisons, RNG seeding and logging, offline I/O.

**Out:** Algorithm math correctness (covered by other tests), report formatting.

## **3) Inputs → outputs**

**Inputs:** Annex B Part 7 fixtures:  
 – **VM-TST-019** generator (large synthetic; fixed seed, pop baselines) for same-OS runs.  
 – **VM-TST-020** (small baseline from VM-TST-001; optional rng\_seed=424242) for cross-OS.

**Outputs (asserted):** sha256(Result), sha256(RunRecord) equal across runs/OS; artifacts end with single LF; zero float occurrences.

## **4) Entities/Tables (minimal)**

## **5) Variables (used here)**

## **6) Functions (test signatures only)**

rust

CopyEdit

#[test] fn vm\_tst\_019\_same\_os\_repeat\_hashes\_identical();

#[test] fn vm\_tst\_020\_cross\_os\_hashes\_identical();

#[test] fn canonical\_json\_sorted\_keys\_lf\_utc();

#[test] fn no\_floats\_anywhere\_in\_artifacts();

(Names mirror 6C acceptance.)

## **7) Test logic (bullet outline)**

**VM-TST-019 (same OS):** Run baseline twice with fixed generator seed; compute SHA-256 over **canonical** bytes; expect identical Result/RunRecord hashes and runtime within profile.

**VM-TST-020 (cross-OS):** Run small baseline; optionally set rng\_seed=424242; compare hashes across OS (CI job aggregates).

**Canonicalization checks:** assert sorted keys, LF line ending, UTC timestamps; stable unit/option order (Unit ID; order\_index).

**RunRecord echo:** confirm FID, EngineVersion, input IDs, RNG seed present.

## **8) State flow (very short)**

Normal pipeline to **BUILD\_RESULT → BUILD\_RUN\_RECORD**; determinism rules binding: same inputs + same engine ⇒ identical outputs.

## **9) Determinism & numeric rules**

Stable ordering (Units by ID; Options by order\_index); integers/rational comparisons; **round-half-even** only at defined points; RNG only with rng\_seed; no OS RNG/time; canonical JSON hashing.

## **10) Edge cases & failure policy**

Any diff in hashes → dump canonical strings, check key order/LF/UTC and input path ordering; if ties present without seed or with OS RNG, fail and report missing rng\_seed.

## **11) Test checklist (must pass)**

**019:** identical hashes on repeat; perf ≤ profile.

**020:** identical hashes across OS; seed logged if used.