Lab 13

Thomas Koetting

December 2025

1. **Is there a fault? Locate & explain.**

   Yes. The for loop starts at index 1 instead of 0, so it ignores the first lecture's attendance. It should start from `i = 0`, otherwise absences in `attendance_records[0]` are never counted.

2. **Test case that does not execute the fault.**

   To not execute the fault, the loop must never run. The problem states that there are 10 lectures, and the loop will always execute for `i = 1...9`. So for a 10-element attendance record, the fault is always executed, so that test case is impossible.

3. **Test case that executes the fault but does not result in an error state.**

   I need the faulty line to run and the absence count to still be correct. This will happen if the first lecture is present, so skipping it doesn't change the count.

   $$attendance\_records = [1, 0, 0, 0, 1, 1, 1, 1, 1, 1]$$

   Correct behavior: absences = 3 ⇒ student fails.
   Faulty behavior: absences = 3 ⇒ student fails.
   The fault is executed, but the count is still correct, so there is no error state and no failure.

4. **Test case that results in an error state but not a failure.**

   Now the absence count needs to be wrong for the error state but still produce the correct result. This happens if the first lecture is absent, so skipping it reduces the count by one, but the threshold is still crossed.

   Example:

   $$attendance\_records = [0, 0, 0, 0, 1, 1, 1, 1, 1, 1]$$

   Correct behavior: absences = 4 ⇒ student fails.
   Faulty behavior: absences = 3 ⇒ student fails.
   The fault is executed and the absence count is wrong (should be 4 not 3), so we have an error state, but the final result is still the same, so there is ”technically” no failure.

5. **Test case that results in a failure.**

   Now I need the wrong absence count to change the final result. Since the faulty code under counts by exactly 1 when the first lecture is absent, we choose a case where the true absence count is 3 (so the student should fail), but the faulty count becomes 2 (and the student incorrectly passes).

   Example:

   $$attendance\_records = [0, 0, 0, 1, 1, 1, 1, 1, 1, 1]$$

   Correct behavior: absences = 3 ⇒ student should fail (expected output: `true`).
   Faulty behavior (skip index 0): absences = 2 ⇒ program returns `false` (passes).
   The fault is executed, it is wrong, and the output differs from the specification, so this test causes a failure.