

Sample answers: Designing a basic test plan

Below are two example test cases – one basic, one detailed. How does your test case compare?

Example 1

1. Test planning:

- a. Objectives: Check if the calculator works.
- b. Scope: Test basic operations and buttons.
- c. Approach: Test manually.

2. Test case design:

- a. Positive test cases:
 - i. Test adding, subtracting, multiplying and dividing.
- b. Negative test cases:
 - i. Test division by zero.

3. Test execution:

- a. Manual execution: Test the calculator and compare results to expected outcomes.

4. Test evaluation:

- a. Defect analysis: Look at any problems found and decide if they need to be fixed before releasing the app.

Example 2

1. Test planning:

- a. Objectives: Verify that the calculator application functions correctly and has a user- friendly interface.
- b. Scope: Test arithmetic operations, user interface elements, error handling, and edge cases.
- c. Approach: Perform manual testing using positive and negative test cases.

2. Test case design:

- a. Positive test cases:
 - i. Test valid input combinations for addition, subtraction, multiplication and division.
 - ii. Test decimal calculations and rounding behaviour.
 - iii. Test the "Clear" button functionality.
- b. Negative test cases:
 - i. Test invalid input combinations, such as multiple decimal points or consecutive operators.
 - ii. Test division by zero scenario and proper error message display.
 - iii. Test input limit for the display field.

3. Test execution:

- a. Manual execution: Manually enter test inputs, perform operations and compare the actual output to the expected output from test cases.
- b. Defect documentation: Record any discrepancies between the actual and expected outputs as defects, along with relevant details such as input values, expected output, actual output and steps to reproduce the issue.

4. Test evaluation:

- a. Defect analysis: Review the defects found during test execution and assess their severity and impact on the overall quality of the calculator application.
- b. Decision making: Determine if the identified defects need to be fixed before the software can be released or if they can be addressed in a future update.