

# Chapter 1

Q1). What is software testing? State objectives of software testing. (4 Mark, S-16) (Definition - 1 mark; any 3 objectives - 3 marks) )

Ans : Software testing is a method of assessing the functionality of a software program. OR Software testing is the process of validating and verifying that a software program or application or product, meets the business and technical requirements that guided its design and development. Objectives of software testing.

1. Finding defects which may get created by the programmer while developing the software.
2. Gaining confidence in and providing information about the level of quality.
3. To prevent defects.
4. To make sure that the end result meets the business and user requirements.
5. To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
6. To gain the confidence of the customers by providing them a quality product.

Q2). State any four testing principles. (4 Mark, S-16) (Any 4 principles - 1 mark each)

Ans : Testing Principles:

- Testing shows presence of defects: Testing can show the defects are present, but cannot prove that there are no defects. Even after testing the application or product thoroughly we cannot say that the product is 100% defect free.
- Exhaustive testing is impossible: Testing everything including all combinations of inputs and preconditions is not possible. So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.
- Early testing: In the software development life cycle testing activities should start as early as possible and should be focused on defined objectives.
- Defect clustering: A small number of modules contains most of the defects discovered during prerelease testing or shows the most operational failures.
- Pesticide paradox: If the same kinds of tests are repeated again and again, eventually the same set of test cases will no longer be able to find any new bugs. To overcome this Pesticide Paradox, it is really very important to review the test cases regularly and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects Testing is basically
- Context dependent : Different kinds of sites are tested differently. For example, safety critical software is tested differently from an e-commerce site.

- Absence of errors fallacy: If the system built is unusable and does not fulfill the user's needs and expectations then finding and fixing defects does not help

Q3). Explain the terms Mistake, Error, Defect, Bug, Fault and Failure in relation with software testing. (6 Mark, S-16) (Description of mistake; error; defect; bug; fault and failure - 6 marks)

Ans: The various terms related to software failure with respect to the area of application are listed as Defect, Variance, Fault, Failure, Problem, Inconsistency, Error, Feature, Incident, Bug, and Anomaly.

- Failure: the inability of a system or component to perform its required functions within specified performance requirements.
- Fault: An incorrect step, process, or data definition in a computer program.
- Error: A human action that produces an incorrect result. An error can be a grammatical error in one or more of the code lines, or a logical error in carrying out one or more of the client's requirements. Not all software errors become software faults. In some cases, the software error can cause improper functioning of the software. In many other cases, erroneous code lines will not affect the functionality of the software as a whole. A failure is said to occur whenever the external behavior of a system does not conform to that prescribed in the system specification. A software fault becomes a software failure only when it is —activated For example `#include void main() { int i , fact, n; printf (—enter the number —); scanf(—%d||,&n); for(i =1 ;i <=n;i++) fact = fact * i; printf (—the factorial of a number is ||%d||, fact); }` As in line number 4 the fact is not initialized to 1, so it takes garbage value and gives a wrong output, this is an example of a bug. If fact is initialized to zero (fact = 0) than the output will be zero as anything multiplied by zero will give the output as zero. This is a bug which can be removed by initializing fact = 1 during initializing. As the fact is declared as integer, for the number till 7! will work perfectly. When the number entered is 8, the output is garbage value as the integer limit is from – 32767 to +32768, so in declaration change the initialization to long int fact

Q4). What is entry and exit criteria of software testing? (4 Mark, S-15) (Entry criteria-2Marks, Exit criteria-2Marks) State the Entry and Exit criteria's for the software testing. (4 Mark, S-16) (Entry criteria - 2 marks; exit criteria - 2 marks)

Ans : When to Start and Stop Testing of Software (Entry and Exit Criteria) Process model is a way to represent any given phase of software development that prevent and minimize the delay between defect injection and defect detection/correction. Entry Criteria , specifies when that phase can be started also included the inputs — for the phase. Tasks or steps that need to be carried out in that phase, along with measurements that characterize the tasks. Verification , which specifies methods of checking that tasks have been carried out correctly. Clear entry criteria make sure that a given phase does not start prematurely. The verification for each phase helps to prevent defects. At least defects can be minimized. Exit Criteria, which stipulate the conditions under which one can consider the phases — as done and included are the outputs for the phase.

Exit criteria may include:

- All test plans have been run
- All requirements coverage has been achieved.
- All severe bugs are resolved.

Q5). What is test case? Which parameters are to be considered while documenting test cases?

(Explanation of test case- 2 Marks, parameters-2 Marks) (4 Mark, W-15) What is a 'test case'? State its specification parameter. (4 Mark, S-16) (Definition- 1 mark; Parameters - 3 marks)

Ans : Test case is a well-documented procedure designed to test the functionality of the feature in the system. For designing the test case, it needs to provide set of inputs and its corresponding expected outputs.

- Parameters: Test case ID: is the identification number given to each test case.
- Purpose: defines why the case is being designed.
- Precondition: for running in the system can be defined, if required, in the test case.
- Input: should not hypothetical. Actual inputs must be provided, instead of general inputs.  
Expected outputs which should be produced when there is no failure