[ISTQB Foundation Sample Question Paper No. 3](http://istqbexamcertification.com/)

**1.Software testing activities should start**

1. as soon as the code is written
2. during the design stage
3. when the requirements have been formally documented
4. as soon as possible in the development life cycle
5. **.Faults found by users are due to:**
6. Poor quality software
7. Poor software and poor testing
8. bad luck
9. insufficient time for testing
10. **.What is the main reason for testing software before releasing it?**
11. to show that system will work after release
12. to decide when the software is of sufficient quality to release
13. to find as many bugs as possible before release
14. to give information for a risk based decision about release
15. **. which of the following statements is not true**
16. performance testing can be done during unit testing as well as during the testing of whole system
17. The acceptance test does not necessarily include a regression test
18. Verification activities should not involve testers (reviews, inspections etc)
19. Test environments should be as similar to production environments as possible
20. **. When reporting faults found to developers, testers should be:**
21. as polite, constructive and helpful as possible
22. firm about insisting that a bug is not a “feature” if it should be fixed
23. diplomatic, sensitive to the way they may react to criticism
24. All of the above
    1. **n which order should tests be run?**
25. the most important tests first
26. the most difficult tests first(to allow maximum time for fixing)
27. the easiest tests first(to give initial confidence)
28. the order they are thought of
29. **The later in the development life cycle a fault is discovered, the more expensive it is to fix. why?**
30. the documentation is poor, so it takes longer to find out what the software is doing.
31. wages are rising
32. the fault has been built into more documentation,code,tests, etc
33. none of the above
34. **Which is not true-The black box tester**
35. should be able to understand a functional specification or requirements document
36. should be able to understand the source code.
37. is highly motivated to find faults
38. is creative to find the system’s weaknesses
39. **A test design technique is**
40. a process for selecting test cases
41. a process for determining expected outputs
42. a way to measure the quality of software
43. a way to measure in a test plan what has to be done
44. **Testware(test cases, test dataset)**
45. needs configuration management just like requirements, design and code
46. should be newly constructed for each new version of the software
47. is needed only until the software is released into production or use
48. does not need to be documented and commented, as it does not form part of the released software system
49. **An incident logging system**

a only records defects

b is of limited value

c is a valuable source of project information during testing if it contains all incidents

d. should be used only by the test team.

1. **Increasing the quality of the software, by better development methods, will affect the time needed for testing (the test phases) by:**
2. reducing test time
3. no change
4. increasing test time
5. can’t say
6. **Coverage measurement**
7. is nothing to do with testing
8. is a partial measure of test thoroughness
9. branch coverage should be mandatory for all software
10. can only be applied at unit or module testing, not at system testing
11. **When should you stop testing?**
12. when time for testing has run out.
13. when all planned tests have been run
14. when the test completion criteria have been met
15. when no faults have been found by the tests run
16. **Which of the following is true?**
17. Component testing should be black box, system testing should be white box.
18. if u find a lot of bugs in testing, you should not be very confident about the quality of software
19. the fewer bugs you find,the better your testing was
20. the more tests you run, the more bugs you will find.
21. **What is the important criterion in deciding what testing technique to use?**
22. how well you know a particular technique
23. the objective of the test
24. how appropriate the technique is for testing the application
25. whether there is a tool to support the technique
26. **If the pseudocode below were a programming language ,how many tests are required to achieve 100% statement coverage?**
27. If x=3 then
28. Display\_messageX;
29. If y=2 then
30. Display\_messageY;
31. Else
32. Display\_messageZ;
33. Else
34. Display\_messageZ;
35. 1
36. 2
37. 3
38. 4
39. **Using the same code example as question 17,how many tests are required to achieve 100% branch/decision coverage?**
40. 1
41. 2
42. 3
43. 4
44. **Which of the following is NOT a type of non-functional test?**
45. State-Transition
46. Usability
47. Performance
48. Security
49. **Which of the following tools would you use to detect a memory leak?**
50. State analysis
51. Coverage analysis
52. Dynamic analysis
53. Memory analysis
54. **Which of the following is NOT a standard related to testing?**
55. IEEE829
56. IEEE610
57. BS7925-1
58. BS7925-2

**22.which of the following is the component test standard?**

1. IEEE 829
2. IEEE 610
3. BS7925-1
4. BS7925-2
5. **which of the following statements are true?**
6. Faults in program specifications are the most expensive to fix.
7. Faults in code are the most expensive to fix.
8. Faults in requirements are the most expensive to fix
9. Faults in designs are the most expensive to fix.
10. **Which of the following is not the integration strategy?**
11. Design based
12. Big-bang
13. Bottom-up
14. Top-down
15. **Which of the following is a black box design technique?**
16. statement testing
17. equivalence partitioning
18. error- guessing
19. usability testing
20. **A program with high cyclometic complexity is almost likely to be:**
21. Large
22. Small
23. Difficult to write
24. Difficult to test
25. **Which of the following is a static test?**
26. code inspection
27. coverage analysis
28. usability assessment
29. installation test
30. **Which of the following is the odd one out?**
31. white box
32. glass box
33. structural
34. functional
35. **A program validates a numeric field as follows:**

values less than 10 are rejected, values between 10 and 21 are accepted, values greater than or equal to 22 are rejected

which of the following input values cover all of the equivalence partitions?

1. 10,11,21
2. 3,20,21
3. 3,10,22

d. 10,21,22

1. **Using the same specifications as question 29, which of the following covers the MOST boundary values?**

a. 9,10,11,22

b. 9,10,21,22

c. 10,11,21,22

d. 10,11,20,21

1. **An input field takes the year of birth between 1900 and 2004. The boundary values for testing this field are**

a. 0,1900,2004,2005

b. 1900, 2004

c. 1899,1900,2004,2005

d. 1899, 1900, 1901,2003,2004,2005

1. **Which one of the following are non-functional testing methods?**
2. System testing
3. Usability testing
4. Performance testing
5. Both b & c
6. **Which of the following tools would be involved in the automation of regression test?**
7. Data tester
8. Boundary tester
9. Capture/Playback
10. Output comparator.
11. **Incorrect form of Logic coverage is:**
12. Statement Coverage
13. Pole Coverage
14. Condition Coverage
15. Path Coverage
16. **Which of the following is not a quality characteristic listed in ISO 9126 Standard?**
17. Functionality
18. Usability
19. Supportability
20. Maintainability
21. **To test a function, the programmer has to write a , which calls the function to be**

**tested and passes it test data.**

1. Stub
2. Driver
3. Proxy
4. None of the above
5. **How many test cases are necessary to cover all the possible sequences of statements (paths) for the following program fragment?**

**if (Condition 1)**

**then statement 1**

**else statement 2**

**fi**

**if (Condition 2)**

**then statement 3**

**fi**

1. 1
2. 2
3. 3
4. None of the above
5. **Acceptance test cases are based on what?**
6. Requirements
7. Design
8. Code
9. Decision table
10. **A common test technique during component test is**
11. Statement and branch testing
12. Usability testing
13. Security testing
14. Performance testing
15. **Statement Coverage will not check for the following.**
16. Missing Statements
17. Unused Branches
18. Dead Code
19. Unused Statement

**Answers:**

1.d

2.b

3.d

4.c

5.d

6.a

7.c

8.b

9.a

10.a

11.c

12.a

13.b

14.c

15.b

16.b

17.c

18.c

19.a

20.c

21.b

22.d

23.c

24.a

25.b

26.d

27.a

28.d

29.c

30.b

31.c

32.d

33.c

34.b

35.c

36.b

37.c

38.a

39.a

40.a