**Module: DevOps Introduction**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Employee ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Marks Secured: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Each Question has multiple choice answers and carries 2 marks

Attempt all questions and mark the correct answers.

1. The difference between unit and integration test.
   1. The unit test is white box and integration test is black box testing
   2. There should not be failures in integration test
   3. The unit test is unnecessary while the integration test is mandatory
   4. The integration test is conducted for every time a unit of application is built.
2. What are the issues and challenges in waterfall model of SDLC?
   1. Managing change in the requirements
   2. Testing of code
   3. Frequent delivery and update
   4. Design issues
3. What is Test Driven Development?
   1. The tests are written after the code is developed.
   2. The application requirements drive the test cases
   3. Test first approach.
   4. Every test should succeed at first time only
4. What is the cultural challenge for DevOps?
5. People need to learn new languages
6. People need to work together across traditional role boundaries
7. People need to work different hours
8. People need to move desks frequently
9. In TDD who defines the unit test cases?
10. The developer
11. Product QA Tester
12. Business Owner
13. The Manager
14. What is TDD cycle?
15. Every Test has first time failure
16. Test-Implementation -Test.
17. Test Failure- Implementation -Test Success-Refactor-Test
18. TDD is just a single stage of development.
19. Red-Green-Refactor
20. What are user acceptance tests?
21. Tests to verify that every functional requirement has been implemented
22. Testing that the user interface works
23. Testing that system components works together
24. Testing user interface usability
25. What should be the main consideration when writing code under TDD?
26. Adding checks for possible bad values
27. Writing the most efficient code
28. Write the minimum code to pass the test
29. Allowing for future expansion
30. What do code quality tools do?
31. Static analysis of source code
32. Check for compilation errors
33. Translate one language into another
34. Debug code
35. How do users get files from a remote Git repository?
36. By doing a check out
37. By doing an FTP transfer
38. By cloning the remote repository to a local copy
39. By adding the files using command gitadd
40. What advantage does Continuous Integration provide?
41. It simplifies the build process
42. It stops developers checking in bad code
43. Build errors are quickly detected and reported
44. There are no real advantages
45. What is a fake?
46. A test double which returns fixed values
47. A mocking framework
48. A full implementation not suitable for production
49. A test double which verifies how many times a method is called
50. What is code refactoring?
51. Restructure the code to match the new requirements
52. Restructure the code without changing the external functional behavior
53. Refactoring is done before writing any test cases.
54. Refactoring is done to improve code maintenance
55. Mark true or false
56. For every function code there should be only one test case
57. In One test case multiple behaviors are tested
58. The Test cases are dependent on each other for execution
59. Component under test is shared across all test cases.

1. What is assertion?
2. Positively assert the unit code
3. Verifies the result values and throws error in case of NOT matching.
4. Controls the unit flow.
5. Process of assertive locking
6. What is true of following?
7. Every test case should test for success only
8. The test cases run in sequence during test execution
9. The unit test cases are part of application delivery to client.
10. Each of the test cases must be executed manually.
11. What is the advantage of TDD?
12. The TDD makes the development incremental and iterative
13. The TDD can be applied to legacy code.
14. TDD increases the code coverage.
15. The TDD increases the flexibility of code to match the changes.
16. What is the use of mocking in testing?
17. It mocks frequently and is very funny.
18. The mocking allows testing of the code in absence of dependencies.
19. The mock objects allow verifying the behavior.
20. Mocks are used in production code.
21. The BDD and TDD difference
22. TDD supports unit test cases.
23. BDD is complete different than TDD
24. BDD is an extension of TDD to support Acceptance TDD
25. The TDD does not support BDD criteria
26. What is Gold Plating in coding?
27. Gold Plating is working on a task beyond the point where the extra effort is worth any value it adds.
28. Writing complicated code
29. Making user interfaces visually appealing
30. Work which adds little or no value

\*\*\*\*\*\*\*\*\*\*\*\*\*