SNS COLLEGE OF TECHNOLOGY

DEPARTMENT OF MCA

CA717 – SOFTWARE TESTING AND QUALITY ASSURANCE – IV SEMESTER

TWO MARKS QUESTIONS & ANSWERS

UNIT - I: TESTING ENVIRONMENT AND TEST PROCESSES

(1) Define software Testing.

Testing can be described as a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attributes.

(2) Define Debugging.

Debugging, or fault localization is the process of

- Locating the fault or defect.
- Repairing the codes.
- Retesting the code.

(3) What is an Error.

An error is mistake or misconception or misunderstanding on the part of a software developer.

(4) What is meant by Faults (Defects).

A fault is introduced into the software as the result of an error. It is an anomaly in the software that may cause nit to behave incorrectly, and not according to its specification.

(5) Define failures.

A failure is the inability of a software or component to perform its required functions within specified performance requirements.

(6) Define Test Cases.

A test case in a practical sense is a test related item which contains the following information.

- A set of test inputs. These are data items received from an external source by the code under test. The external source can be hardware, software, or human.

- Execution conditions. These are conditions required for running the test, for example, a certain state of a database, or a configuration of a hardware device.

- Expected outputs. These are the specified results to be produced by the code under test.

(7) Write short notes on Test, Test Set, and Test Suite.

A Test is a group of related test cases, or a group of related test cases and test Procedure.

A group of related test is sometimes referred to as a test set. A group of related tests that are associated with a database, and are usually run together, is sometimes referred to as a Test Suite.

(8) Define Validation.

Validation is the process of evaluating a software system or component during, or at the end of, the development cycle in order to determine whether it satisfies specified requirements.

(9) Define Verification.

Verification is the process of evaluating a software system or component to determine whether the product of a given development phase satisfy the conditions imposed at the start of that phase.

(10) Differentiate between verification and validation?

Verification	Validation
Verification is the process of evaluating software system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.	Validation is the process of evaluating software system or component during or at the end of the , the development phase satisfy the conditions imposed at the start of that phase.
Verification is usually associated with activities such as inspections and reviews of the s/w deliverables.	Validation is usually associated with Traditional execution _based testing, i.e., Exercising the code with test case.

(11) Differentiate between testing and debugging

Testing	Debugging
Testing as a dual purpose process • Reveal defects • And to evaluate quality attributes	Debugging or fault localization is the process of • Locating the fault or defect • Repairing the code, and • Retesting the code

(12) Define Test Oracle.

Test Oracle is a document, or a piece of software that allows tester to determine whether a test as been passed or failed.

(13) Define Test Bed.

A test bed is an environment that contains all the hardware and software needed to test a software component or a software system.

(14) Define Software Quality.

Quality relates to the degree to which a system, system component, or process meets specified requirements. Quality relates to the degree to which a system, system component, or process meets customer or user needs, or expectations.

(15) List the Quality Attributes.

- Correctness
- Reliability
- Usability
- Integrity
- Portability
- Maintainability
- Interoperability

(16) Define reviews.

A review is a group meeting whose purpose is to evaluate a software artifact or a set of Software artifacts. Review and audit is usually conducted by a SQA group.

(17) Define Errors.

An error is a mistake, misconception, or misunderstanding on the part of a software developer.

(18) Define Faults.

A fault (defect) is introduced into the software as the result of an error. It is an anomaly in the software that may cause it to behave incorrectly, and not according to its specification.

(19) Define Failures.

A failure is the inability of a software system or component to perform its required functions within specified performance requirements.

(20) Define Test case.

A test case in the practical sense is a test- related item which contain the following information:

- A set of test inputs.
- Execution conditions.
- Expected outputs.

SNS COLLEGE OF TECHNOLOGY

DEPARTMENT OF MCA

CA717 – SOFTWARE TESTING AND QUALITY ASSURANCE – IV SEMESTER

TWO MARKS QUESTIONS & ANSWERS

UNIT - II: TESTING TECHNIQUES AND LEVELS OF TESTING

(1) List the two basic Testing strategies.

- Black box testing.
- White box testing.

(2) What is Black-box testing?

- Knowing the specified function that a product has been designed to perform, test to see if that function is fully operational and error free
- Includes tests that are conducted at the software interface
- Not concerned with internal logical structure of the software

(3) What is White-box testing?

- Knowing the internal workings of a product, test that all internal operations are performed according to specifications and all internal components have been exercised
- Involves tests that concentrate on close examination of procedural detail
- Logical paths through the software are tested
- Test cases exercise specific sets of conditions and loops

(4) What are the knowledge sources for Black box testing?

- Requirements
- Document specification
- Domain knowledge
- Defect analysis data

(5) What are the knowledge sources for White box testing?

- High level design
- Detailed design
- Control flow graphs
- Cyclomatic complexity

(6) List the methods of Black box testing.

- Equivalence class partitioning
- Boundary value analysis
- State transition testing
- Cause and effect graphing
- Error guessing

(7) List the methods of White box testing.

- Statement testing
- Branch testing
- Path testing
- Data flow testing
- Mutation testing
- Loop testing

(8) What are the basic primes for all structured program.

- Sequential (e.g., Assignment statements)
- Condition (e.g., if/then/else statements)
- Iteration (e.g., while, for loops)

(9) Define Random testing.

Each software system or module has an input domain from which test input data is selected. If a tester randomly selects input from the domain, this is called Random testing.

(10) What is Equivalence class partitioning.

If a tester is viewing the software-under-test as a black box with well defined inputs and outputs, a good approach to selecting test inputs is to use a method called Equivalence class partitioning.

(11) List the advantages of Equivalence class partitioning.

- It eliminates the need for exhaustive testing, which is not feasible.
- It guides a tester in selecting a subset of test inputs with a high probability of detecting a defect.
- It allows a tester to cover a larger domain of inputs/outputs with a smaller subset selected from an Equivalence class.

(12) What is Cause effect graphing?

It is a technique that can be used to combine conditions and derive an effective set of test cases that may disclose inconsistencies in a specification.

(13) Define State.

A state is an internal configuration of a system or component. It is defined in terms of values assumed at a particular time for the variables that characterize the system or component.

(14) What is meant by Finite-state machine?

It is an abstract machine that can be represented by a state graph having a finite number of states and a finite number of transitions between states.

(15) Define Usage profiles.

Usage profiles are characterizations of the population of intended uses of the software in its intended environment.

(16) What is Certification?

Certification refers to third-party assurance that a product, process, or service meets a specific set of requirements.

(17) What is Test data set?

A test data set is statement, or branch, adequate if a test set T for program P causes all the statements, or branches, to be executed respectively.

(18) Define Path.

A path is a sequence of control flow nodes usually beginning from the entry node of a graph through to the exit node.

(19) List the two major assumptions in Mutation testing.

- The component programmer hypothesis
- The coupling effect

(20) Define Error guessing.

Error guessing approach is based on the testers/developers past experience with code similar to code-under-test, and their intuition as to where defects may lurk in the code.

(21) What is the goal of smart tester?

The goal of the smart tester is to understand the functionality, input/output domain, and the environment of use for the code being tested.

(22) List the different levels of testing.

- Unit test
- Integration test
- System test
- Acceptance test.

(23) Define Unit Testing

A unit is the smallest possible testable software component that can be characterized in several ways.

(24) List the components suitable for unit test.

- Procedures and functions
- Classes/objects and methods
- Procedure-sized reusable components.

(25) List the phases in the unit test planning.

- Phase 1: Describe unit test approach and risks.
- Phase 2: Identify unit features to be tested.
- Phase 3: Add levels of detailed to the plan.

(26) What is Test harness?

The auxiliary code developed to support to testing of units and components is called a test harness. The harness consists of drivers that call the target code and stubs that represent modules it calls.

(27) List the major goals of Integration test.

- To detect defects that occurs on the interfaces of units.
- To assemble the individual units into working subsystems and the finally a complete system that is ready for system test

(28) What are the Integration strategies?

- Top Down: In this strategy integration of the module begins with testing the upper level modules.
- Bottom Up: In this strategy integration of the module begins with testing the lowest level modules.

(29) What is the advantage of Bottom up integration?

Bottom-up integration has the advantage that the lower-level modules are usually well tested early in the integration process. This is important if these modules are candidates for reuse.

(30) What is meant by a stub and driver?

Driver

A simple main program that accepts test case data, passes such data to the component being tested, and prints the returned results.

Stubs

Serve to replace modules that are subordinate to (called by) the component to be tested.

(31) List the several types of system tests.

- Functional testing
- Performance testing
- Stress testing
- Configuration testing
- Security testing
- Recovery testing

(32) What are the two major requirements in the Performance testing?

- Functional Requirement: User describes what functions the software should perform. We test for compliance of the requirement at the system level with the functional based system test.
- Quality Requirement: They are non functional in nature but describe quality levels expected for the software.

(33) Define stress Testing.

When a system is tested with a load that causes it to allocate its resources in maximum amounts. It is important because it can reveal defects in real-time and other types of systems.

(34) Define Alpha and Beta Test.

Alpha test developer's to use the software and note the problems. Beta test who use it under real world conditions and report the defect to the Developing organization.