UNIT - 5

CHAPTER 5

Software Testing Strategies

5.1 Software Testing Strategies

Syllabus Topic : Levels of Testing : Introduction

5.1.1 Levels of Testing

Q. 5.1.1 Explain levels of testing (Ref. Sec. 5.1.1) **(5 Marks)**

- Testing is the process of exercising a program with the specific intent of finding errors prior to delivery to the end user.
- A level of software testing is a process where every unit or component of a software/system is tested. The main goal of system testing is to evaluate the system's compliance with the specified needs.
- There are many different testing levels which help to check behavior and performance for software testing.

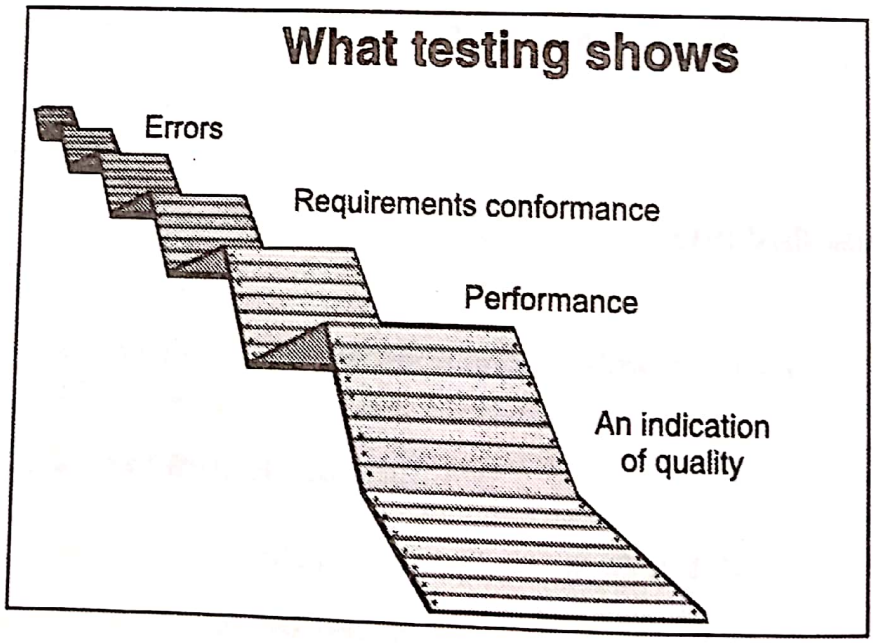

Fig. 5.1.1

Syllabus Topic : Proposal and Requirement Testing

5.1.2 Proposal and Requirement Testing

Q. 5.1.2 What is proposal and requirement testing? (Ref. Sec. 5.1.2) (5 Marks)

- Proposal and Requirements testing is done to clarify whether project requirements are feasible or not in terms of time, resources and budget.
- Many bugs emerge in software because of incompleteness, inaccuracy and ambiguities in functional requirements.
- That's why it is highly important to test requirements and eliminate ambiguities before you start to develop a project.
- This type of testing covers testing of requirements specification that describes :
  - project functionality
  - user interface
  - software and hardware interfaces
  - performance criteria
  - implementation issues and risks
  - security and system correctness criteria

5.1.2.1 Stages in Requirements based Testing

1. Defining Test Completion Criteria

Testing is completed only when all the functional and non-functional testing is complete.

2. Design Test Cases

A Test case has five parameters namely the initial state or precondition, data setup, the inputs, expected outcomes and actual outcomes.

3. Execute Tests

Execute the test cases against the system under test and document the results.

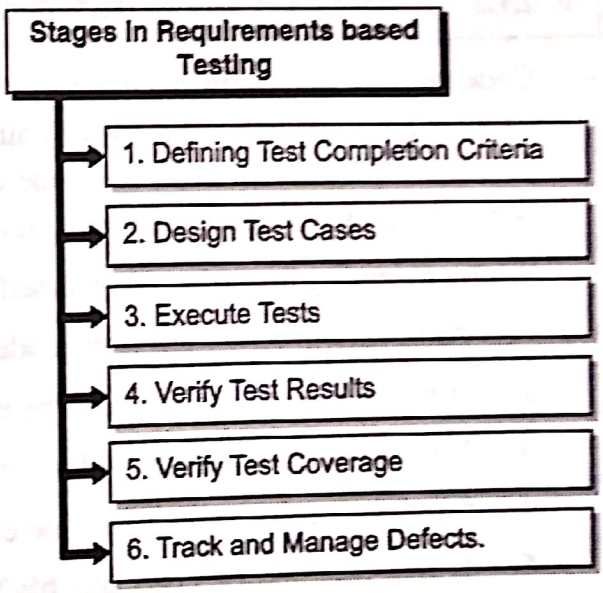

Fig. 5.1.2

➜ **4. Verify Test Results**

Verify if the expected and actual results match each other.

➜ **5. Verify Test Coverage**

Verify if the tests cover both functional and non-functional aspects of the requirement.

➜ **6. Track and Manage Defects**

Any defects detected during the testing process goes through the defect life cycle and are tracked to resolution. Defect Statistics are maintained which will give us the overall status of the project.

5.1.2.2 Requirements Testing Process

- Testing must be carried out in a timely manner.
- Testing process should add value to the software life cycle, hence it needs to be effective.
- Testing the system exhaustively is impossible hence the testing process needs to be efficient as well.
- Testing must provide the overall status of the project, hence it should be manageable.

Syllabus Topic : Code Review

5.1.3 Code Review

Q. 5.1.3 Explain code review. (Ref. Sec. 5.1.3) (5 Marks)

- Code review is a software quality assurance activity in which one or multiple humans check a program, primarily by watching and reading parts of its source code, and they do so after implementation. At least one of the humans must not be the code's author. Those performing the checking, excluding the author, are called "reviewers".
- While reviewing the code, ask yourself the following basic questions :
  1. Am I able to **understand** the code easily?
  2. Is the code written following the **coding standards/guidelines**?
  3. Is the same code **duplicated** more than twice?
  4. Can I **unit test / debug** the code easily to find the root cause?
  5. Is this function or class **too big**? If yes, is the function or class having too many responsibilities?

Code reviews are usually performed to achieve

1. Better code quality

Improve internal code quality

2. Finding defects

Improve quality by finding performance problems, security vulnerabilities, injected malware, etc.

3. Knowledge transfer

Help in transferring knowledge about the codebase, solution approaches, expectations regarding quality, etc.; both to the reviewers as well as to the author.

4. Increase sense of mutual responsibility

Increase a sense of collective code ownership.

5. Finding better solutions

Generate ideas for new and better solutions.

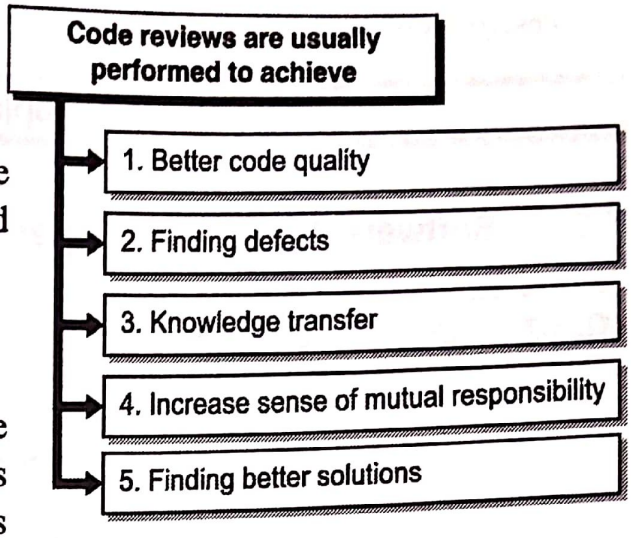

Fig. 5.1.3

5.1.3.1 Principles of Good Code review

- The first and foremost principle of a good review is this: if you commit to review code, review it thoroughly! Expect to spend a decent amount time on this. Be sure to read the code, don't just skim it, and apply thought to both the code and its style.
- Aim to understand every changed line. Research things you don't understand. Ask questions.
- Don't assume the code works - build and test it yourself! You should actually pull down the code and test it out.
- Follow up on reviews. After suggesting changes, you should be prepared to review it again. Ensure the necessary changes were made, and any problems you found were reasonably resolved.

Syllabus Topic : Design Testing

5.1.4 Design Testing

Q. 5.1.4 What is design testing? (Ref. Sec. 5.1.4) **(5 Marks)**

- Design is not just what it looks like and feels like, Design is how it works. Design depends on several key factors such as usability, utility, desirability, attractiveness etc.
- It is used to test on designs that are still in progress and before they're linked together as a prototype. Design testing offers a chance to gather feedback early in the process and shape design decisions.

Syllabus Topic : Testing Stages

5.2 Software Testing Strategies / Stages

Q. 5.2.1 List and explain software testing strategies. (Ref. Sec. 5.2) **(5 Marks)**

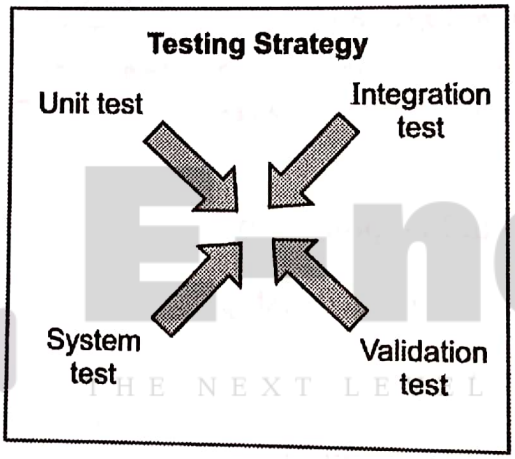

Fig. 5.2.1

- We start by 'testing-in-the-small' and head toward 'testing -in-the-large'
- For conventional software, Our initial focus is the module (component) and integration of modules follows.

Syllabus Topic : Unit Testing

5.2.1 Unit Testing

Q. 5.2.2 Explain Unit testing. (Ref. Sec. 5.2.1) **(5 Marks)**

Unit testing focuses verification effort on the smallest unit of software design – the software component or module.

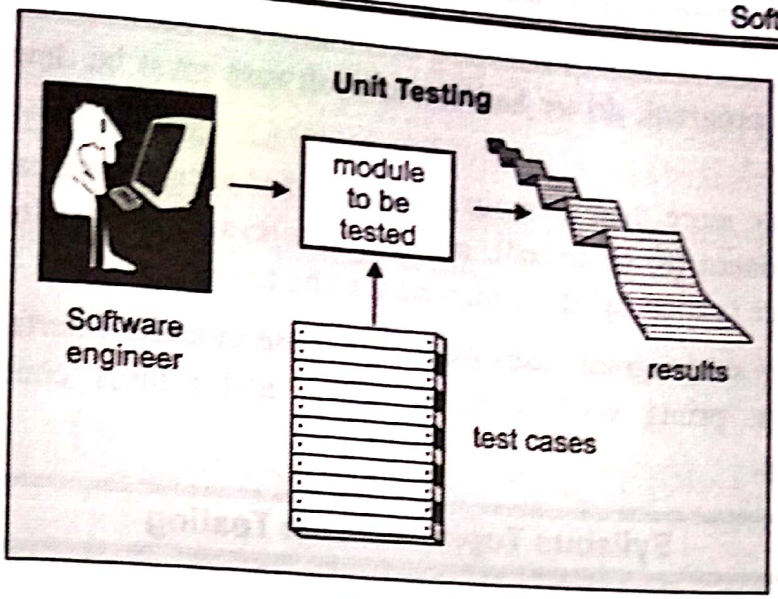

Fig. 5.2.2

5.2.1.1 Unit Test Considerations

The module interface is tested to ensure that information properly flows into and out of the program unit under test.

- The local data structure is checked to ensure that the data stored temporarily retains its Integrity during all steps in an algorithm's execution.
- Boundary conditions are tested to make sure that the module operates properly at boundaries established to restrict processing.
- All independent paths (basis paths) through the control structure are exercised to ensure that all statements in a module have been executed at least once.
- And finally, all error handling paths are tested.

5.2.1.2 Unit Test Procedures

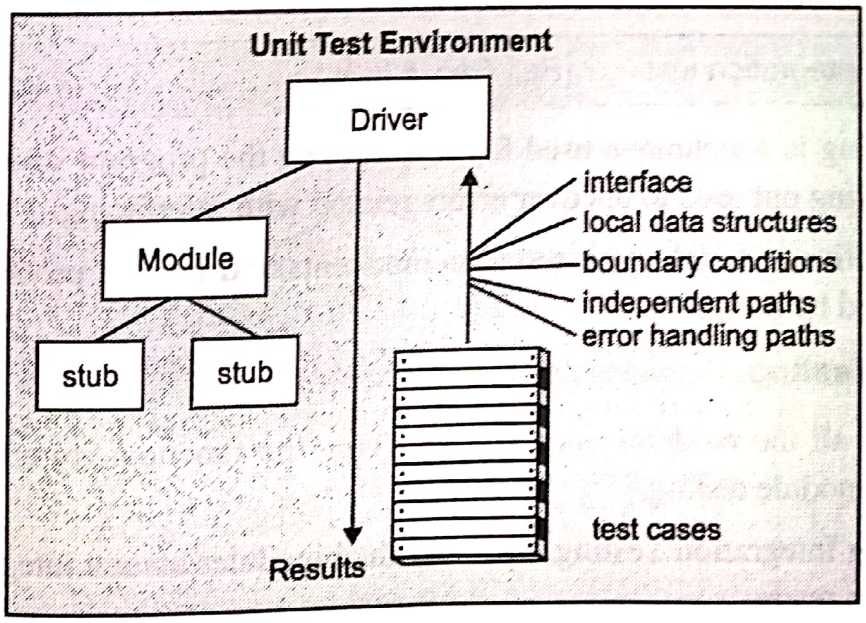

Fig. 5.2.3

- Each test case should be combined with a set of expected results. Because a component is not a stand-alone program, driver and/or stub software must be developed for each unit test.
- A driver is nothing more than a "main program" that accepts test case data, passes such data to the component (to be tested), and prints relevant results. Stubs replace modules that are subordinate (called by) the component to be tested.
- A stub or "dummy subprogram" uses the subordinate modules interface, may do minimal data manipulation, prints verification of entry, and returns control to the module undergoing testing.

Syllabus Topic : Module Testing

5.2.2 Module Testing

Q. 5.2.3 What is module testing? (Ref. Sec. 5.2.2) **(5 Marks)**

- It is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program.
- Module testing is largely a white box oriented. The objective of doing Module, testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module testing allows to implement parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

Syllabus Topic : Integration Testing, Big Bang Testing, Sandwich Testing

5.2.3 Integration Testing

Q. 5.2.4 What is integration testing? (Ref. Sec. 5.2.3) **(5 Marks)**

- Integration testing is a technique used for constructing the program structure while at the same time carrying out tests to uncover errors related with interfacing.
- The main objective is to take unit tested components and build a program structure that has been dictated by design.

(I) Big Bang Testing

- Combining all the modules once and verifying the functionality after completion of individual module testing.
- In Big Bang Integration Testing, the individual modules are not integrated until all the modules are ready.

- Then they will run to check whether it is performing well. In this type of testing, some disadvantages might occur like, defects can be found at the later stage.
- It would be difficult to find out whether the defect arouse in interface or in module.
- As shown in the diagram below, all the modules from 'Module -1' to 'Module- 6' are integrated simultaneously and then the testing is carried.

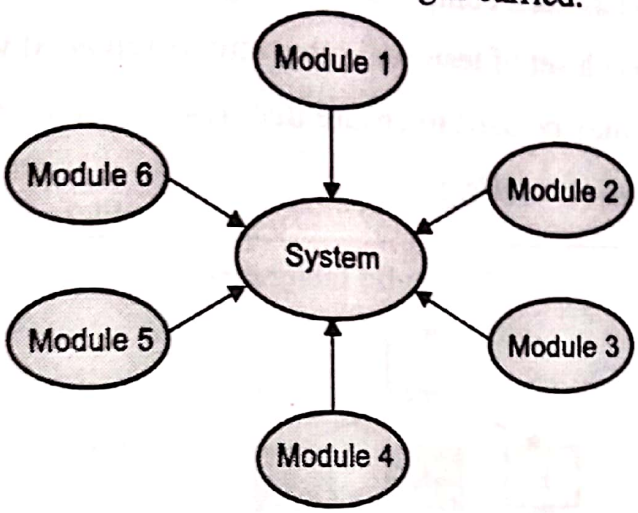

Fig. 5.2.4

- Advantages and disadvantages of Big Bang testing are:

Advantages

- The main advantage is that everything is finished before integration testing starts.
- It is extremely convenient for small systems.

Disadvantages

- There exists difficulty in fault localization.
- High chance of missing some interfaces links to be tested.

(II) Top-down integration testing

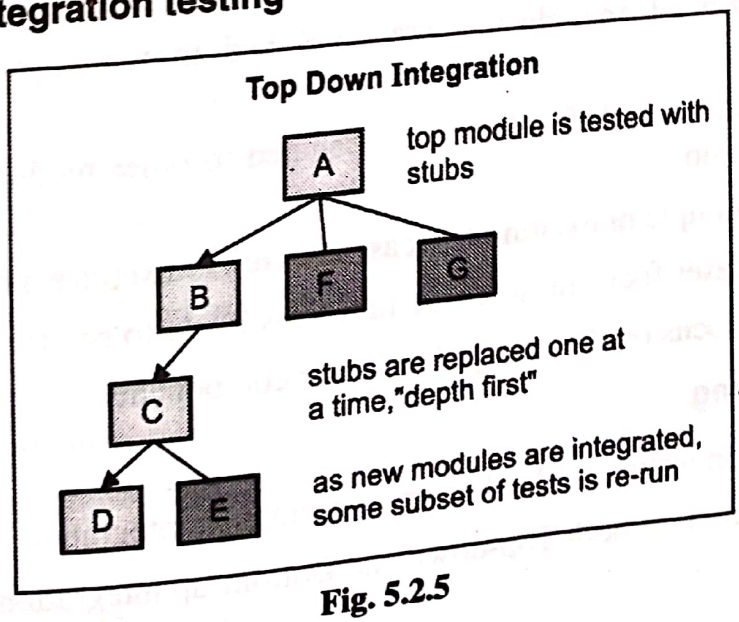

Fig. 5.2.5

- Main control module used as a test driver and stubs are substitutes for components directly subordinate to it.
- Subordinate stubs are replaced one at a time with real components (following the depth-first or breadth-first approach).
- Tests are conducted as each component is integrated.
- On completion of each set of tests and other stub is replaced with a real component.
- Regression testing may be used to ensure that new errors not introduced.

(iii)Bottom-up Integration testing

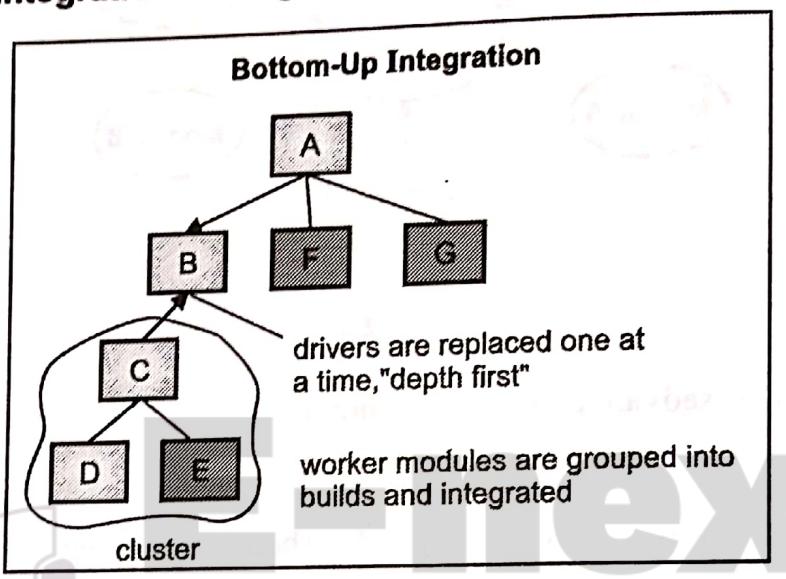

Fig. 5.2.6

- Low level components are combined in clusters that perform a specific software function.
- A driver (control program) is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure.
- Regression testing (check for defects propagated to other modules by changes made to existing program)
- Representative sample of existing test cases is used to exercise all software functions.
- Additional test cases focusing software functions likely to be affected by the change.
- Tests cases that focus on the changed software components.

(iv)Sandwich Testing

- Hybrid integration testing is also known as Sandwich integration testing.
- It is the combination of both Top-down and Bottom-up integration testing.

- In Hybrid Integration Testing, we exploit the advantages of Top-down and Bottom-up approaches. As the name suggests, we make use of both the Integration techniques.

Fig. 5.2.7

(v) Sandwich Integration Testing - Features

- It is viewed as three layers; viz - The Main Target Layer, a layer above the target layer and a layer below the target layer.
- Testing is mainly focused for the middle level target layer and is selected on the basis of system characteristics and the structure of the code.
- Hybrid Integration testing can be adopted if the customer wants to work on a working version of the application as soon as possible aimed at producing a basic working system in the earlier stages of the development cycle.

5.2.4 Validation Testing

- Ensure that each function or performance characteristic conforms to its specification.
- Deviations (deficiencies) must be negotiated with the customer to establish a means for resolving the errors.
- Configuration review or audit is used to ensure that all elements of the software configuration have been properly developed, cataloged, and documented to allow its support during its maintenance phase.

Syllabus Topic : Critical Path First

5.2.5 Critical Path First

Q. 5.2.5 Explain Critical path first method (Ref. Sec. 5.2.5) **(5 Marks)**

- Critical Path Analysis (CPA) is an effective method for planning and managing projects. Let's consider a very simple example, cooking and serving a fried breakfast. We can begin by listing all the tasks required in roughly correct order :
  - assemble plates and cutlery
  - assemble ingredients
  - prepare cooking equipment
  - make toast
  - fry sausages and eggs
  - grill bacon and tomatoes

- set table
- warm plates
- serve breakfast

– The order isn't quite right as some of these tasks need to take place simultaneously. Also, some tasks must be started before others, and some tasks must be completed before others can be started, eg:
  - The plates need to warm while other tasks are taking place
  - The toast needs to be made while the sausages are frying, and the bacon and tomatoes are grilling.
  - The eggs need to be fried last.

– The critical path analysis represents what tasks need done, and when they need done, as a diagram. Timescales and costs can be attached to each activity and resource. The diagram alongside shows the CPA for making a fried breakfast.

Syllabus Topic : Sub System Testing

5.2.6 Sub System Testing

Q. 5.2.6 Explain sub-system testing. (Ref. Sec. 5.2.6) **(5 Marks)**

– Sub-system integration testing focuses on testing the external APIs (Application Programming Interfaces) between sub-systems.

– Sub-system integration test plans must be created prior to system integration test execution. The following is a sub-system integration testing test plan for the microwave oven example.

Sub-System	Test Description
Command Processing System Testing	Tests for errors in integration and functioning of system sub-components. Assures proper operation of Input Monitoring and Processing, Clock, Timer, Auto Cook, and Auto Defrost functions.
Cooking Control System Testing	Tests for errors in integration and functioning of cooking control function and oven control function.
Output Processing System	Tests for errors in integration of timer display, clock display, and beeper signaling.

Syllabus Topic : System Testing

5.2.7 System Testing

Q. 5.2.7 What is system testing? (Ref. Sec. 5.2.7) **(5 Marks)**

- System testing of software or hardware is testing conducted on a whole, integrated system to estimate the systems compliance with its specified set of requirements.
- The only things Tester should be testing at the System Test stage are things that he and she couldn't test before.
- It is done to check how the system as a whole is functioning. Here the functionality and performance of the system is validated.
- User is not involved in System Testing.
- It is performed before Acceptance Testing.
- System Testing involves both Functional and Non- Functional Testing.

5.2.7.1 Types of System Testing

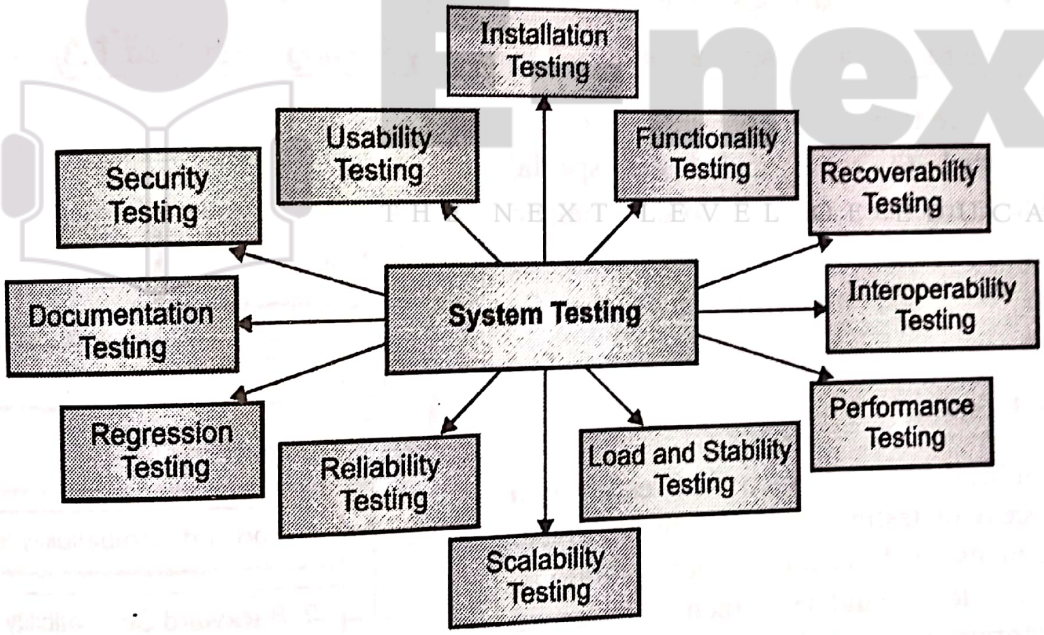

Fig. 5.2.8

- **Functionality Testing :** To make sure that functionality of product is working as per the requirements defined, within the capabilities of the system.
- **Recoverability Testing :** To make sure how well the system recovers from various input errors and other failure situations.
- **Interoperability Testing :** To make sure whether the system can operate well with third-party products or not.
- **Performance Testing :** To make sure system's performance under the various condition, in terms of performance characteristics.

- **Scalability Testing :** To make sure system's scaling abilities in various terms like user scaling, geographic scaling and resource scaling.
- **Reliability Testing :** To make sure system can be operated for longer duration without developing failures.
- **Regression Testing :** To make sure system's stability as it passes through an integration of different subsystems and maintenance tasks.
- **Documentation Testing :** To make sure that system's user guide and other help topics documents are correct and usable.
- **Security Testing :** To make sure that system does not allow unauthorized access to data and resources.
- **Usability Testing :** To make sure that system is easy to use, learn and operate.

Syllabus Topic : Special Tests : Introduction, GUI Testing, Compatibility Testing

5.3 Special Tests

Q. 5.3.1 Explain the below special tests : GUI Testing. (Ref. Sec. 5.3) **(5 Marks)**

Q. 5.3.2 Explain the below special tests : Compatibility Testing. (Ref. Sec. 5.3) **(5 Marks)**

There are certain types of tests that are generally viewed to come under the special category. These tests are planned and documented according to the same rules and standards as the other types of tests, but they have specific applications.

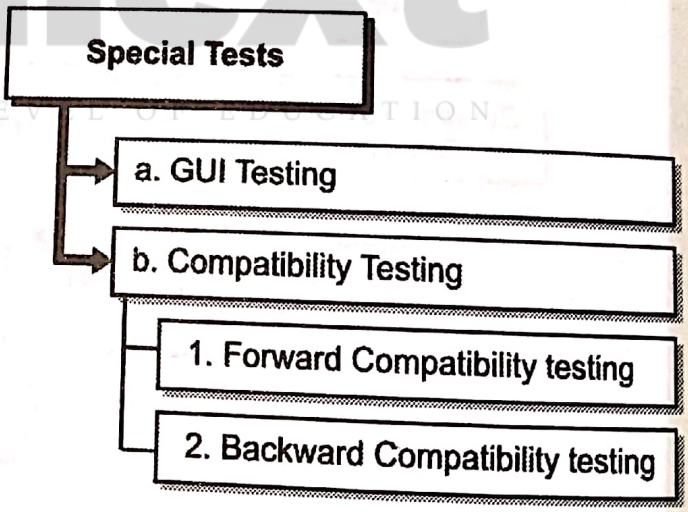

Fig. 5.3.1

➜ (a) GUI Testing

- Graphical User Interface GUI testing is a process of testing an application's visual elements, such as images, texts, buttons, etc., to validate their expected performance as well as their functional accuracy.
- In addition to functionality, GUI testing evaluates design elements such as layout, colors, fonts, font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, links and content.
- GUI testing processes can be either manual or automatic, and are often performed by third -party companies, rather than developers or end users.
- A normal User first observes the design and looks of the Application/Software and how easy it is for him to understand the UI.

- If user is not comfortable with the interface or finds it difficult to use then there are very less chances of user using that application ever again. This makes GUI testing important.
- GUI testing ensures the following :
  (i) The position, size, width and length of the element
  (ii) Whether the GUI element accepts the input
  (iii) Font used is readable
  (iv) Error messages are displayed properly
  (v) Alignment of the text is proper
  (vi) Images have good quality and are aligned properly

(b) Compatibility Testing

Compatibility Testing is a type of Software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or Mobile devices.

Need for compatibility testing

- Software released should be compatible with all type of hardware, platforms and OS. This improves quality of the software.
- These quality products in turn increase the sales and reputation of the company.
- It also increases the stability of the software and ensures trust of the customers.
- **For E.g. :** Compatibility testing for a mobile software application should be done to check whether it is compatible on different.
  - OS versions - Android versions such as Marshmallow, Nougat, Oreo or iOS versions such as iOS 12, iOS 11
  - Screen size such as 4.7 inch, 5.5 inch etc.
  - Devices - Phones, Tabs etc
  - Networks 3G, 4G , Wi-fi etc
- There are two types of compatibility testing:
  - **Forward Compatibility testing**

    This testing is done to verify the behaviour of the software with upcoming or new versions of hardware, OS, platforms etc
  - **Backward Compatibility testing**
  - This testing is done to verify the behaviour of the software with older versions of hardware, OS, platforms etc

Syllabus Topic : Security Testing

5.4 Security Testing

Q. 5.4.1 Explain the below special tests : Security Testing. (Ref. Sec. 5.4) (5 Marks)

- This testing is done to verify whether all the data and resources are protected from any outside attack. It aims at evaluating various aspects of security such as confidentiality, Authenticity, Integrity, Availability, Authorization and Non-repudiation.
- Software applications now days are used extensively for almost all purposes. There are certain software which stores sensitive data. These applications needs to be tested to
  - Uncover security vulnerability
  - Proper authorization i.e. only authorized users can access the application
  - Integrity issue i.e. data should not be changed etc

5.4.1 Testing Techniques

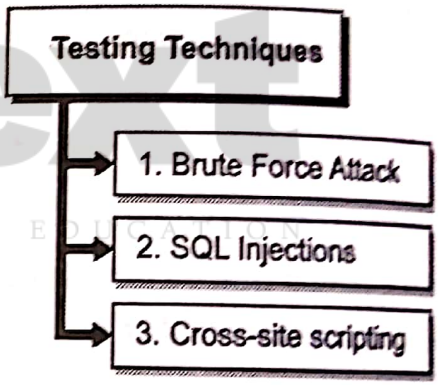

Fig. 5.4.1

➜ **1. Brute Force Attack**

This is the simplest way to gain access to a website or a server.

It tries various combinations of username and password until it is cracked. This action is similar to an army attacking to get into the fort.

➜ **2. SQL Injections**

In this an attacker embeds a malicious code in a poorly designed application which is then passed to SQL database. A successful attack may result in the unauthorized viewing of user lists, the deletion of entire tables and, in certain cases, the attacker gaining administrative rights to a database, all of which are highly detrimental to a business.

➜ **3. Cross-site scripting**

It is a type of security attack in which an attacker injects malicious code or script into a website.

Syllabus Topic : Performance Testing, Volume Testing, Stress Testing

5.5 Performance Testing

Q. 5.5.1	Explain the below special tests : Performance Testing. (Ref. Sec. 5.5)	(5 Marks)
Q. 5.5.2	Explain the below special tests : Volume Testing. (Ref. Sec. 5.5)	(5 Marks)
Q. 5.5.3	Explain the below special tests : Stress Testing. (Ref. Sec. 5.5)	(5 Marks)

- This testing is carried out to measure the performance of software parameter such as responsiveness and stability under certain load. The goal of performance testing is to eliminate performance issues. It is a form of non-functional testing.
- The focus of software testing is on software's :
  - (i) **Speed :** response time of an application
  - (ii) **Scalability:** How many users an application can handle
  - (iii) **Stability:** Application is stable and does not break under the load applied
  - (iv) **Reliability :** Provides reliable result even under load

5.5.1 Types of Performance Testing

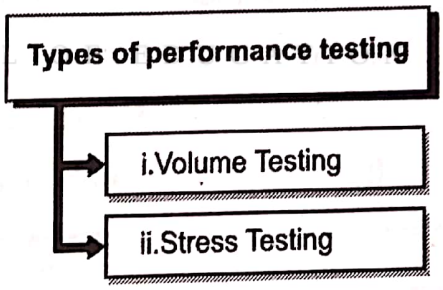

Fig. 5.5.1

(i) Volume Testing

Application database is loaded with lot of data and overall performance of the system is monitored. It is done to check the software or app for its performance against huge data of the database. It is also referred to as flood testing.

Need for volume testing

- Helps to identify the issue or problems that are likely to occur with increasing amount of data.
- Identifies the point where degradation of the performance starts
- Determines the capacity of the system or application

(ii) Stress Testing

It is performed to evaluate the behavior of software under the stress beyond its limits. The system is monitored after subjecting it to overload to check how much stress it can withstand. The objective is to identify the breaking point.

Need for stress testing

- To check whether the system works under abnormal conditions.
- System failure under extreme conditions could result in enormous revenue loss. It prepares for extreme conditions
- Helps to accommodate abnormal increase in traffic for e.g. During any offer the traffic of the website increase.

Syllabus Topic : Recovery Testing

5.6 Recovery Testing

Q. 5.6.1 Explain the below special tests: Recovery Testing. (Ref. Sec. 5.6) (5 Marks)

- In recovery testing the system is forced to fail in order to verify whether recovery is possible after a disaster. It is basically testing how well a system recovers from crashes, hardware failures, or any other failure.
- Recovery testing verifies the following:
  - System can recover all lost data
  - How much time is required to resume normal operations
  - How much recovery is possible
  - Automatic reconnecting occurs
- It is important for all businesses. No business can afford loss of data.

Syllabus Topic: Installation Testing

5.6.1 Installation Testing

Q. 5.6.2 Explain the below special tests: Installation Testing. (Ref. Sec. 5.6.1) (5 Marks)

- Installation testing is aimed to verify successful installation, updates or deletion of software. Most software systems have installation procedures that are needed before they can be used for their main purpose.
- Testing these procedures to achieve an installed software system that may be used is known as installation testing.
- A smooth installation creates a good impression on the user. It is performed under various conditions to check whether the installation is successful such as numerous system running, diverse environment and hardware configuration etc.

Syllabus Topic : Requirement Testing

5.6.2 Requirement Testing

Q. 5.6.3 Explain the below special tests : Requirement Testing. (Ref. Sec. 5.6.2) (5 Marks)

- It is a testing process in which the test cases, scenario, conditions are written from the requirements. It is the process of gathering requirements, understanding them, writing test cases and executing them to verify system satisfies all the requirements.
- Requirements testing are done to clarify whether project requirements are feasible or not in terms of time, resources and budget. Many errors occur due to ambiguous or incomplete requirements hence it is important to test requirements and remove such ambiguity.

Syllabus Topic : Regression Testing

5.6.3 Regression Testing

Q. 5.6.4 Explain the below special tests : Regression Testing. (Ref. Sec. 5.6.3) (5 Marks)

Regression testing is performed to verify fixing of a defect has not caused any other feature to break. We do regression testing by re-executing the tests against the modified application to evaluate whether the modified code breaks anything which was working earlier. Anytime we modify an application, we should do regression testing.

Syllabus Topic : Error Handling Testing

5.6.4 Error Handling Testing

Q. 5.6.5 Explain the below special tests : Error Handling Testing. (Ref. Sec. 5.6.4) (5 Marks)

- This testing is done to ensure that the system is capable of handling errors. Error handling testing can be done by performing some improper transactions in between proper transactions and check how the system is behaves during improper transactions whether it identifies the problem and handles it appropriately.
- A group of people should anticipate what can go wrong and design system in such a way that these situations are handled.

Syllabus Topic : Manual Support Testing

5.6.5 Manual Support Testing

Q. 5.6.6 Explain the below special tests : Manual support Testing. (Ref. Sec. 5.6.5) **(5 Marks)**

- It is a testing technique that involves testing of all the functions performed by the people while preparing the data and using these data for automated system.
- In this manual testing support document and procedure are verified. It checks whether people involved in the process are adequately trained.

Syllabus Topic : Intersystem Testing

5.6.6 Intersystem Testing

Q. 5.6.7 Define Intersystem testing. (Ref. Sec. 5.6.6) **(5 Marks)**

- An application is many times hosted across different locations; however, all data needs to be deployed over a central location.
- The process of testing the integration points for single application hosted at different locations and then ensuring correct data flow across each location is known as inter system testing.

Syllabus Topic : Control Testing

5.6.7 Control Testing

Q. 5.6.8 Define : Control testing. (Ref. Sec. 5.6.7) **(5 Marks)**

- Control testing is a tool of management that helps them to keep a tab on the activities performed by the testing team.
- It is used to make sure activities performed are in accordance to the management plan.

Syllabus Topic : Smoke Testing

5.6.8 Smoke Testing

Q. 5.6.9 Explain the below special tests : Smoke Testing. (Ref. Sec. 5.6.8) **(5 Marks)**

- Smoke testing is a testing technique to check whether the software or system is ready/stable for conducting further testing.
- If the test fails the build is declared as unstable to continue testing. It is also known as build verification testing.

- The purpose of the smoke testing is to ensure that the critical functionalities of an application are working fine. Smoke testing is like a normal health check-up of the build of an application.

Syllabus Topic : Adhoc Testing

5.6.9 Adhoc Testing

Q. 5.6.10 Explain the below special tests : Adhoc Testing. (Ref. Sec. 5.6.9) (5 Marks)

- This testing is performed without any proper planning and documentation in a random order.
- The tests are conducted informally and randomly without any formal expected results. The defects found are difficult to reproduce.
- Defects found in this process would never have been found by executing the written test case.

Syllabus Topic : Parallel Testing

5.6.10 Parallel Testing

Q. 5.6.11 What is parallel testing? (Ref. Sec. 5.6.10) (5 Marks)

- Parallel testing is a testing technique in which the same inputs are entered in two different versions of the application and reporting the anomalies.
- It reduces time and verifies the compatibility newly developed system with the old system.

Syllabus Topic : Execution Testing

5.6.11 Execution Testing

Q. 5.6.12 Define execution testing. (Ref. Sec. 5.6.11) (5 Marks)

Execution testing ensures that test cases are executed in proper format.

Syllabus Topic : Operations Testing

5.6.12 Operations Testing

Q. 5.6.13 What is operation testing? (Ref. Sec. 5.6.12) (5 Marks)

- Operational testing ensures operation readiness of the software. It confirms that a product, service, process or system meets operational requirements.

- Operational requirements include things such as performance, security, stability, maintainability, accessibility, interoperability, localization, backup, recovery and support documentation.
- Such requirements are typically developed by business operations teams who are responsible for maintaining and operating a process, system or service.

Syllabus Topic : Compliance Testing

5.6.13 Compliance Testing

Q. 5.6.14 Explain the below special tests : Compliance Testing. (Ref. Sec. 5.6.13) (5 Marks)

Compliance testing confirms the system complies with all the standards, process and procedures. It determines whether all the defined standards are implemented.

Syllabus Topic : Usability Testing

5.6.14 Usability Testing

Q. 5.6.15 Explain the below special tests : Usability Testing. (Ref. Sec. 5.6.14) (5 Marks)

This testing is user centric testing. It evaluates the software to check its usability from user point of view. It concentrates on effectiveness of the system.

Syllabus Topic : Decision Table Testing

5.6.15 Decision Table Testing

Q. 5.6.16 Explain the below special tests : Decision Table Testing. (Ref. Sec. 5.6.15) (5 Marks)

- Decision table testing is a technique for writing test cases. It is a good way to deal with conditions where there are multiple inputs which results in different outputs. This method is also known as cause-effect table.
- Consider an example of a ticket booking page:
- If the user enters both "From" and "To" city only then search button should be enabled.

From City	Y	N	Y	N
To City	N	Y	Y	N
Output				
Search Button	Disabled	Disabled	Disabled	Enabled

- Such representation of input and output combination in a table makes it easy to write test scenarios.

Syllabus Topic : Documentation Testing

5.6.16 Documentation Testing

Q. 5.6.17 Define documentation testing (Ref. Sec.5.6.16) **(5 Marks)**

- All the documents specifying certain activities, requirements, procedures or results are tested to check its quality.
- Documents tested generally includes requirements document, test plan, test suites etc. If the documentation is not right: there will be major problems.
- Documentation testing can start at the very beginning of the software process and hence save large amounts of money, since the earlier a defect is found the less it will cost to be fixed.

Syllabus Topic : Training Testing

5.6.17 Training Testing

Q. 5.6.18 Define training testing (Ref. Sec.5.6.17) **(5 Marks)**

- Training testing is concerned with verifying whether the resources working have proper training to perform the task assigned.
- It enables to evaluate the gap between the required skill set and actual skill set of the resources and provide necessary training.
- If the people working are trained properly then chance of mistakes or errors reduces.

Syllabus Topic : Rapid Testing

5.6.18 Rapid Testing

Q. 5.6.19 Explain the below special tests : Rapid Testing. (Ref. Sec. 5.6.18) **(5 Marks)**

- Rapid testing is a context-driven methodology for testing any product that includes or involves software.
- Through interactive discussion and hands-on activities, we challenge assumptions and expose common misconceptions about testing practices.
- Then we show you a clear and powerful way to think about testing that allows you to test responsibly and systematically, so that you focus on business risk and do the deep testing that your project needs.

Syllabus Topic : Control Flow Graph

5.7 Control Flow Graph

Q. 5.7.1 Explain control flow graph. (Ref. Sec. 5.7) **(5 Marks)**

- A control flow graph is the graphical representation of a program during its execution. The control flow graph G = (N, E) of a program consists of a set of nodes N and a set of edge E. Each node represents a set of program statements.
- There is an edge from node n1 to node n2 if the control may flow from the last statement in n1 to the first statement in n2.
- Control flow testing uses the control structure of a program to develop the test cases for the program.
- The test cases are developed to sufficientlyv cover the whole control structure of the program. The control structure of a program can be represented by the control flow graph of the program.

```
int evensum(int i)
{
int sum = 0;
while (i <= 10)
{
if (i/2 == 0)
sum = sum + i;
i++;
}
return sum;
}
```

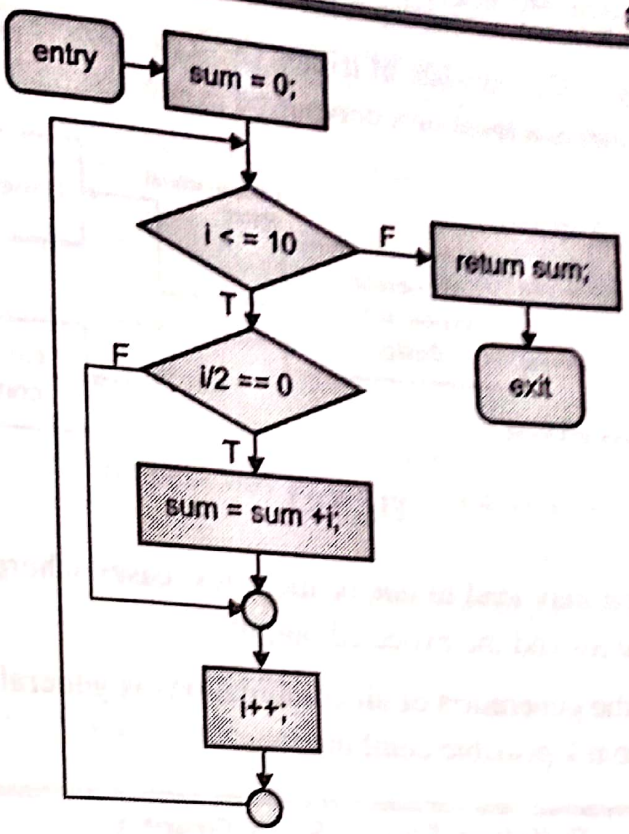

Fig. 5.7.1

Syllabus Topic : Generating Tests on the Basis of Combinatorial Designs

5.8 Generating Tests on the Basis of Combinatorial Designs

Q. 5.8.1 Explain generating test cases on the basis of combinatorial designs. (Ref. Sec. 5.8) **(5 Marks)**

- Software applications are often designed to work in a variety of environments. Combinations of factors such as the operating system, network connection, and hardware platform, lead to a variety of environments.
- Each environment corresponds to a given set of values for each factor, known as a test configuration. For example, Windows XP, Dial-up connection, and a PC with 512 MB of main memory, is one possible configuration.
- To ensure high reliability across the intended environments, the application must be tested under as many test configurations, or environments, as possible.
- Consider program P that takes two integers $x > 0$ and $y > 0$ as inputs. The input space of P is the set of all pairs of positive non-zero integers.
- Now suppose that this program is intended to be executed under the Windows and the Mac OS operating system, through the Netscape or Safari browsers, and must be able to print to a local or a networked printer.

- The configuration space of P consists of triples (X, Y, Z) where X represents an operating system, Y a browser, and Z a local or a networked printer.

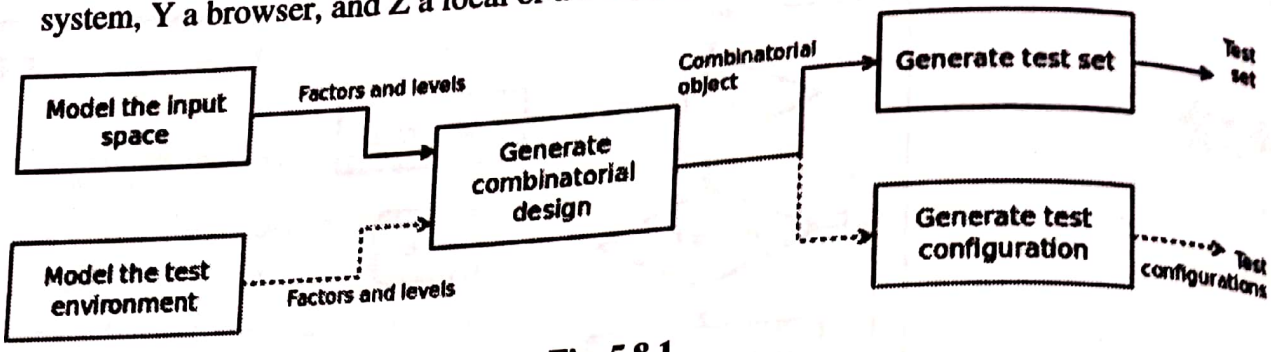

Fig. 5.8.1

- Each factor combination may lead to one or more test cases where each test case consists of values of input variables and the expected output.
- Nevertheless, as usual the generation of all combinations is generally not feasible k factors with n level each lead to n k possible combinations.

Syllabus Topic : State Graphs

5.8.1 State Graphs

Q. 5.8.2 Explain state transition diagram. (Ref. Sec. 5.8.1) (5 Marks)

- State graph shows the current state of the system and what state it will move to based on the input provided.
- A System's transition is represented as shown in the below diagram :

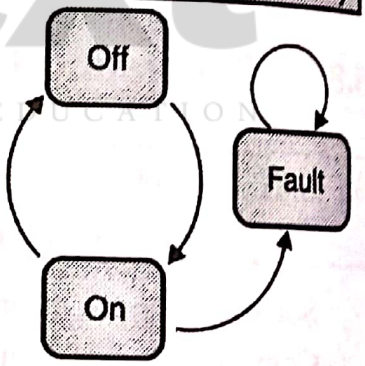

Fig. 5.8.2

- The tests are derived from the above state and transition and below are the possible scenarios that need to be tested.

Tests	Test 1	Test 2	Test 3
Start State	Off	On	On
Input	Switch ON	Switch Off	Switch off
Output	Light ON	Light Off	Fault
Finish State	ON	OFF	On

- All the possible inputs to the system are enumerated across the columns of the table. All the possible states are enumerated across the rows.
- The above diagram and table shows that if the start state of switch is "OFF" and we change the state of the switch to "ON" the output will be light on and the final state of the switch will be "ON".

Syllabus Topic : Risk Associated with New Technologies

5.9 Risk Associated with New Technologies

Q. 5.9.1 What are the risks associated with new technology? (Ref. Sec. 5.9) (5 Marks)

- New technology brings benefits but it also brings some risks. New technology brings the risks of unknown.
- It is nearly impossible to create software which is unhackable. Hackers can get into the system and steal information. The best way to prevent attacks is to have policies in place to limit damages.
- New threats are emerging every day, with growing technology it is becoming much easier to hack data or get entry into the system.
- So the team should always be working towards protecting their system against these threats. It involves lot of money and time, not all the companies can bear this extra cost.
- Successful implementation of technology relies wholly on your team. Trainings should be provided to team members to help them understand and use technology.

Syllabus Topic : Process Maturity Level of Technology

5.9.1 Process Maturity Level of Technology

Q. 5.9.2 What is Capability Maturity Model? (Ref. Sec. 5.9.1) (5 Marks)

- The Capability Maturity Model (CMM) is a methodology used to develop and refine an organization's software development process.
- CMM can be used to assess an organization against a scale of five process maturity levels.
- Each level ranks the organization according to its standardization of processes in the subject area being assessed.

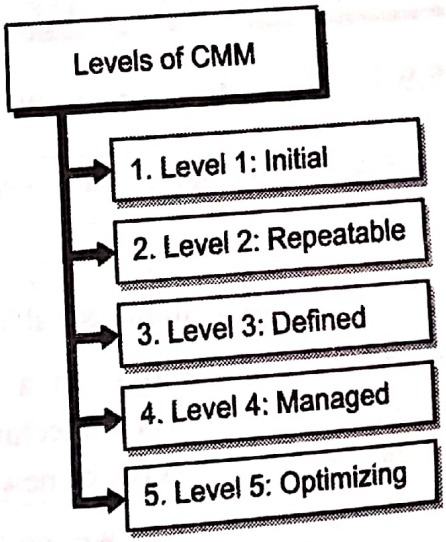

Fig. 5.9.1

Levels of CMM

1. Level 1 : Initial

- Processes are usually ad hoc and the organization usually does not provide a stable environment.
- The competence of the people determines the success of the projects. It is characteristic of processes at this level that they are (typically) undocumented and in a state of dynamic change, tending to be driven in an ad hoc, uncontrolled and reactive manner by users or events.
- This provides a chaotic or unstable environment for the processes.

➜ **2. Level 2 : Repeatable**

Process discipline helps ensure that existing practices are retained during times of stress. When these practices are in place, projects are performed and managed according to their documented plans. Focus is on establishing basic project management policies.

➜ **3. Level 3 : Defined**

- At this level, documentation of the standard guidelines and procedures takes place.
- It is a well defined integrated set of project specific software engineering and management processes.

➜ **4. Level 4 : Managed**

- Using precise measurements, management can effectively control the software development effort.
- In particular, management can identify ways to adjust and adapt the process to particular projects without measurable losses of quality or deviations from specifications.
- At this level organization set a quantitative quality goal for both software process and software maintenance.

➜ **5. Level 5 : Optimizing**

- It is a characteristic of processes at this level that the focus is on continually improving process performance through both incremental and innovative technological changes/improvements.
- Use of new tools, techniques and evaluation of software processes is done to prevent recurrence of known defects.

Syllabus Topic : Testing Adequacy of Control in New Technology Usage

5.9.2 Testing Adequacy of Control in New Technology Usage

Q. 5.9.3 Why is control needed while using new technology? (Ref. Sec. 5.9.2) (5 Marks)

- New technology is simply an advancement of old technology. The impact of technology in modern life is immeasurable.
- We use technology on a daily basis to accomplish specific tasks or interests. New technology or evolved technology at times may replace previously used technology due to its increased benefits or newfound popularity.
- Sometimes the way we implement various technologies do more damage than good hence it is important to control the use of new technology. Management should take proper steps for this control.

Syllabus Topic : Object Oriented Application Testing

5.9.3 Object Oriented Application Testing

Q. 5.9.4 Explain object oriented application testing (Ref. Sec. 5.9.3) (5 Marks)

- Research confirms that testing methods proposed for procedural approach are not adequate for Object Oriented (OO) approach. OO software testing poses additional problems due to the distinguishing characteristics of OO.
- Testing time for OO software found to be increased compared to testing procedural software. In procedural software, unit testing is testing a single module, function, or a procedure.
- In Object oriented software, unit testing is testing a class, integration testing is testing interaction between classes. Methods are usually tested in the context of the class they belong to.

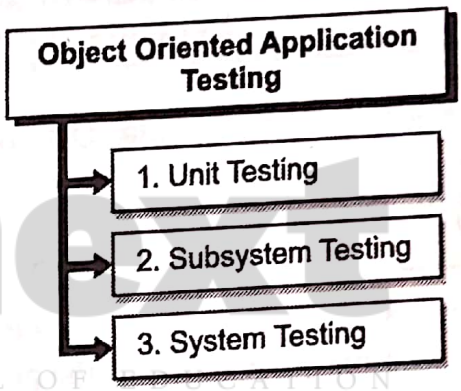

Fig. 5.9.2

1. Unit Testing

- In unit testing, the individual classes are tested. It is seen whether the class attributes are implemented as per design and whether the methods and the interfaces are error-free.
- Unit testing is the responsibility of the application engineer who implements the structure.

2. Subsystem Testing

- This involves testing a particular module or a subsystem and is the responsibility of the subsystem lead.
- It involves testing the associations within the subsystem as well as the interaction of the subsystem with the outside. Subsystem tests can be used as regression tests for each newly released version of the subsystem.

3. System Testing

- System testing involves testing the system as a whole and is the responsibility of the quality-assurance team.
- The team often uses system tests as regression tests when assembling new releases.

Syllabus Topic : Testing of Internal Controls

5.9.4 Testing of Internal Controls

Q. 5.9.5 Explain testing of internal controls. (Ref. Sec. 5.9.4) **(5 Marks)**

- A strong internal control environment is an essential key for ensuring businesses remain thriving for many more years to come.
- It is important to test internal control to :
  - (i) Check the correctness of systems or procedures
  - (ii) Efficiency and effectiveness of internal reporting mechanisms and controls
  - (iii) Appropriateness of the level of responsibility and accountability among staff

Syllabus Topic : Commercial Off-the-Shelf (COTS) Testing

5.9.5 Commercial Off-the-Shelf (COTS) Testing

Q. 5.9.6 Explain the below special tests : Commercial Off-the-Shelf (COTS) Testing. (Ref. Sec. 5.9.5) **(5 Marks)**

- Commercial off-the-shelf products are packaged solutions which are then adapted to satisfy the needs of the purchasing organization.
- However, unlike software products that we can just install and start using right out-of-the-box, these COTS systems must typically undergo configuration, customization and/or extension before they will meet the full business needs of the end-user. This can get expensive.
- When we are testing COTS systems, requirements are the user manuals, the insights come from the vendor and their trainers, and we don't have access to the developers or even experienced users.
- It is a much darker black box that conceals significant risk. Avoid complex combinations of tests and the idea of "testing everything." Instead, base tests on functional or business processes used in the real world environment.
- COTS testing help to identify the quality of product. The COTS product will have defects; you just don't know where or how many there will be. It will also uncover compatibility issues.

Syllabus Topic : Client Server Testing

5.9.6 Client Server Testing

Q. 5.9.7 Explain the below special tests : Client Server Testing. (Ref. Sec. 5.9.6) **(5 Marks)**

- This type of testing is done on 2 tier application generally designed for LAN.
- Application is loaded on server machine while the application exe on every client machine.
- You will test broadly in categories like, GUI on both sides, functionality, Load, client-server interaction, backend. This environment is mostly used in Intranet networks.

Syllabus Topic : Web Application Testing

5.9.7 Web Application Testing

Q. 5.9.8 Explain the below special tests : Web Application Testing. (Ref. Sec. 5.9.7) (5 Marks)

- A Web Application or Web App is Software that runs on a Web Server. Unlike traditional desktop applications, which are launched by a Web Browser.
- Few basic technique of web application testing :

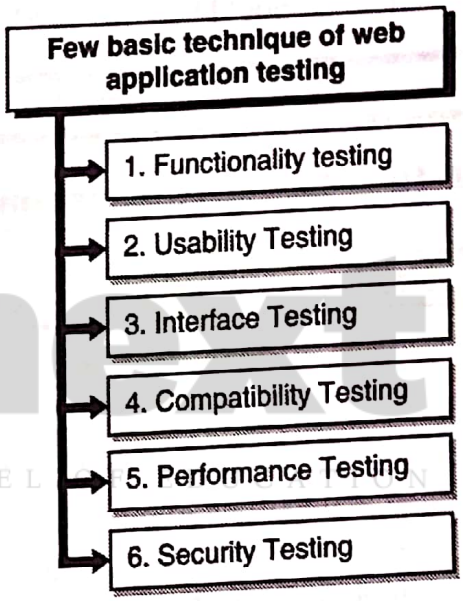

Fig. 5.9.3

➜ **1. Functionality testing**

This is done to test whether the web application designed is as per the requirement. Testing activities include testing all links, images, buttons etc. It also verifies the workflow. Testing cookies (sessions) are deleted either when cache is cleared or when they reach their expiry. Delete cookies (sessions) and test that login credentials are asked for when you next visit the site.

➜ **2. Usability Testing**

This testing checks the navigation and user friendliness of the web pages. Through this testing it is ensured whether the content is properly checked and is easily understandable to the users. It also checks whether the anchor text links are working properly, whether sitemaps and help files are having proper information and all the links are working.

➜ **3. Interface Testing**

This checks if the web server and application server interface, application server and database server interface have proper interaction or not. This test ensures that the users do not see any error messages.

➜ **4. Compatibility Testing**

Compatibility Testing is very important as it checks browser compatibility, operating system compatibility, mobile browsing and printing options.

➜ 5. Performance Testing

Performance testing includes web load testing and web stress testing. Web load testing technique checks if many users can access the same page at the same time and whether a web page can handle heavy load on any specific page. Web stress testing is done on the site to see that how will the site react and recover during the stress time.

➜ 6. Security Testing

This checks the security of the web applications. For security purposes, internal pages should not open if you are not logged into the website. Other statistics should not be seen even if the user is logged in. The files should only be given the option for downloading and it should not be accessed without downloading. CAPTCHA for automates scripts logins should be tested. SSL should be tested for security measures.

Syllabus Topic : Mobile Application Testing

5.10 Mobile Application Testing

Q. 5.10.1 Why is mobile application testing needed? (Ref. Sec. 5.10) (5 Marks)

- Mobile applications are widely used. Due to the ever-growing demand for mobile apps, decision makers are focusing on creating mobile strategies and roadmap before implementing the application for their users.
- It is important to build an app with all features and functionality required by the customer and which is beneficial to the app user, but it is even more critical to have a rigorous mobile app testing plan before deploying it.

5.10.1 Need for Mobile Application Testing

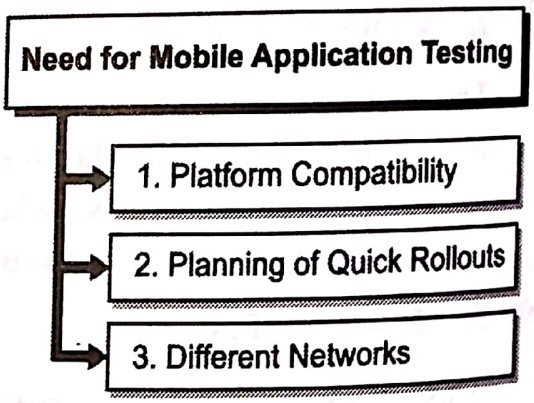

Fig. 5.10.1

➜ 1. Platform Compatibility

- With so many new mobile devices coming up every day like iPhone, iPad, Smart phones, Tablets, Windows Mobile and a wide range of Andriod devices etc, mobile application providers have to provide the multi-platform compatibility to reach their audience.
- There are many combinations while testing mobile apps like Screen resolution, memory sizes, battery, Operating System etc.
- The creation of separate test case and execution on each device can be the most expensive and time-consuming task.

2. Planning of Quick Rollouts

- The companies are looking for the golden business opportunities in unique Mobile apps and expecting a rapid rollout of quality application or improvements and bug fixes if an application is already launched.
- They want to push the applications in the market as quickly as possible to avail the benefits of a market boom-mobile sector.
- As a result, the QA testing cycle which generally takes two to three weeks depends on the complexity and the size of the application is now reduced to half or one week.
- Due to clutch in the timelines the QA it is very difficult to problems if mobile applications don't meet the customer expectations.

3. Different Networks

Almost all applications require internet so this test case needs to run over different connections like WiFi, 3G, 4G etc.

Syllabus Topic : E-Business e-Commerce Testing

5.11 E-Business e-Commerce Testing

Q. 5.11.1 What is E-commerce testing? Explain with the help of an example. (Ref. Sec. 5.11) (5 Marks)

- E-business or e-commerce applications refer to the buying and selling of goods or services using the internet.
- Shopping cart is an important feature of any e-commerce website or app. It allows for customers to select and store multiple items in the cart and purchase them all at once.
- There are some key test cases which should be part of testing a shopping cart.

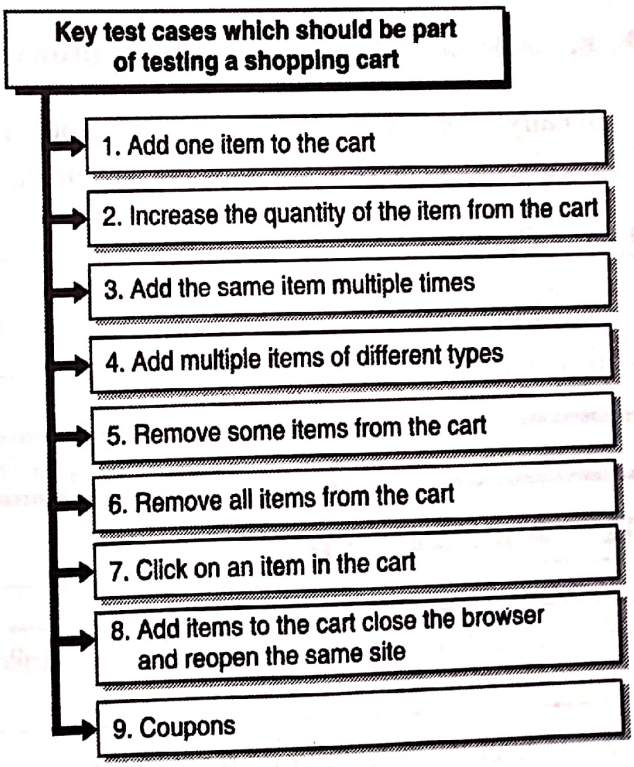

Fig. 5.11.1

1. Add one item to the cart

The cart should be updated with the item with correct name, image, and price.

➜ **2. Increase the quantity of the item from the cart**

The price should be updated to reflect the correct figure.

➜ **3. Add the same item multiple times**

There should be one item in the cart, but the quantity should reflect the number of additions and the total price should reflect the sum of the price of each item.

➜ **4. Add multiple items of different types**

For each item added, we should see a corresponding name, image, and price and total price of all items.

➜ **5. Remove some items from the cart**

The cart should update showing the existing items in the cart, total price should reflect the new sum.

➜ **6. Remove all items from the cart**

Cart balance should be zero, no items should be displayed in the cart.

➜ **7. Click on an item in the cart**

We should be able to see more information about the product we just clicked either as a popup or redirecting to the product page.

➜ **8. Add items to the cart close the browser and reopen the same site**

Ideally, the cart should still hold your items. N.B this particularly depends on the requirements on how the cart should behave.

➜ **9. Coupons**

Need to check that the price of the cart is discounted when we apply a coupon and not discounted when we apply an invalid or expired coupon.

Syllabus Topic : Agile Development Testing

5.12 Agile Development Testing

Q. 5.12.1 Explain the below special tests : Agile Development Testing. (Ref. Sec. 5.12) **(5 Marks)**

– Agile is an iterative and incremental model. Each story is expanded, coded and tested. A Possible release occurs after each iteration.

- Agile Testing involves a cross-functional Agile team actively relying on the special expertise contributed by testers. This allows the combined team to better meet the project's defined business, software usability, quality, and timeline objectives.
- The combined team, including both development and testing, takes responsibility for analyzing the business specifications.
- Together, they define the Sprint goal. The QA team defines the testing scope (i.e., test plan). That is then validated and approved by the entire team and the client. Simultaneously, while the development team starts the implementation of modules (in the very first Sprint), the QA team begins work on the test case design.
- These are properly documented either in a testing tool or in an Excel spreadsheet that is handed over to the development team and project sponsor from the business side to review. This is to ensure that test coverage is as complete as possible.
- Once the test case review and any modifications are completed for a particular Sprint, the QA team then begins testing on the QA environment. Defects found during testing are logged properly in a defect tracking tool.
- Depending on the severity and priority of defects, fixing them can be delayed but then is taken care of in upcoming Sprints. At the end of each Sprint, the team determines, along with the project sponsor, which defects are to be fixed in the current iteration.
- This iteration continues until all planned Sprints are completed. QA, along with the development team and business organization, defines which main flows (test cases) will be automated.
- When code is ready to test (after the end of each Sprint), QA works with development to execute test cases on the development environment, in order to identify the early stage defects so developers can fix them during the next round, on a priority basis.
- This process is then repeated throughout the development process. Automated test cases are run daily throughout the SDLC.
- White-board/stand-up meetings are conducted daily involving members of all teams associate ted with product development, support and testing. This helps to resolve the issues faced by team members and provides a clear picture of progress in both the coding and testing areas.
- Agile promotes the introduction of requirements at all stages/iterations of the SDLC; however, the testing team determines when to end this process to ensure product stability.

Syllabus Topic : Data Warehousing Testing

5.13 Data Warehousing Testing

Q. 5.13.1 What is warehouse testing? (Ref. Sec. 5.13) **(5 Marks)**

- Testing is undoubtedly an essential part of Data Warehouse life-cycle but cycle but it received a few attentions with respect to other phases.
- Software testing is predominantly focused on program code, while Software testing is predominantly focused on program code, while DW testing is directed at data and information.
- DW testing focuses on the correctness and usefulness of the information delivered to users. Differently from generic software systems, DW testing involves a huge data volume, which significantly impacts performance and productivity.
- It is aimed at supporting any views of data, so the possible number of use scenarios is practically infinite and only few of them are known from the beginning.
- It is almost impossible to predict a It is almost impossible to predict all the possible types of errors that will be encountered in real operational data.
- One of the main difficulties in testing the DW systems is that DW systems are different in different organizations.
- Each organization has its own DW system that conforms with its own requirements and needs, which leads to differences between DW systems in several aspects (such as database technology, tools used, size, number of users, number of data sources, how the components are connected, etc.)
- Another big challenge that is faced by the DW testers is regarding the test data preparation. Making use of real data for testing purpose is a violation of citizen's privacy laws in some countries (for example, using real data of bank accounts and other information is illegal in many countries).
- For a proper testing of a DW, large amount of test data is necessary. In real-time environment, the system may behave differently in presence of terabytes of data.

5.14 Exam Pack (Review Questions)

☞ **Syllabus Topic : Levels of Testing : Introduction**

Q. 1 Explain levels of testing. *(Ans. : Refer section 5.1.1)* **(5 Marks)**

☞ **Syllabus Topic : Proposal and Requirement Testing**

Q. 2 What is proposal and requirement testing? *(Ans. : Refer section 5.1.2)* **(5 Marks)**

☞ **Syllabus Topic : Code Review**

Q. 3 Explain code review. *(Ans. : Refer section 5.1.3)* **(5 Marks)**

☞ **Syllabus Topic : Design Testing**

Q. 4 What is design testing? *(Ans. : Refer section 5.1.4)* **(5 Marks)**

Syllabus Topic : Testing Stages

Q. 5 List and explain software testing strategies. *(Ans. : Refer section 5.2)* **(5 Marks)**

Syllabus Topic : Unit Testing

Q. 6 Explain Unit testing. *(Ans. : Refer section 5.2.1)* **(5 Marks)**

Syllabus Topic : Module Testing

Q. 7 What is module testing? *(Ans. : Refer section 5.2.2)* **(5 Marks)**

Syllabus Topic : Integration Testing, Big Bang Testing, Sandwich Testing

Q. 8 What is integration testing ? *(Ans. : Refer section 5.2.3)* **(5 Marks)**

Syllabus Topic : Critical Path First

Q. 9 Explain critical path first method. *(Ans. : Refer section 5.2.5)* **(5 Marks)**

Syllabus Topic : Sub System Testing

Q. 10 Explain sub-system testing. *(Ans. : Refer section 5.2.6)* **(5 Marks)**

Syllabus Topic : System Testing

Q. 11 What is system testing? *(Ans. : Refer section 5.2.7)* **(5 Marks)**

Syllabus Topic : Special Tests : Introduction, GUI Testing, Compatibility Testing

Q. 12 Explain the below special tests : GUI Testing. *(Ans. : Refer section 5.3)* **(5 Marks)**

Q. 13 Explain the below special tests : Compatibility Testing. *(Ans. : Refer section 5.3)* **(5 Marks)**

Syllabus Topic : Security Testing

Q. 14 Explain the below special tests : Security Testing. *(Ans. : Refer section 5.4)* **(5 Marks)**

Syllabus Topic : Performance Testing, Volume Testing, Stress Testing

Q. 15 Explain the below special tests : Performance Testing. *(Ans. : Refer section 5.5)* **(5 Marks)**

Q. 16 Explain the below special tests : Volume Testing. *(Ans. : Refer section 5.5)* **(5 Marks)**

Q. 17 Explain the below special tests : Stress Testing. *(Ans. : Refer section 5.5)* **(5 Marks)**

Syllabus Topic : Recovery Testing

Q. 18 Explain the below special tests : Recovery Testing. *(Ans. : Refer section 5.6)* **(5 Marks)**

☞ **Syllabus Topic : Installation Testing**

Q. 19 Explain the below special tests : Installation Testing. **(5 Marks)**
(Ans. : Refer section 5.6.1)

☞ **Syllabus Topic : Requirement Testing**

Q. 20 Explain the below special tests : Requirement Testing. **(5 Marks)**
(Ans. : Refer section 5.6.2)

☞ **Syllabus Topic : Regression Testing**

Q. 21 Explain the below special tests : Regression Testing. **(5 Marks)**
(Ans. : Refer section 5.6.3)

☞ **Syllabus Topic : Error Handling Testing**

Q. 22 Explain the below special tests : Error Handling Testing. **(5 Marks)**
(Ans. : Refer section 5.6.4)

☞ **Syllabus Topic : Manual Support Testing**

Q. 23 Explain the below special tests : Manual support Testing. **(5 Marks)**
(Ans. : Refer section 5.6.5)

☞ **Syllabus Topic : Intersystem Testing**

Q. 24 Define Intersystem testing. *(Ans. : Refer section 5.6.6)* **(5 Marks)**

☞ **Syllabus Topic : Control Testing**

Q. 25 Define Control testing. *(Ans. : Refer section 5.6.7)* **(5 Marks)**

☞ **Syllabus Topic : Smoke Testing**

Q. 26 Explain the below special tests : Smoke Testing. *(Ans. : Refer section 5.6.8)* **(5 Marks)**

☞ **Syllabus Topic : Adhoc Testing**

Q. 27 Explain the below special tests : Adhoc Testing. *(Ans. : Refer section 5.6.9)* **(5 Marks)**

☞ **Syllabus Topic : Parallel Testing**

Q. 28 What is parallel testing? *(Ans. : Refer section 5.6.10)* **(5 Marks)**

☞ **Syllabus Topic : Execution Testing**

Q. 29 Define execution testing. *(Ans. : Refer section 5.6.11)* **(5 Marks)**

☞ **Syllabus Topic : Operations Testing**

Q. 30 What is operation testing? *(Ans. : Refer section 5.6.12)* **(5 Marks)**

Syllabus Topic : Compliance Testing

Q. 31 Explain the below special tests : Compliance Testing. *(Ans. : Refer section 5.6.13)* **(5 Marks)**

Syllabus Topic : Usability Testing

Q. 32 Explain the below special tests : Usability Testing. *(Ans. : Refer section 5.6.14)* **(5 Marks)**

Syllabus Topic : Decision Table Testing

Q. 33 Explain the below special tests : Decision Table Testing. *(Ans. : Refer section 5.6.15)* **(5 Marks)**

Syllabus Topic : Documentation Testing

Q. 34 Define documentation testing. *(Ans. : Refer section 5.6.16)* **(5 Marks)**

Syllabus Topic : Training Testing

Q. 35 Define training testing. *(Ans. : Refer section 5.6.17)* **(5 Marks)**

Syllabus Topic : Rapid Testing

Q. 36 Explain the below special tests : Rapid Testing. *(Ans. : Refer section 5.6.18)* **(5 Marks)**

Syllabus Topic : Control Flow Graph

Q. 37 Explain control flow graph. *(Ans. : Refer section 5.7)* **(5 Marks)**

Syllabus Topic : Generating Tests on the Basis of Combinatorial Designs

Q. 38 Explain generating test cases on the basis of combinatorial designs. *(Ans. : Refer section 5.8)* **(5 Marks)**

Syllabus Topic : State Graphs

Q. 39 Explain State Transition diagram. *(Ans. : Refer section 5.8.1)* **(5 Marks)**

Syllabus Topic : Risk Associated with New Technologies

Q. 40 What are the risks associated with new technology? *(Ans. : Refer section 5.9)* **(5 Marks)**

Syllabus Topic : Process Maturity Level of Technology

Q. 41 What is Capability Maturity Model? *(Ans. : Refer section 5.9.1)* **(5 Marks)**

Syllabus Topic : Testing Adequacy of Control in New Technology Usage

Q. 42 Why is control needed while using new technology? *(Ans. : Refer section 5.9.2)* **(5 Marks)**

☞ **Syllabus Topic : Object Oriented Application Testing**

Q. 43 Explain object oriented application testing. *(Ans. : Refer section 5.9.3)* **(5 Marks)**

☞ **Syllabus Topic : Testing of Internal Controls**

Q. 44 Explain testing of internal controls. *(Ans. : Refer section 5.9.4)* **(5 Marks)**

☞ **Syllabus Topic : Commercial Off-the-Shelf (COTS) Testing**

Q. 45 Explain the below special tests : Commercial Off-the-Shelf (COTS) Testing. *(Ans. : Refer section 5.9.5)* **(5 Marks)**

☞ **Syllabus Topic : Client Server Testing**

Q. 46 Explain the below special tests : Client Server Testing. *(Ans. : Refer section 5.9.6)* **(5 Marks)**

☞ **Syllabus Topic : Web Application Testing**

Q. 47 Explain the below special tests : Web Application Testing. *(Ans. : Refer section 5.9.7)* **(5 Marks)**

☞ **Syllabus Topic : Mobile Application Testing**

Q. 48 Why is mobile application testing needed? *(Ans. : Refer section 5.10)* **(5 Marks)**

☞ **Syllabus Topic : E-Business e-Commerce Testing**

Q. 49 What is E-commerce testing? Explain with the help of an example. *(Ans. : Refer section 5.11)* **(5 Marks)**

☞ **Syllabus Topic : Agile Development Testing**

Q. 50 Explain the below special tests : Agile Development Testing. *(Ans. : Refer section 5.12)* **(5 Marks)**

☞ **Syllabus Topic : Data Warehousing Testing**

Q. 51 What is warehouse testing? *(Ans. : Refer section 5.13)* **(5 Marks)**