MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION
(Autonomous)
(ISO/IEC - 27001 - 2013 Certified)

SUMMER – 19 EXAMINATION

Subject Name: Software Testing **Model Answer** **Subject Code: 17624**

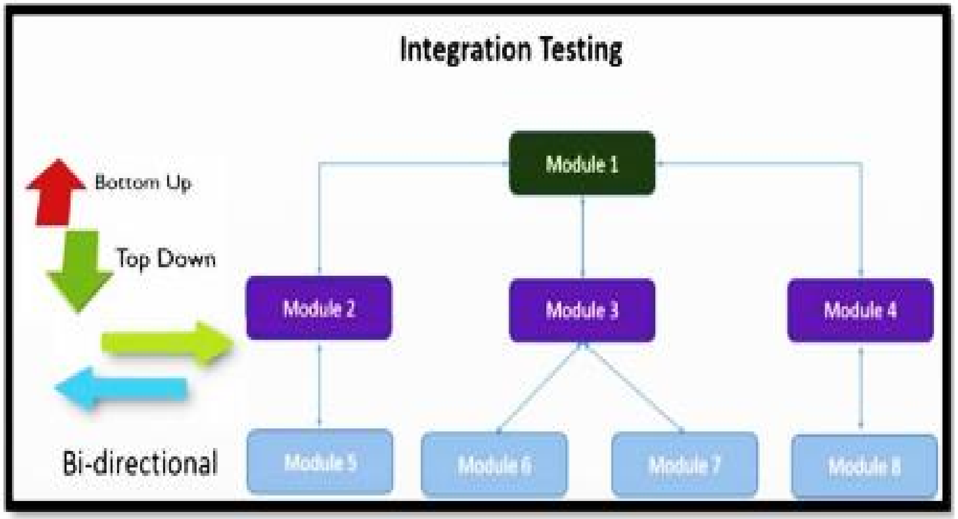
Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No.	Sub Q. N	Answer	Marking Scheme
1		**Attempt any FIVE of the following:**	**20 M**
	a	**Define software testing and role of testing.**	
	Ans	**Software testing:** • Software testing is defined as performing Verification and Validation of the Software Product for its correctness and accuracy of working. • Software Testing is the process of executing a program with the intent of finding errors. • A successful test is one that uncovers an as-yet-undiscovered error. • Testing can show the presence of bugs but never their absence. • Testing is a support function that helps developers look good by finding their mistakes before anyone else does. **Role of testing:** 1. Finding defects which may get created by the programmer while developing the software. 2. Gaining confidence in and providing information about the level of quality. 3. To prevent defects. 4. To make sure that the end result meets the business and user requirements. 5. To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications. 6. To gain the confidence of the customers by providing them a quality	2 M for Definition and 2 M for Roles **OR** Answer with Relevant Contents

		product.	
	b	**Explain inspection and walk through.**	
	Ans	**Inspection:** i. Inspections are the most formal type of reviews. ii. They are highly structured and require training for each participant. iii. Inspections are different from peer reviews and walkthroughs in that the person who presents the code, the presenter or reader, isn't the original programmer. iv. These forces someone else to learn and understand the material being presented, potentially giving a different slant and interpretation at the inspection meeting. v. The other participants are called inspectors. vi. Each is tasked with reviewing the code from a different perspective, such as a User, a tester, or a product support person. vii. This helps bring different views of the product under review and very often Identifies different bugs. viii. One inspector is even tasked with reviewing the code backward—that is, from the end to the beginning—to make sure that the material is covered evenly and completely. **Walkthrough:** i. Walkthroughs are the next step up in formality from peer reviews. ii. In a walkthrough, the programmer who wrote the code formally presents (Walks through) it to a small group of five or so other programmers and testers. iii. The reviewers should receive copies of the software in advance of the review so they can examine it and write comments and questions that they want to ask at the review. iv. Having at least one senior programmer as a reviewer is very important.	2 M for Inspection and 2 M for Walkthrough **OR** Answer with Relevant Contents
	c	**Explain bi-directional integration**	
	Ans		2 M for diagram, 2 M for explanation **OR** Answer with

<table>
<tr>
<td></td>
<td></td>
<td>

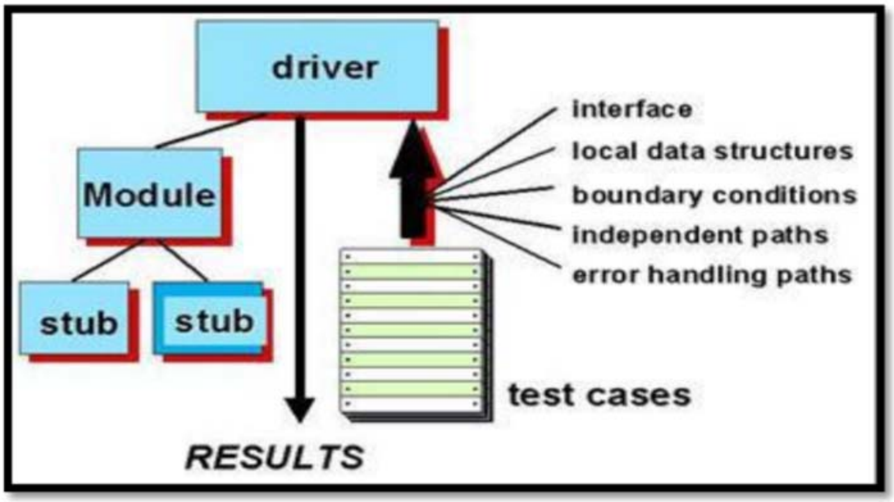


1. Bi-directional Integration is a kind of integration testing process that combines top-down and bottom-up testing.
2. With an experience in delivering Bi-directional testing projects custom software development services provide the best quality of the deliverables right from the development of software process.
3. Bi-directional Integration testing is a vertical incremental testing strategy that tests the bottom layers and top layers and tests the integrated system in the computer software development process.
4. Using stubs, it tests the user interface in isolation as well as tests the very lowest level functions using drivers.
5. Bi-directional Integration testing combines bottom-up and top-down testing.
6. Bottom-up testing is a process where lower level modules are integrated and then tested.
7. This process is repeated until the component of the top of the hierarchy is analyzed. It helps custom software development services find bugs easily without any problems.
8. Top down testing is a process where the top integrated modules are tested and the procedure is continued till the end of the related module.
9. Top down testing helps developers find the missing branch link easily

</td>
<td>Relevant Contents</td>
</tr>
<tr>
<td></td>
<td><b>d</b></td>
<td><b>Explain how a test plan is prepared</b></td>
<td></td>
</tr>
<tr>
<td></td>
<td><b>Ans</b></td>
<td>The test plan acts as the anchor for the execution, tracking and reporting of the entire testing project and covers.</td>
<td>4 M for correct explanation</td>
</tr>
</table>

		Preparing test plan: • What needs to be tested – the scope of testing, including clear identification of what will be the tested & what will not be tested. • How the testing is going to be performed – breaking down the testing into small and manageable tasks and identifying the strategies to be used for carrying out the tasks. • What resources are needed for testing- computer as well as human resources. • The time lines by which the testing activities will be performed. Risks that may be faced in all of the above, with appropriate mitigation and contingency plans.	**OR** Answer with Relevant Contents
	e	**Explain the process, how the bug is reported.**	
	Ans	A bug report documents an anomaly discovered during testing. It includes all the information needed to reproduce the problem, including the author, release/build number, open/close dates, problem area, problem description, test environment, bug type, how it was detected, who detected it, priority, severity, status, etc. After uncovering a bug, testers generate a formal bug report. The purpose of a bug report is to state the problem as clearly as possible so that developers can replicate the bug easily and fix it While reporting the bug to developer, your Bug Report should contain the following information • Defect ID - Unique identification number for the defect. • Defect Description - Detailed description of the Defect including information about the module in which Defect was found. • Version - Version of the application in which defect was found. • Steps - Detailed steps along with screenshots with which the developer can reproduce the defects. • Date Raised - Date when the defect is raised • Reference- where in you Provide reference to the documents like requirements, design, architecture or maybe even screenshots of the error to help understand the defect • Detected By - Name/ID of the tester who raised the defect • Status - Status of the defect , more on this later • Fixed by - Name/ID of the developer who fixed it • Date Closed - Date when the defect is closed • Severity which describes the impact of the defect on the application • Priority which is related to defect fixing urgency. Severity Priority could be High/Medium/Low based on the impact urgency at which the defect should be fixed respectively	4 M for correct explanation **OR** Answer with Relevant Contents

	f	**Explain criteria to select testing tool.**	
	Ans	**Criteria for Selecting Test Tools:** The Categories for selecting Test Tools are, 1. Meeting requirements; 2. Technology expectations; 3. Training/skills; 4. Management aspects. **1. Meeting requirements-** There are plenty of tools available in the market but rarely do they meet all the requirements of a given product or a given organization. Evaluating different tools for different requirements involve significant effort, money, and time. Given of the plethora of choice available, huge delay is involved in selecting and implementing test tools. **2. Technology expectations-** Test tools in general may not allow test developers to extends/modify the functionality of the framework. So extending the functionality requires going back to the tool vendor and involves additional cost and effort. A good number of test tools require their libraries to be linked with product binaries. **3. Training/skills-** While test tools require plenty of training, very few vendors provide the training to the required level. Organization level training is needed to deploy the test tools, as the user of the test suite are not only the test team but also the development team and other areas like configuration management. **4. Management aspects-** A test tool increases the system requirement and requires the hardware and software to be upgraded. This increases the cost of the already- expensive test tool. **OR** **Guidelines for selecting a tool:** 1. The tool must match its intended use. Wrong selection of a tool can lead to problems like lower efficiency and effectiveness of testing may be lost. 2. Different phases of a life cycle have different quality-factor requirements. Tools required at each stage may differ significantly. 3. Matching a tool with the skills of testers is also essential. If the testers	1M for each factor **OR** Answer with Relevant Contents

		do not have proper training and skill then they may not be able to work effectively. 4. Select affordable tools. Cost and benefits of various tools must be compared before making final decision. 5. Backdoor entry of tools must be prevented. Unauthorized entry results into failure of tool and creates a negative environment for new tool introduction.	
2		**Attempt any FIVE of the following:**	**16 M**
	a	**Explain the need of stubs and drivers with diagram and its importance in software testing.**	
	Ans	**Ans:** **Drivers:** The module where the required inputs for the module under test are simulated for the purpose of module or unit testing is known as a Driver module. The driver module may print or interpret the result produced by the module under test. **Stubs:** The module under testing may also call some other module which is not ready at the time of testing. There is need of dummy modules required to simulate for testing, instead of actual modules. These are called stubs.  driver Module stub stub interface local data structures boundary conditions independent paths error handling paths test cases RESULTS For example if a developer is developing a loop for searching functionality of an application which is a very small unit of the whole code of that application then to verify that the particular loop is working properly or not is known as unit testing.	4 M for need of stubs and drivers, 2 M for diagram, 2 M for importance **OR** Answer with Relevant Contents

Importance:

- Stubs and Drivers works as a substitute for the missing or unavailable module.
- They are specifically developed, for each module, having different functionalities.
- Generally, developers and unit testers are involved in the development of stubs and drivers.
- Their most common use may be seen in the integration incremental testing, where stubs are used in top bottom approach and drivers in a bottom up approach.

b **Explain code functional testing and code coverage testing with example**

Ans **Code Functional Testing**:

i. Code Functional Testing involves tracking a piece of data completely through the software.
ii. At the unit test level this would just be through an individual module or function.
iii. The same tracking could be done through several integrated modules or even through the entire software product—although it would be more time consuming to do so.
iv. During data flow, the check is made for the proper declaration of variables declared and the loops used are declared and used properly.

For example

```
#include<stdio.h>
void main()
{
int i , fact= 1, n;
printf("Enter the number:");
scanf("%d", &n);
for(i =1; i<=n; i++)
fact = fact * i;
printf("The factorial of a number is: "%d", fact);
}
```

Code Coverage Testing:

i. The logical approach is to divide the code just as you did in black-box testing into its data and its states (or program flow).
ii. By looking at the software from the same perspective, you can more easily map the white-box information you gain to the

4 M for Code Functional Testing and 4 M for Code Coverage Testing

OR

Answer with Relevant Contents

black-box cases you‘ve already written.

iii. Consider the data first. Data includes all the variables, constants, arrays, data structures, keyboard and mouse input, files and screen input and output, and I/O to other devices such as modems, networks, and so on.

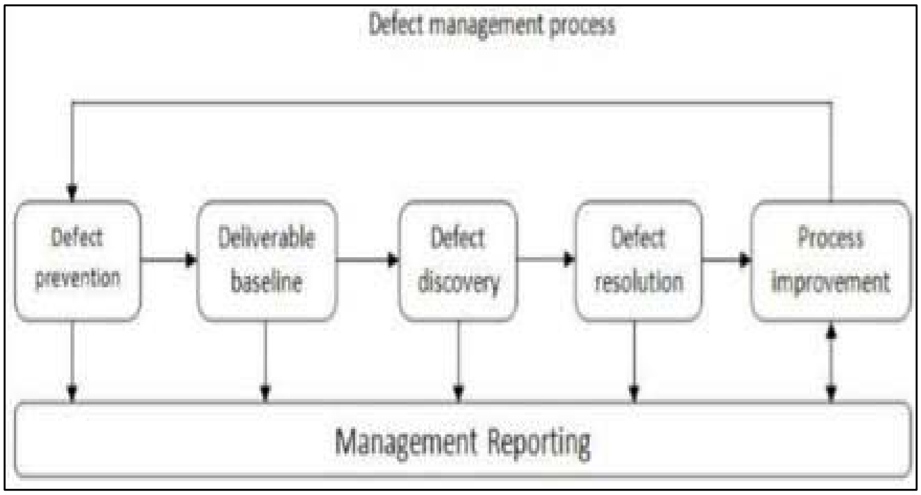
For example

```
#include<stdio.h>
void main()
{
int i , fact= 1, n;
printf("Enter the number:");
scanf("%d",&n);
for(i =1; i<=n; i++)
fact = fact * i;
printf("The factorial of a number is: "%d", fact);
}
```

The declaration of data is complete with the assignment statement and the variable declaration statements. All the variable declared are properly utilized.

c **Explain Defect Management Process with suitable diagram**

Ans



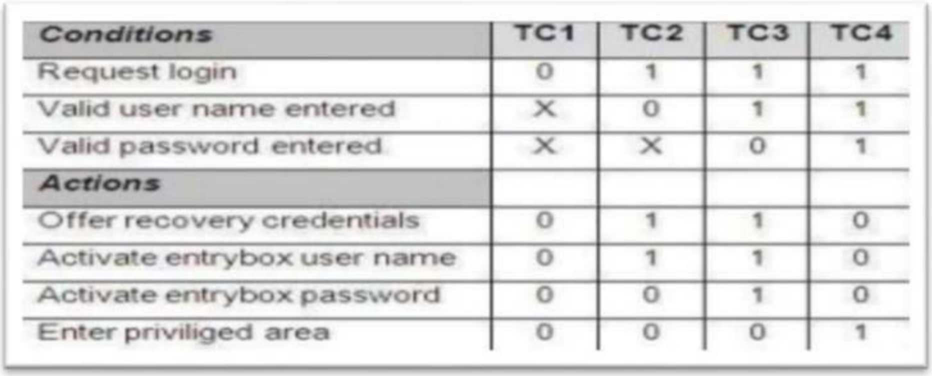
1. **Defect Prevention:** Implementation of techniques, methodology and standard processes to reduce the risk of defects.
2. **Deliverable Baseline:** Deliverables are considered to be ready for further development. i.e. the deliverables meet exit criteria.
3. **Defect Discovery:** To find the defect through the process of

2 M for diagram, 1 M for each point

OR

Answer with Relevant Contents

		verification and validation. 4. **Defect Resolution:** Defect is corrected or corrective action is taken and notification is given to tester. 5. **Process Improvement:** To identify ways to improve the process to prevent further future occurrences of similar defects i.e. Corrective and preventive action is taken for processes improvement. 6. **Management Reporting:** Reporting is about status of application and processes.	
3		**Attempt any Four of the following:**	**16 M**
	a	**Explain when to start and stop testing.**	
	Ans	Process model is a way to represent any given phase of software development that prevent and minimize the delay between defect injection and defect detection/correction. • **Entry criteria**, specifies when that phase can be started also included the inputs for the phase. • Tasks or steps that need to be carried out in that phase, along with measurements that characterize the tasks. • Verification, which specifies methods of checking that tasks have been carried out correctly. • Clear entry criteria make sure that a given phase does not start prematurely. • The verification for each phase helps to prevent defects. At least defects can be minimized. Exit criteria, which stipulate the conditions under which one can consider the phases as done and included are the outputs for the phase. **Exit criteria** may include: 1. All test plans have been run 2. All requirements coverage has been achieved. 3. All severe bugs are resolved.	Start testing 2 marks & End testing :2 marks **OR** Answer with Relevant Contents
	b	**Explain the importance of decision table in Testing**	
	Ans		2 M –table , 2 M for explanation **OR** Answer with Relevant Contents



Conditions	TC1	TC2	TC3	TC4
Request login	0	1	1	1
Valid user name entered	X	0	1	1
Valid password entered	X	X	0	1
Actions				
Offer recovery credentials	0	1	1	0
Activate entrybox user name	0	1	1	0
Activate entrybox password	0	0	1	0
Enter priviliged area	0	0	0	1

I. Decision table testing is black box test design technique to determine the test scenarios for complex business logic.

II. Decision tables provide a systematic way of stating complex business rules, which is useful for developers as well as for testers.

III. Decision tables can be used in test design whether or not they are used in specifications, as they help testers explore the effects of combinations of different inputs and other software states that must correctly implement business rules.

IV. It helps the developers to do a better job can also lead to better relationships with them.

V. Testing combinations can be a challenge, as the number of combinations can often be huge.

VI. Testing all combinations may be impractical if not impossible.

VII. We have to be satisfied with testing just a small subset of combinations but making the choice of which combinations to test and which to leave out is also important.

VIII. If you do not have a systematic way of selecting combinations, an arbitrary subset will be used and this may well result in an ineffective test effort.

Importance of Decision Table: Essentially it is a structured exercise to formulate requirements when dealing with complex business rules. Decision tables are used to model complicated logic. They can make it easy to see that all possible combinations of conditions have been considered and when conditions are missed, it is easy to see

	c	**Explain GUI testing with example**	
	Ans	**GUI Testing:** i. GUI testing is a testing technique in which the application's user interface is tested whether the application performs as expected with respect to user interface behavior. **ii.** GUI Testing includes the application behavior towards keyboard	Explanation of GUI Testing: 4 M **OR** Answer with Relevant

and mouse movements and how different GUI objects such as toolbars, buttons, menu bars, dialog boxes, edit fields, lists, behaviour to the user input.

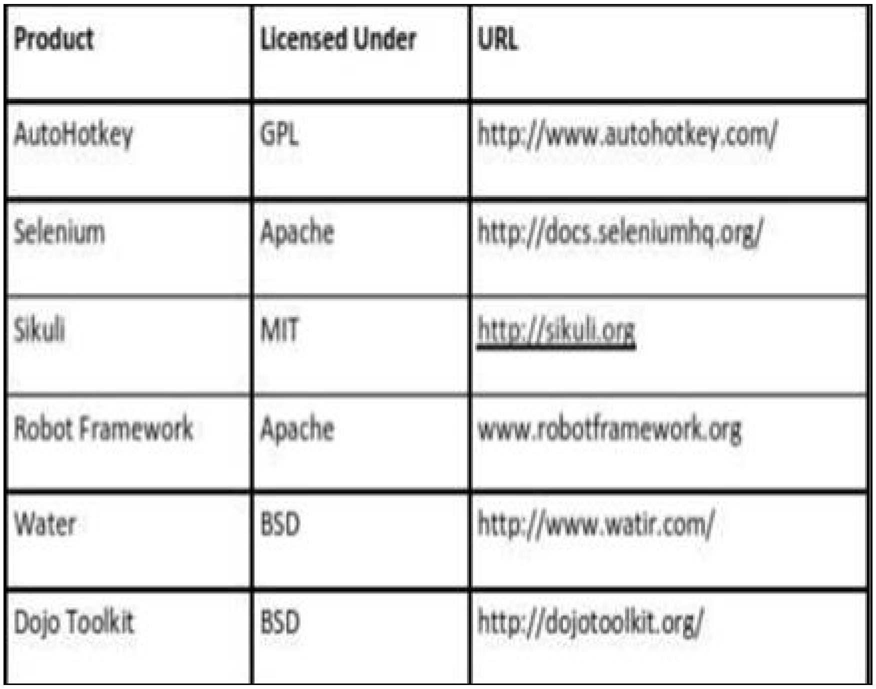
Contents

GUI Testing Guidelines

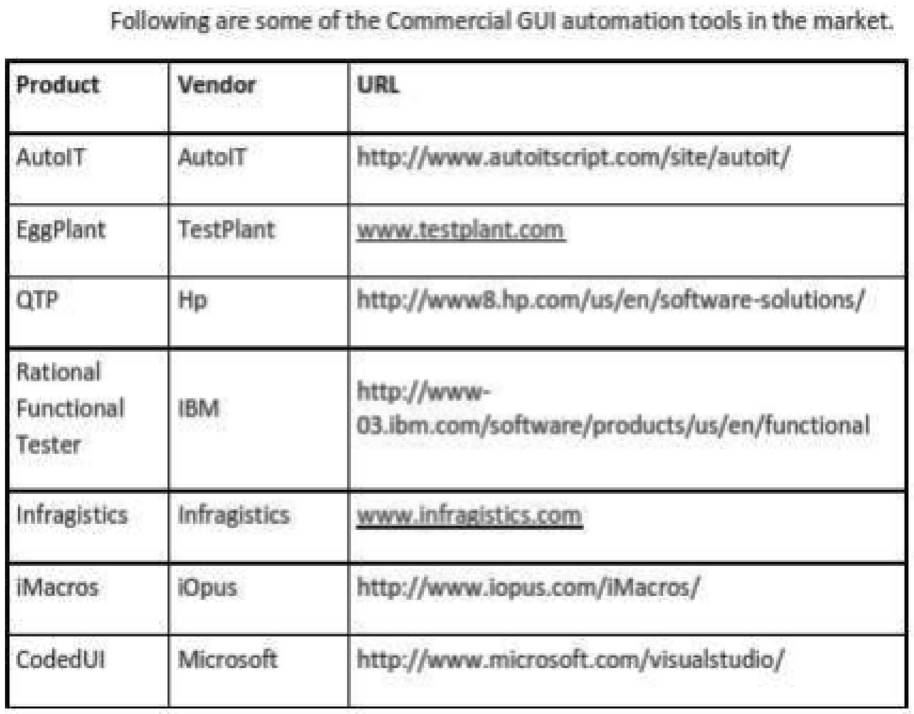
- **i.** Check Screen Validations
- **ii.** Verify All Navigations
- **iii.** Check usability Conditions
- **iv.** Verify Data Integrity
- **v.** Verify the object states
- **vi.** Verify the date Field and Numeric Field Formats

GUI Automation Tools

Following are some of the open source GUI automation tools in the market:



Product	Licensed Under	URL
AutoHotkey	GPL	http://www.autohotkey.com/
Selenium	Apache	http://docs.seleniumhq.org/
Sikuli	MIT	http://sikuli.org
Robot Framework	Apache	www.robotframework.org
Water	BSD	http://www.watir.com/
Dojo Toolkit	BSD	http://dojotoolkit.org/



Following are some of the Commercial GUI automation tools in the market.

Product	Vendor	URL
AutoIT	AutoIT	http://www.autoitscript.com/site/autoit/
EggPlant	TestPlant	www.testplant.com
QTP	Hp	http://www8.hp.com/us/en/software-solutions/
Rational Functional Tester	IBM	http://www-03.ibm.com/software/products/us/en/functional
Infragistics	Infragistics	www.infragistics.com
iMacros	iOpus	http://www.iopus.com/iMacros/
CodedUI	Microsoft	http://www.microsoft.com/visualstudio/

d **Explain the test infrastructure management with its components.**

Ans Testing requires a robust infrastructure to be planned upfront. This infrastructure is made up of three essential elements.

Listing of component : 2 M, Explanation: 2 M

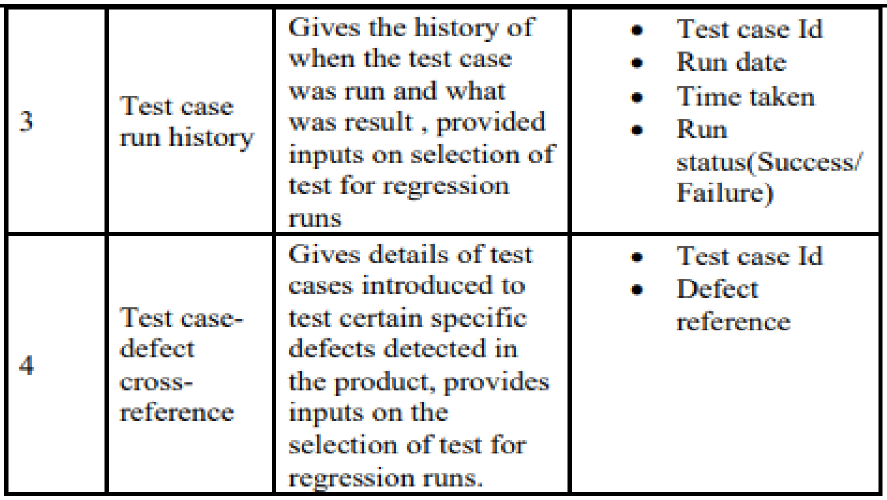
OR

Answer with Relevant Contents

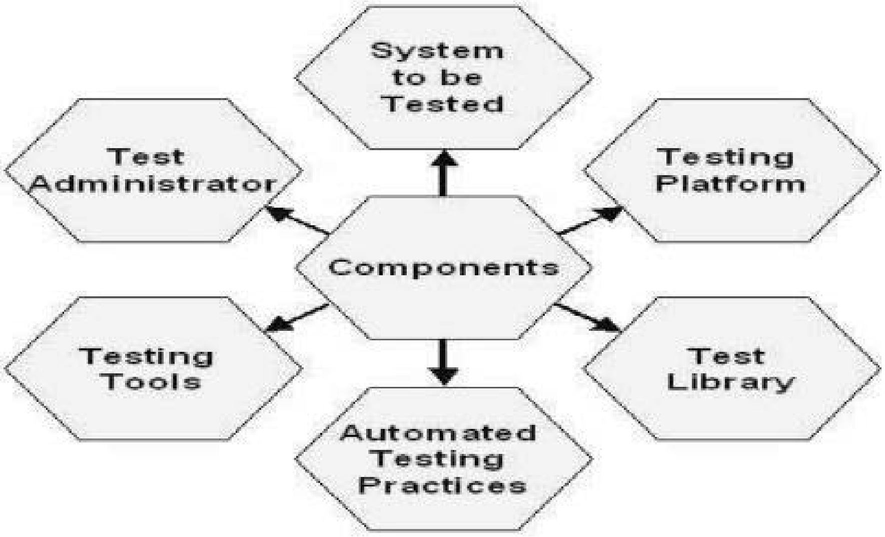
- A test case database (TCDB) (additional): A test case database captures all the relevant information about the test cases in an organization. Some of the entities and the attributes are given in following table

Sr. No.	Test Case	Purpose	Attributes
1	Test case	Records all static information about tests.	• Test case Id • Test case name (File name) • Test case owner • Associated files for test case.
2	Test case product cross reference	Provide mapping between the tests and the corresponding product features, enables identification of test cases for given feature.	• Test case Id • Module Id

-



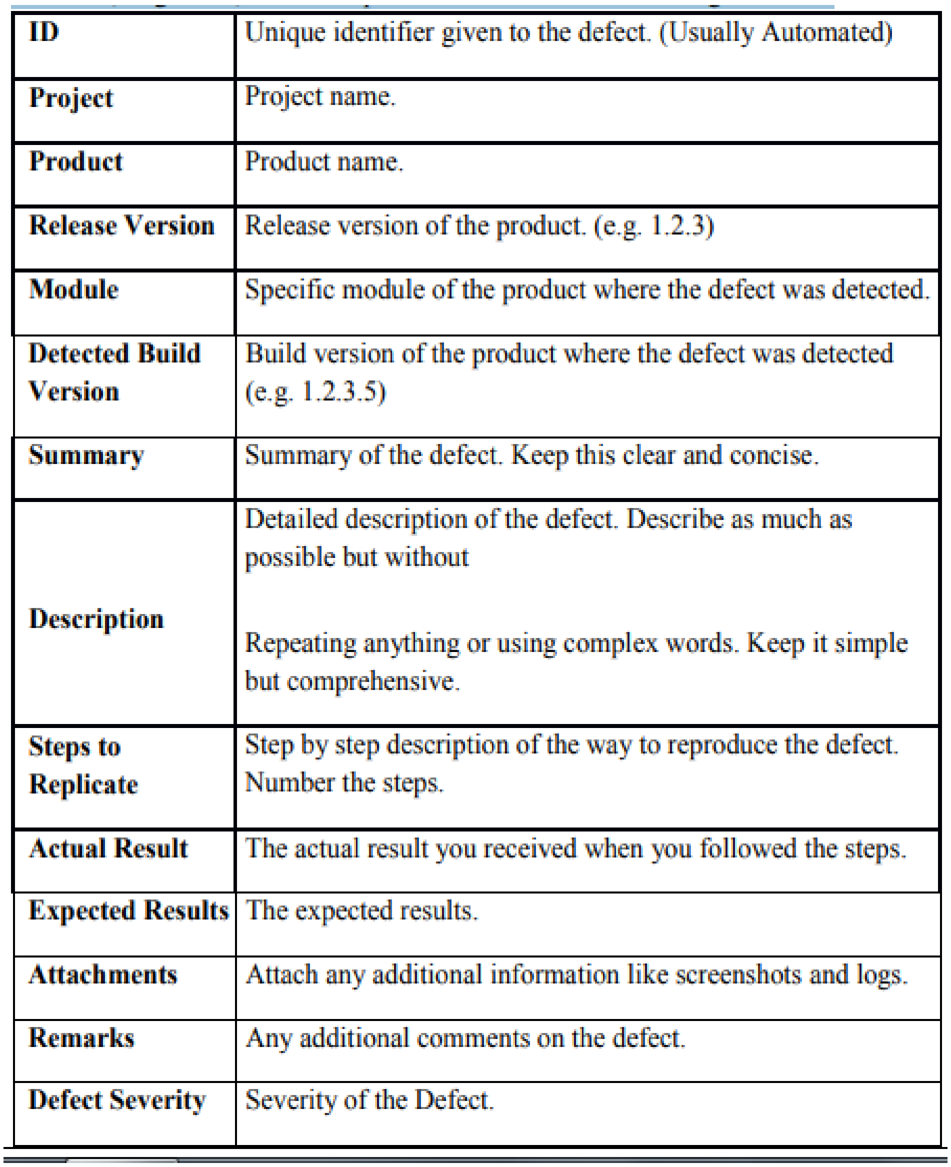
3	Test case run history	Gives the history of when the test case was run and what was result , provided inputs on selection of test for regression runs	• Test case Id • Run date • Time taken • Run status(Success/ Failure)
4	Test case-defect cross-reference	Gives details of test cases introduced to test certain specific defects detected in the product, provides inputs on the selection of test for regression runs.	• Test case Id • Defect reference



Test Infrastructure

	e	**Describe different template with its attribute**	
	Ans	Defect template: A defect report documents an anomaly discovered during testing. It includes all the information needed to reproduce the problem, including the author, release/build number, open/close dates, problem area, problem description, test environment, defect type, how it was detected, who detected it, priority, severity, status, etc. After uncovering a defect (bug), testers generate a formal defect report. The purpose of a defect report is to state the problem as clearly as possible so that developers can replicate the defect easily and fix it. **DEFECT REPORT TEMPLATE** In most companies, a defect reporting tool is used and the elements of a	Description 4 m **OR** Answer with Relevant Contents

report can vary. However, in general, a defect report can consist of the following elements.

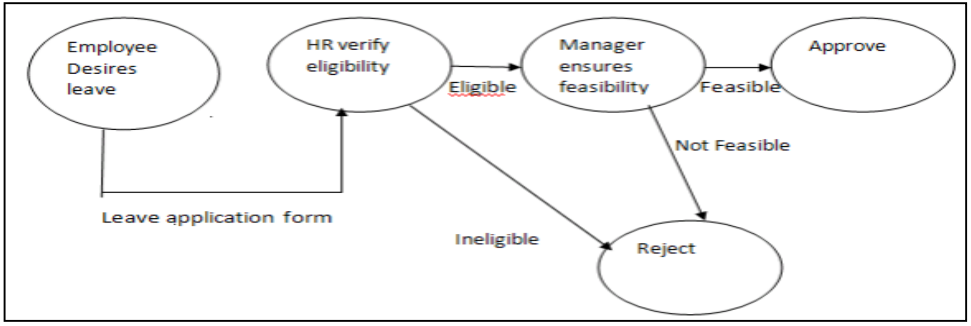


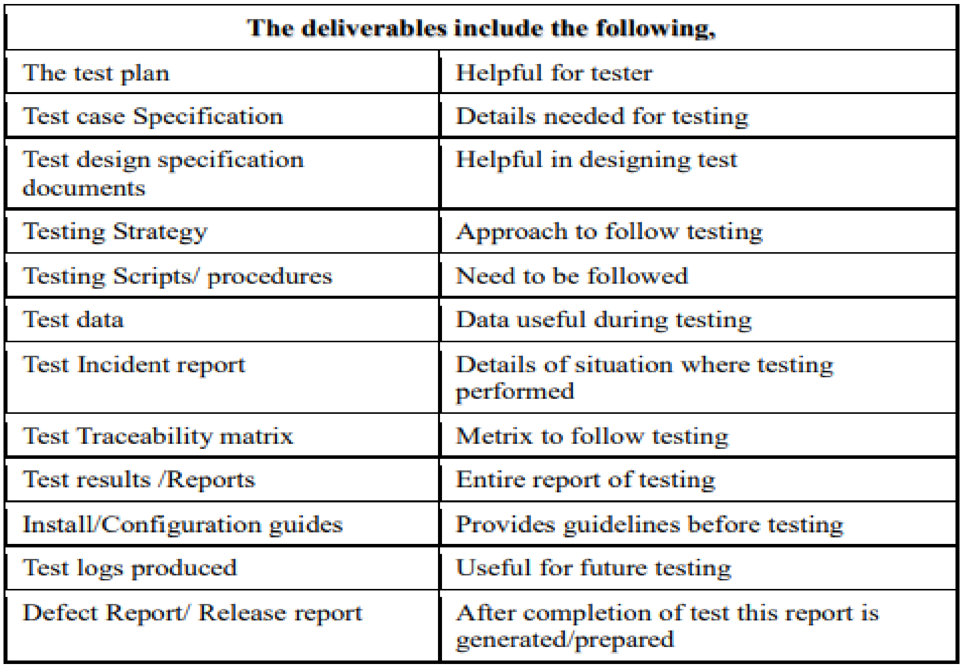
ID	Unique identifier given to the defect. (Usually Automated)
Project	Project name.
Product	Product name.
Release Version	Release version of the product. (e.g. 1.2.3)
Module	Specific module of the product where the defect was detected.
Detected Build Version	Build version of the product where the defect was detected (e.g. 1.2.3.5)
Summary	Summary of the defect. Keep this clear and concise.
Description	Detailed description of the defect. Describe as much as possible but without Repeating anything or using complex words. Keep it simple but comprehensive.
Steps to Replicate	Step by step description of the way to reproduce the defect. Number the steps.
Actual Result	The actual result you received when you followed the steps.
Expected Results	The expected results.
Attachments	Attach any additional information like screenshots and logs.
Remarks	Any additional comments on the defect.
Defect Severity	Severity of the Defect.

	f	**State various advantages and disadvantages of using manual testing tools.**	
	Ans	**Advantages of using manual testing tools :** **1. Speed.** The automation tools tests the software under tests with the very faster speed. There is a vast difference between the speed of user entering the data and the automated tools generating and entering the data required for the testing of the software. Speed of this software also completes the work faster.	Any two advantage : 2 M, Any two disadvantage :2 M **OR** Answer with

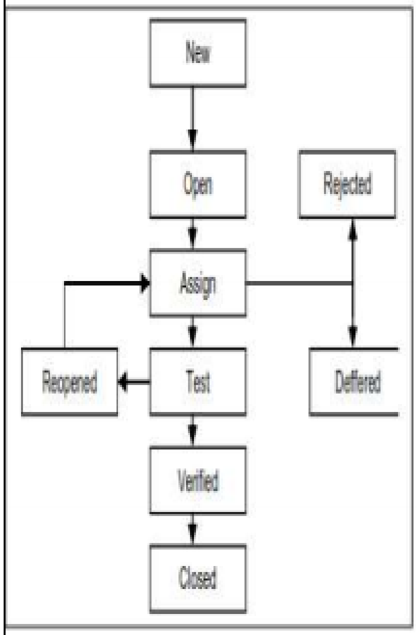
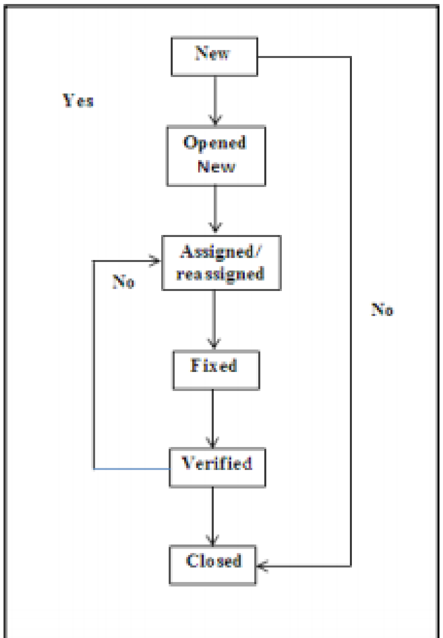
		2. **Efficiency.** While testers are busy running test cases, testers can't be doing anything else. If the tester have a test tool that reduces the time it takes for him to run his tests, he has more time for test planning and thinking up new tests. 3. **Accuracy and Precision.** After trying a few hundred cases, tester_s attention span will wane and he may start to make mistakes. A test tool will perform the same test and check the results perfectly, each and every time. 4. **Resource Reduction.** Sometimes it can be physically impossible to perform a certain test case. The number of people or the amount of equipment required to create the test condition could be prohibitive. A test tool can be used to simulate the real world and greatly reduce the physical resources necessary to perform the testing. 5. **Simulation and Emulation**. Test tools are often used to replace hardware or software that would normally interface to your product. This "fake" device or application can then be used to drive or respond to your software in ways that you choose and ways that might otherwise be difficult to achieve. 6. **Relentlessness.** Test tools and automation never tire or give up. they can keep going and going and on and on without any problem; whereas the tester gets tired to test again and again. **OR** • Reduce time of testing • Improve the bugs finding • Deliver the quality software/product • Allow to run tests many times with different data • Getting more time for test planning • Save resources or reduce requirement • It is never tired and expert person can work at a time many tools. **Disadvantages of using manual testing tools :** • It's more expensive to automate. Initial investments are bigger than manual testing • Manual tests can be very time consuming. • You cannot automate everything; some tests still have to be done manually. • You cannot rely on testing tools always.	Relevant Contents

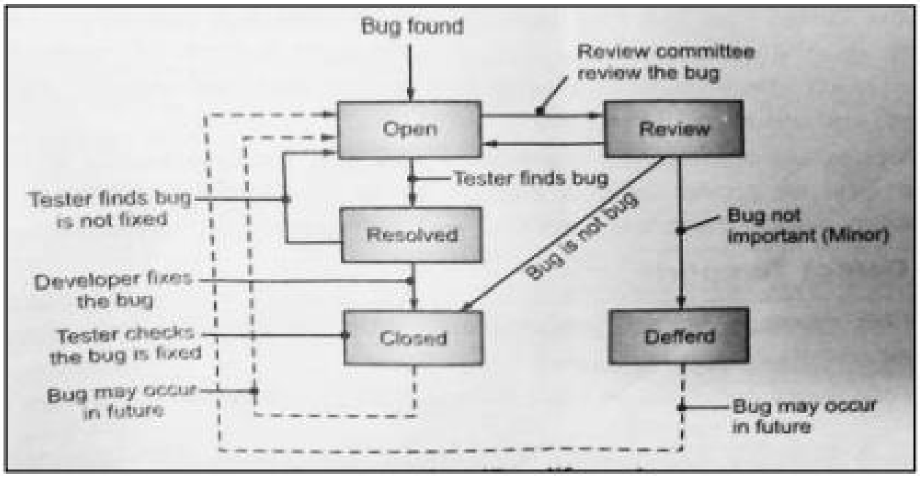
4		**Attempt any Four of the following:**	**16 M**
	a	**Explain quality assurance and quality control**	
	Ans	**Quality Assurance:** i. It is Process oriented activities. ii. A part of quality management focused on providing confidence that quality requirements will be fulfilled. iii. All the planned and systematic activities implemented within the quality system that can be demonstrated to provide confidence that a product or service will fulfill requirements for quality iv. Quality Assurance is fundamentally focused on planning and documenting those processes to assure quality including things such as quality plans and inspection and test plans. v. Quality Assurance is a system for evaluating performance, service, of the quality of a product against a system, standard or specified requirement for customers. vi. Quality Assurance is a complete system to assure the quality of products or services. It is not only a process, but a complete system including also control. It is a way of management. **Quality Control:** i. It is Product oriented activities. ii. A part of quality management focused on fulfilling quality requirements. iii. The operational techniques and activities used to fulfill requirements for quality. iv. Quality Control on the other hand is the physical verification that the product conforms to these planned arrangements by inspection, measurement etc. v. Quality Control is the process involved within the system to ensure job management, competence and performance during the manufacturing of the product or service to ensure it meets the quality plan as designed. vi. Quality Control just measures and determines the quality level of products or services.	Quality assurance: 2 M, Quality Control: 2 M **OR** Answer with Relevant Contents
	b	**Explain graph-based testing with example**	
	Ans	Black-box methods based on the nature of the relationships (links) among the program objects (nodes), test cases are designed to traverse the entire graph Transaction flow testing – nodes represent steps in some transaction and links represent logical connections between steps that need to be validated **i.** Finite state modeling – nodes represent user observable states of the software and links represent transitions between states **ii.** Data flow modeling – nodes are data objects and links are transformations from one data object to another	Explanation: 2 M, Example: 2 M, OR Any other relevant example shall be considered

		iii. Timing modeling – nodes are program objects and links are sequential connections between these objects, link weights are required execution times. **Steps in graph testing**: I. Build a graph model. II. Identify the test requirements. III. Select test paths to cover those requirements. Derive test data so that those test paths can be executed.  Employee Desires leave HR verify eligibility Manager ensures feasibility Approve Eligible Feasible Not Feasible Leave application form Ineligible Reject	
	c	**Explain web based testing for useable website.**	
	Ans	**Web Based Testing**: Web application testing, a software testing technique exclusively adopted to test the applications that are hosted on web in which the application interfaces and other functionalities are tested. Web Application Testing Techniques: 1. Functionality Testing: FUNCTIONAL TESTING is a type of software testing whereby the system is tested against the functional requirements/specifications. 2. Usability testing: It is done to Check "Ease of use" of an application to a common user who will use the application in production environment. 3. Interface testing: Interface Testing is defined as a software testing type which verifies whether the communication between two different software systems is done correctly. A connection that integrates two components is called interface. 4. 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 5. Performance testing : Performance testing is the process of determining the speed, responsiveness and stability of a computer, network, software program or device under a workload 6. Security testing: SECURITY TESTING is a type of software testing that intends to uncover vulnerabilities of the system and determine that its data and resources are protected from possible intruders.	Explanation 4 m **OR** Answer with Relevant Contents

	d	**Explain the need of test deliverables for test planning**	
	Ans		any eight points :1/2 M each **OR** Answer with Relevant Contents

The deliverables include the following,	
The test plan	Helpful for tester
Test case Specification	Details needed for testing
Test design specification documents	Helpful in designing test
Testing Strategy	Approach to follow testing
Testing Scripts/ procedures	Need to be followed
Test data	Data useful during testing
Test Incident report	Details of situation where testing performed
Test Traceability matrix	Metrix to follow testing
Test results /Reports	Entire report of testing
Install/Configuration guides	Provides guidelines before testing
Test logs produced	Useful for future testing
Defect Report/ Release report	After completion of test this report is generated/prepared

	e	**Explain defect life cycle with diagram**	
	Ans	 **OR**  Fig: Defect life cycle **OR**	Diagram 2 m, Explanation 2 m **OR** Answer with Relevant Contents



- **Open**: After a tester has posted a bug, the lead of the tester approves that the bug is genuine and he changes the state as —OPEN‖.

- **Assign**: Once the lead changes the state as —OPEN‖, he assigns the bug to corresponding developer or developer team. The state of the bug now is changed to —ASSIGN‖.

- **Test/Retest**: Once the developer fixes the bug, he has to assign the bug to the testing team for next round of testing. Before he releases the software with bug fixed, he changes the state of bug to —TEST‖. It specifies that the bug has been fixed and is released to testing team.// At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.

- **Deferred:** The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software.

- **Rejected:** If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to —REJECTED.

- **Verified:** Once the bug is fixed and the status is changed to —TEST, the tester tests the bug. If the bug is not present in the software, he approves that the bug is fixed and changes the status to —VERIFIED.

- **Reopened:** If the bug still exists even after the bug is fixed by the

		developer, the tester changes the status to ―REOPENED. The bug traverses the life cycle once again.	
	f	**Explain needs of automation testing**	
	Ans	i. An automated testing tool is able to playback pre-recorded and predefined actions. Compare the results to the expected behavior and report the success or failure of these manual tests to a test engineer. ii. Once automated tests are created they can easily be repeated and they can be extended to perform tasks impossible with manual testing. iii. Because of this, savvy managers have found that automated software testing is an essential component of successful development projects. **Needs of automation testing:** 1. **Speed:** Think about how long it would take you to manually try a few thousand test cases for the windows Calculator. You might average a test case every five seconds or so. Automation might be able to run 10, 100 even 1000 times that fast. 2. **Efficiency**: While you are busy running test cases, you can't be doing anything else. If you have a test tool that reduces the time it takes for you to run your tests, you have more time for test planning and thinking up new tests. 3. **Accuracy and Precision**: After trying a few hundred cases, your attention may reduce and you will start to make mistakes .A test tool will perform the same test and check the result perfectly, each and every time. 4. **Resource Reduction**: Sometimes it can be physically impossible to perform a certain test case. The number of people or the amount of equipment required to create the test condition could be prohibitive. A test tool can used to simulate the real world and greatly reduce the physical resources necessary to perform the testing. 5. **Simulation and Emulation**: Test tools are used to replace hardware or software that would normally interface to your product. This "face" device or application can then be used to drive or respond to your software in ways that you choose-and ways that might otherwise be difficult to achieve. 6. **Relentlessness**: Test tool and automation never tire or give up. It will continuously test the software. **OR** **Needs of Automation Testing are:**	Explanation 4 m **OR** Answer with Relevant Contents

		1. **Save Time /Speed**: Due to advanced computing facilities, automation test tools prevail in speed of processing the tests. Automation saves time as software can execute test cases faster than human. 2. **Reduces the tester's involvement in executing tests**: It relieves the testers to do some other work. 3. **Repeatability/Consistency**: The same tests can be re-run in exactly the same manner eliminating the risk of human errors such as testers forgetting their exact actions, intentionally omitting steps from the test scripts, missing out steps from the test script, all of which can result in either defects not being identified or the reporting of invalid bugs (which can again, be time consuming for both developers and testers to reproduce) 4. **Simulated Testing:** Automated tools can create many concurrent virtual users/data and effectively test the project in the test environment before releasing the product. 5. **Test case design:** Automated tools can be used to design test cases also. Through automation, better coverage can be guaranteed than if done manually.	
5		**Attempt any TWO of the following:**	
	a	**Explain verification and validation with neat diagram**	
	Ans	Verification and validation model makes the V-model. It is sequential path of execution of processes. Each phase must be completed before the next phase begins. Under V-model, the corresponding testing phase of the development phase is planned in parallel. So there is verification on one side of V & validation phase on the other side of V. **Verification Phase:** 1. **Overall Business Requirement:** In this first phase of the development cycle, the product requirements are understood from customer perspective. This phase involves detailed communication with the customer to understand his expectations and exact requirements. The acceptance test design planning is done at this stage as business requirements can be used as an input for acceptance testing. 2. **Software Requirement**: Once the product requirements are clearly known, the system can be designed. The system design comprises of understanding & detailing the complete hardware , software & communication set up for the product under development. System test plan is designed based on system design. Doing this at earlier	Diagram-4 M, Explanation of Verification Phase- 2 M, Explanation of Validation Phase-2 M **OR** Answer with Relevant Contents

		stage leaves more time for actual test execution later. 3. **High level design:** High level specification are understood & designed in this phase. Usually more than one technical approach is proposed & based on the technical & financial feasibility, the final decision is taken. System design is broken down further into modules taking up different functionality. 4. **Low level design:** In this phase the detailed integral design for all the system modules is specified. It is important that the design is compatible with the other modules in the system & other external system. Components tests can be designed at this stage based on the internal module design, 5. **Coding:** The actual coding of the system modules designed in the design phase is taken up in the coding phase. The base suitable programming language is decided base on requirements. Coding is done based on the coding guidelines & standards. **Validation:** 1. **Unit Testing:** Unit testing designed in coding are executed on the code during this validation phase. This helps to eliminate bugs at an early stage. 2. **Components testing:** This is associated with module design helps to eliminate defects in individual modules. 3. **Integration Testing:** It is associated with high level design phase & it is 4. performed to test the coexistence & communication of the internal modules within the system 5. **System Testing:** It is associated with system design phase. It checks the entire system functionality & the communication of the system under development with external systems. Most of the software & hardware compatibility issues can be uncovered using system test execution. 6. **Acceptance Testing:** It is associated with overall & involves testing the product in user environment. These tests uncover the compatibility issues with the other systems available in the user environment. It also uncovers the non-functional issues such as load & performance defects in the actual user environment.	

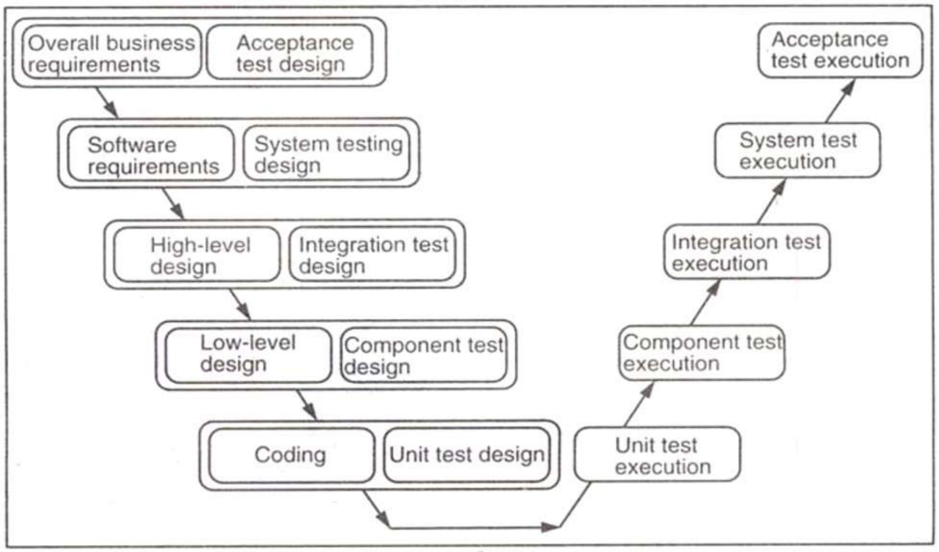


Fig. V-MODEL (Verification And Validation Phases)

Or

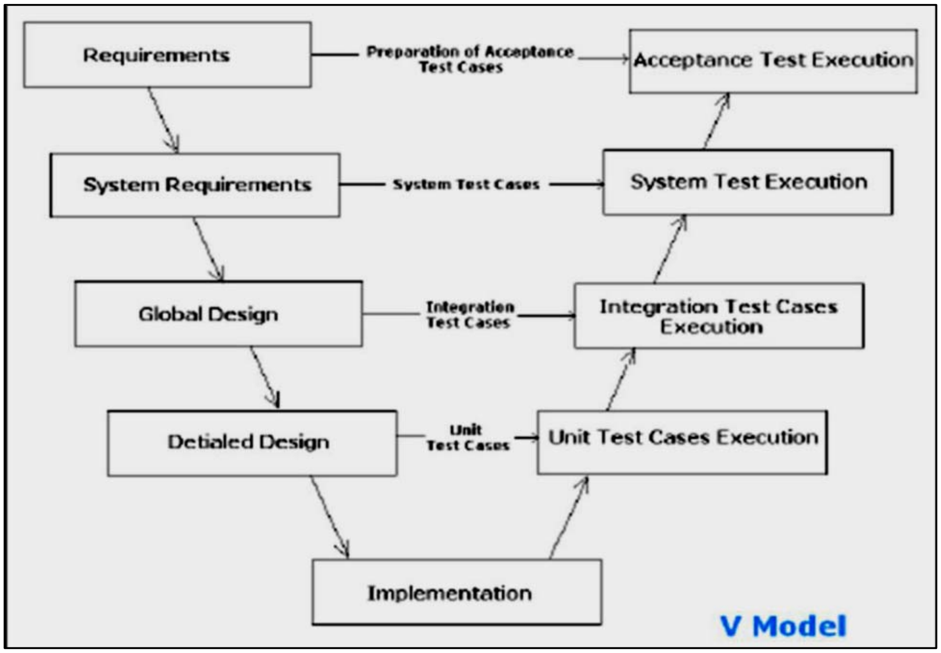


Fig. V-MODEL (Verification And Validation Phases)

b **Differentiate between alpha testing and beta testing.**

Ans

Sr. No	**Alpha testing**	**Beta testing**
1	Performed at developer's site	Performed at end user's site

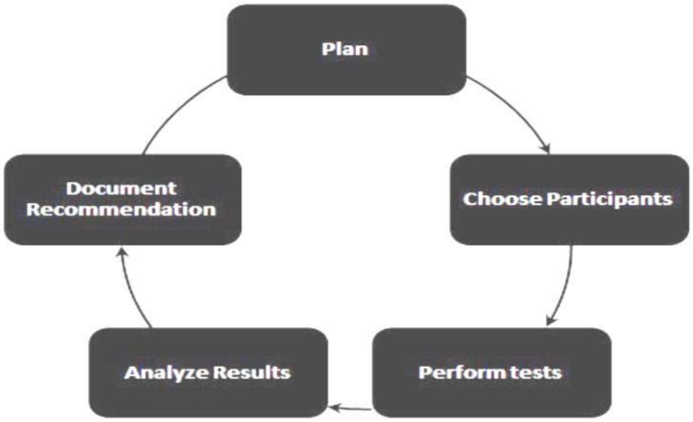
any valid 8 differences :8 M 1 M each)
(Note :any other

<table>
<tr><td></td><td></td><td>
<table>
<tr><td>2</td><td>Performed in controlled environment in developers presence</td><td>Performed in uncontrolled environment in developers absence</td></tr>
<tr><td>3</td><td>Less probability of finding errors as it is driven by developer</td><td>High probability of finding errors as it is used by end user.</td></tr>
<tr><td>4</td><td>It is done during implementation phase of software</td><td>It is done at the pre-release of the software</td></tr>
<tr><td>5</td><td>It is not considered as live application of software</td><td>It is considered as a live application of the software.</td></tr>
<tr><td>6</td><td>Less time consuming as developer can make necessary changes in given time</td><td>More time consuming as user has to report the bugs if any via appropriate channels.</td></tr>
<tr><td>7</td><td>Alpha testing involves both white box and black box testing</td><td>Beta testing typically uses black box testing only</td></tr>
<tr><td>8</td><td>Long execution cycles may be required for alpha testing</td><td>Only a few weeks of execution are required for beta testing</td></tr>
</table>
</td><td>relevant points shall be given marks
OR
Answer with Relevant Contents</td></tr>
<tr><td></td><td>**c**</td><td>**Explain the need of staff training and resource requirements in Test planning in software testing.**</td><td></td></tr>
<tr><td></td><td>**Ans**</td><td>**staff training** : This activity of test planning will give the idea about the following points:
1. How many staff needs training?
2. Who are the attendees?
3. What training needs to be given?
4. What are the pre requisites of the training?
5. How long will be the training?
6. Where training will be conducted
Etc.
Resource requirements:
Factors to be considered while selecting the resource requirements are :
People: How many people are required?

How much experience they should possess? What kind of experience is needed? What should they be expertise in? Should they be full-time, part-time, contract, students?
Equipment: How many Computers are required?</td><td>staff training and resource requirements : explanation : 8 M
OR
Answer with Relevant Contents</td></tr>
</table>

		What configuration computers will be required? What kind of test hardware is needed? Any other devices like printers, tools etc. **Office and lab space:** Where will they be located? How big will they be? How will they be arranged? **Software**: Word processors, databases, custom tools. What will be purchased, what needs to be written? **Outsource companies:** Will they be used? What criteria will be used for choosing them? How much will they cost? **Miscellaneous supplies:** Disks, phones, reference books, training material. What else might be necessary over the course of the project? The specific resource requirements are very project-, team-, and company-dependent, so the test plan effort will need to carefully evaluate what will be needed to test the software.	
6		**Attempt any Four of the following:**	
	a	**Explain white box testing as technical review.**	
	Ans	Technical review is a static white box testing method. **Technical Review:** **i. Formal Review:** • A formal review is the process under which static white box testing is performed. • A formal review can range from a simple meeting between two programmers to a detailed, rigorous inspection of the code. There are four essential elements to a formal review 1. Identify Problems: 2. Follow Rules: 3. Prepare: - 4. Write a Report: **ii. Peer Reviews:** The easiest way to get team members together and doing their first formal reviews of the software is through peer reviews, the least formal method. Sometimes called buddy reviews, this method is really more of a discussion. Peer reviews are often held with just the programmer who wrote the code and one or two other programmers or testers acting as reviewers. Small group simply reviews the code together and looks for problems and oversights. To assure that the review is highly effective all the participants need to make sure that the four key elements of a formal review are in place: Look for problems, follow rules, prepare for the review, and write a report. As peer reviews are informal, these elements are often scaled back. Still,	Explanation of technical review as a part of white box testing : 4 M **OR** Answer with Relevant Contents

		just getting together to discuss the code can find bugs.	
	b	**Explain impact of equivalence partitioning in coding and testing.**	
	Ans	Based on the equivalence portioning technique, the equivalence partitions that are based on age are given below: • Below 35 years of age (valid input) • Between 35 and 59 years of age (valid input) • Above 6 years of age (valid input) • Negative age (invalid input) • Age as 0(invalid input) • Age as any three-digit number (valid input)	Equivalence partitioning with any suitable example : 4 M **OR** Answer with Relevant Contents
	c	**Explain acceptance testing and usability testing.**	
	Ans	**Acceptance testing:** • It is associated with overall & involves testing the product in user environment. • These tests uncover the compatibility issues with the other systems available in the user environment. • It also uncovers the non-functional issues such as load & performance defects in the actual user environment. **Advantages:** • It is conducted to ensure that system requirements meet business needs. • The UAT process allows for any issues to be fixed before the system goes live. • It helps in simulating the real-time user behavior and environment. • It allows the company to improve the software quality by involving customer feedback. **Usability testing :** i. Usability testing, a non-functional testing technique that is a measure of how easily the system can be used by end users. ii. It is difficult to evaluate and measure but can be evaluated based on the below parameters: Levels of Skill required learn/use the software. It should maintain the balance for both novice and expert user. Time required to get used to in using the software. • The measure of increase in user productivity if any. • Assessment of a user's attitude towards using the software. • Usability testing, a non-functional testing technique that is a measure of how easily the system can be used by end users. • It is difficult to evaluate and measure but can be evaluated based on the below parameters:	Acceptance testing 2 marks, Usability testing :2 marks **OR** Answer with Relevant Contents

		- Levels of Skill required learn/use the software. It should maintain the balance for both novice and expert user. - Time required to get used to in using the software. - The measure of increase in user productivity if any. - Assessment of a user's attitude towards using the software. **Fig . Usability testing process**  Plan Choose Participants Perform tests Analyze Results Document Recommendation	
	d	**Explain how summary report is prepared in test planning.**	
	Ans	**Test summary report:** The final step in a test cycle is to recommend the suitability of a product for release. A report that summarizes the result of a test cycle is the test summary report. There are two types of test summary report: 1.Phase wise test summary ,which is produced at the end of every phase 2. Final test summary report. A Summary report should present 1. Test Summary report Identifier 2. Description Identify the test items being reported in this report with test id 3. Variances mention any deviation from test plans, test procedures, if any. 4. Summary of results all the results are mentioned here with the resolved incidents and their solutions. 5. Comprehensive assessment and recommendation for release should include Fit for release assessment and recommendation of release OR - The final step in a test cycle is to recommend the suitability of a product for release. - A report that summarizes the result of a test cycle is the test summary report. There are two types of test summary report:	Test summary report : 4 M explanation **OR** Answer with Relevant Contents

		• Phase wise test summary, which is produced at the end of every phase • Final test summary report. A Summary report should present: Test Summary report Identifier Description Identify the test items being reported in this report with test id 1) Variances: Mention any deviation from test plans, test procedures, if any. 2) Summary of results All the results are mentioned here with the resolved incidents and their solutions. 3) Comprehensive assessment and recommendation for release should include Fit for release assessment and recommendation of release.	
	e	**Explain people management in test planning**	
	Ans	**Test People Management** People management is an integral part of any project management and test planning. People management also requires the ability to hire, motivate, and retain the right people. These skills are seldom formally taught. Testing projects present several additional challenges. We believe that the success of a testing organization depends vitally on judicious people management skills **Test Lead responsibilities and activities:** • Identify how the test teams formed and aligned within organization • Decide the roadmap for the project • Identify the scope of testing using SRS documents. • Discuss test plan, review and approve by management/ development team. • Identify required metrics • Calculate size of project and estimate efforts and corresponding plan. • Identify skill gap and balance resources and need for training education. • Identify the tools for test reporting , test management, test automation, • Create healthy environment for all resources to gain maximum throughput. • Identify how the test teams formed and aligned within organization management/ development team. • Test team responsibilities and activities: • Initiate the test plan for test case design • Conduct review meetings • Monitor test progress , check for resources, balancing and allocation • Check for delays in schedule discuss, resolve risks if any.	people management : explanation : 4 M) (Note :any relevant answer shall be given M **OR** Answer with Relevant Contents

		• Intimate status to stake holders and management • Bridge the gap between test team and management. Consider followings for managing test ✓ Understand testers ✓ Test work environment ✓ Role of the test team	
	f	**What are static and dynamic testing tool.**	
	Ans	Static testing tools are used during static analysis of a system. **Static testing tools**: are used throughout a software development life cycle, e.g. tools used for verification purposes. • There are many varieties of static testing tools used by different people as per the type of system being developed. • These tools do not involve actual input and output. Rather, they take a symbolic approach to testing, i.e. they do not test the actual execution of the software. e.g. Flow analyzers, Coverage analyzers, Interface analyzer • Code complexity measurement tools can be used to measure the complexity of a given code. Similarly, data-profiling tools can be used to optimize a database. Code-profiling tools can be used to optimize code. Test-generators are used for generating a test plan form code. Syntax-checking tools are used to verify correctness of code. **Dynamic testing tools** are used at different levels of testing starting from unit testing & which may go up to system testing & performance testing. • These tools are generally used by tester. • These tools test the software system with live data. e.g. Test driver, Test beds**,** Emulators • There are many different tools used for dynamic testing. • Some of the areas covered by testing tools are: 1. Regression testing using automated tools. 2. Defect tracking and communication systems used by tracking & communication. Performance, Load, stress-testing tools.	Static tools- 2 M, Dynamic tools- 2 M **OR** Answer with Relevant Contents