SCHEME OF VALUATION

(Scoring Indicators)

Revision: 2015

Course Title: SOFTWARE TESTING

Course Code: 6136

Q.No	Scoring Indicator	Split up score	Sub total	Total
	PART A			
I(1)	Software testing is a process that detects bugs with the objective of having better quality software.	2	2	2
I(2)	It is one of the major techniques in dynamic testing for designing effective test cases. It considers only the functional requirements of the software or module.	1+1	2	2
I(3)	It is the process of mutating some segment of the code (putting some error in the code) and testing this mutated code with some test data.	2	2	2
I(4)	Winrunner, Silktest, Loadrunner, Jmeter, Testdirector. [any two]	1 + 1	2	2
I(5)	Debugger is a tool to help track down, isolate and remove bugs from the software program.	2	2	2
	PART B			
II(1)	a) Immediate goals -- --- Bug Discovery, Bug Prevention. b) Long term goals -- -- Quality, Customer Satisfaction, Risk Management. c) Post implementation goals – Reduced Maintenance cost, improved software testing process. Explanation of the above required.	List-3 Explanation-3	3+3=6	6
II(2)	Software testing methodology is the organisation of software testing by means of which the test strategy and test tactics are achieved. Figure shown in Annexure 1. The following steps are to be mentioned. a. Software testing strategy – test factors and test phase. b. Test strategy matrix – Select and rank test factors, Identify system development phases, Identify risks associated with the system under development, creating a test strategy, select and rank test factors, identify the test phases, identify the risk associated with each test factor and its corresponding test phase, plan the test strategy for each risk identified. c. Development of test strategy. d. Testing life cycle model. e. Validation activities. f. Testing tactics.	Definition-1 Figure - 2 Explanation-3	1+2+3= 6	6
II(3)	Boundary value analysis is the method to uncover the bugs by looking at the boundary of the inputs used in the program. The idea is that the programmer tests the program by having nominal values of the inputs [a high chance to find errors]. Explanation with an example of a small program / function and a suitable test case as input.	Explanation 3 + example 3	6	6
II(4)	1. Statement coverage 2. Branch coverage 3. Condition coverage	Explanation-2 marks each	2+2+2=6	6
II(5)	1. Bug reduction. 2. Productivity. 3. Real time feedback to software engineers. 4. Reduction in development resource. 5. Quality improvement. 6. Project management. 7. Checking coupling and cohesion. 8. Learning through inspection. 9. Process improvement.	For listing any 6 half marks each. Explanation of each half marks	3+3 = 6	6

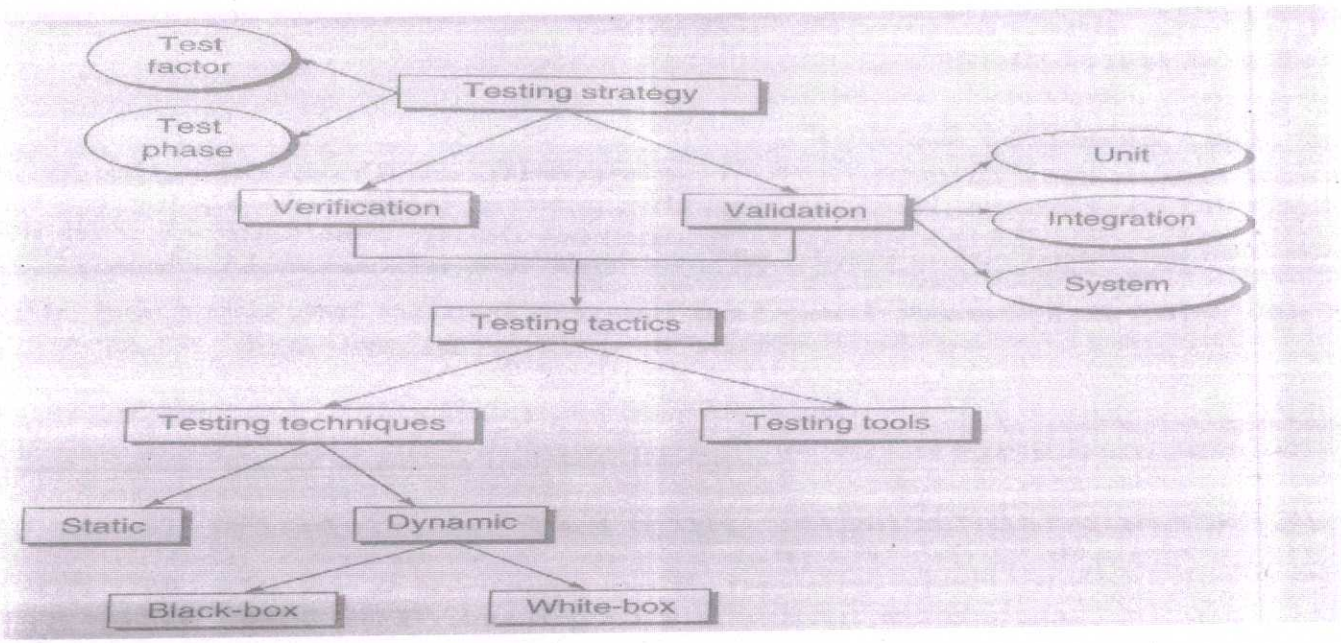
II(6)	1. Reduction of testing effort. 2. Facilitates regression testing. 3. Avoid human mistakes. 4. Reduces overall cost of the software. 5. Simulated testing. 6. Internal testing. 7. Test enablers. 8. Test case design.	For listing any 6 half marks each. Explanation of each half marks	3 +3 = 6	6
II(7)	1. Fresh thinking leads to good debugging. 2. Don't isolate the bugs from your colleagues. 3. Don't attempt code modifications in the first attempt. 4. Additional test cases are a must if you don't get the symptom or clues to solve the problem. 5. Regression testing is a must after debugging. 6. Design should be referred before fixing the error.	For listing 6 half marks each. Explanation of each half marks 3+3=6	3 +3 = 6	6
III(a)	1. Software and software model. 2. Bug model. 3. Testing methodology and testing. Explanation with diagram needed. Figure shown in Annexure 2.	Explanation 6 + figure 4 =10	10	10
III(b)	1. Exhaustive or complete software testing means that every statement in the program and every possible path combination with every possible combination of data must be executed. 2. Effective testing provides the flexibility to select only the subsets of the domain of testing based on project priority such that the chances of failure in a particular environment are minimized. Comparison of the above with testing domain, valid inputs, invalid inputs, defects etc.	2.5 * 2=5	5	5
IV	The testing process divided into a well defined sequence of steps is termed as software testing life cycle. [STLC]. a. Test planning. b. Test design. c. Test execution. d. Post execution/ test review. Explanation of four phases with a neat diagram. Annexure 3.	3* 4 = 12 + Diagram 3 marks	15	15
V.	Tables are useful tools for representing and documenting many types of information relating to test case design. These are beneficial for the applications which can be described using state transition diagrams and state tables. 1. Finite state machine [FSM]. 2. State transition diagram or state graph.	3* 4 = 12 + example	15	15

	3. State table. 4. State table based testing. Explain the above four steps with the help of an example.	3 marks		
VI(a)	**Structured walkthroughs are a less formal and less rigorous technique. A structured walkthrough team consists of** **a.** Coordinator. **b.** Presenter/developer. **c.** Scribe/Recorder. **d.** Reviewer/Tester. **e.** Maintenance Oracle. **f.** Standards bearer. **g.** User Representative. Steps in Walkthrough process Organisation →Preparation →Walkthrough →Research and Follow-up.	Definitio n-2 Team details 4 Steps 2	2+4+ 2= 8	8
VI(b)	**Unit is the smallest building block of the software system, it is the first piece of system to be validated. Before validating the entire software, units or modules must be validated. For that stubs and drivers are necessary.** **Need for stubs and drivers to be explained.**	3+ 2+2 = 7	7	7
VII(a)	**a. Match the tool to its appropriate use.** **b. Select the tool to its appropriate SDLC phase.** **c. Select the tool to the skill of the tester.** **d. Select the tool which is affordable.** **e. Determine how many tools are required for testing the system.** **f. Select the tool after examining the schedule of testing.**	Listing and Explanat ion – 8	8	8
VII(b)	a. Consider building the tool instead of buying one, if possible. b. Test the tool on an application prototype. c. Not all tests should be automated. d. Select the tool according to organizational needs. e. Use proven test-script development techniques. f. Automate the regression tests whenever feasible.	Listing and Explanat ion – 7	7	7
VIII	Winrunner – tool used for functional/regression testing. Silktest – also used for functional/regression testing.[in OOPS scripting language]. Loadrunner --- used for load testing. Jmeter – open source software tool for performance /load testing. Testdirector --- is a test management tool.	3 marks each. 3*5=15	15	15
IX(a)	The goal of debugging process is to determine the exact nature of failure with the help of symptom identified, locate the bugs and errors and finally correct it. Explanation and figure needed. Annexure 4.	Explanat ion 5 Figure 5	5+5= 10	10
IX(b)	The program's final output failure may not give sufficient clues about the bug. At a particular point of execution in the program, values of variables or other actions can be verified. These particular points of execution are known as watch points. Methods:- a. Output statements. b. Breakpoint execution..	Explanat ion-5	5	5

	c. Single stepping. d. Step-into. e. Step-over.			
X(a)	i) In this technique, a printout of all registers and relevant memory locations is obtained and studied. The storage locations are in octal or hexadecimal format. The relevant data of the program is observed through these memory locations and registers for any bug in the program. ii) This is a logical approach for debugging a program. The following steps to be done. a. Observe the symptom. b. Trace the source code. c. Isolate the module. d. Error removal.	Explanation-5each	10	10
X(b)	a. Kernel debugger. b. Basic -machine level debugger. c. In-circuit emulator. d. Interpretive programming environment debugger.	5	5	5

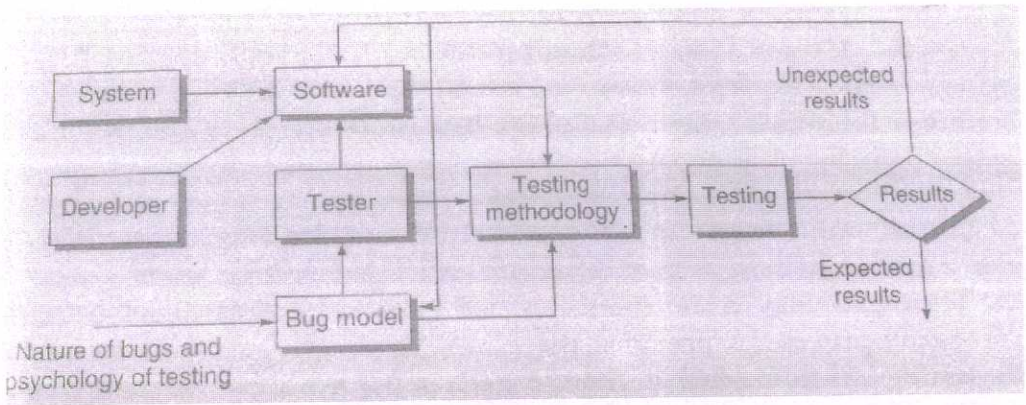
Question Part B—II(2) –Annexure 1

Software testing methodology



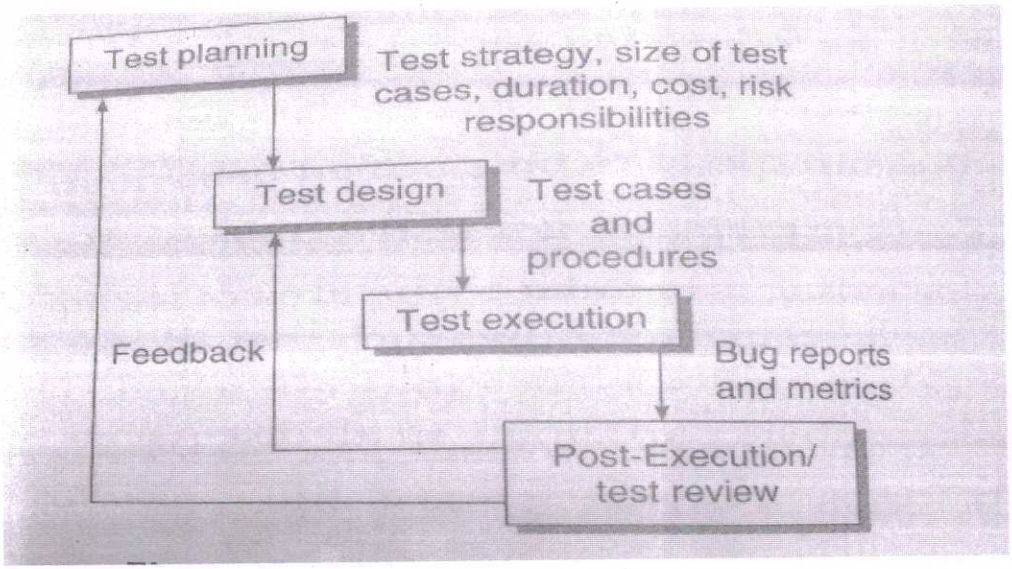
Question Part C—III(a) –Annexure 2

Model of software testing



Question Part C—IV(a) –Annexure 3

STLC



Question Part C—IX(a) –Annexure 4

Debugging Process