<u>**QUESTION BANK**</u>

SUBJECT: 190E125 – SOFTWARE TESTING AND AUTOMATION
YEAR /SEM: III /VI

UNIT I	INTRODUCTION

Testing as an Engineering Activity – Testing as a Process – Testing axioms – Basic definitions – Software Testing Principles – The Tester"s Role in a Software Development Organization – Origins of Defects – Cost of defects – Defect Classes – The Defect Repository and Test Design – Defect Examples – Developer/Tester Support of Developing a Defect Repository – Defect Prevention strategies.

Q.No	PART – A	CO	Blooms Taxonomy Level
1	1. List the levels of TMM. **(Apr/May – 2018)** • The testing maturity model or TMM contains five levels. They are: • Level1: Initial • Level2: Phase definition • Level3: Integration • Level4: Management and Measurement • Leval5: Optimization /Defect prevention and Quality Control	**C311.1**	**BTL-1**
2	2. Define test, test oracle and test bed. **(Apr/May – 2018)** • A test is a group of related test cases, or a group of related test cases and test procedures. • A test oracle is a document, or piece of software that allows testers to determine whether a test has been passed or failed. • A test bed is an environment that contains all the hardware and software needed to test a software component or a software system.	**C311.1**	**BTL-1**
3	3. Differentiate errors, faults and failures. **(Apr/May – 2017)** • An error is a mistake, misconception, or misunderstanding on the part of a software developer. • A fault (defect) is introduced into the software as the result of an error. It is an anomaly in the software that may cause it to behave incorrectly, and not according to its specification. • A failure is the inability of a software system or component to perform its required functions within specified performance requirements.	**C311.1**	**BTL-1**

4	4. What are the origins of defects? **(Apr/May – 2017)** • Lack of Education • Poor communication • Oversight	**C311.1**	**BTL-2**
	☐ Transcription Immature process		
5	5. Define software testing. **(Nov/Dec – 2016)** • Testing is generally described as a group of procedures carried out to evaluate some aspect of a piece of software. • Testing can be described as a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attributes	**C311.1**	**BTL-1**
6	6.Define feature defects. **(Nov/Dec – 2016) (Nov/Dec – 2018)** ☐ Features may be described as distinguishing characteristics of a software component or system. The defects that are appearing in those components are known as feature defects.	**C311.1**	**BTL-1**
7	7.What is the role of test specialist? ☐ A test specialist who is trained as an engineer should have knowledge of test-related principles, processes, measurements, standards, plans, tools, and methods, and should learn how to apply them to the testing tasks to be performed.	**C311.1**	**BTL-1**
8	8. How can software development be viewed as an engineering approach? ☐ Using an engineering approach to software development implies that: • The development process is well understood. • Projects are planned. • Life cycle models are defined and adhered to. • Standards are in place for product and process. • Measurements are employed to evaluate product and process quality. • Components are reused.	**C311.1**	**BTL-2**
9	9. Define validation. Validation is the process of evaluating a software system or component during, or at the end of, the development cycle in order to determine whether it satisfies specified requirements	**C311.1**	**BTL-1**
10	10.Define verification. Verification is the process of evaluating a software system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase	**C311.1**	**BTL-1**

11	11. What do you mean by debugging or fault localization? Debugging, or fault localization is the process of (1) locating the fault or defect, (2) repairing the code, and (3) retesting the code.	C311.1	BTL-1
12	12. Define test cases. • A test case in a practical sense is a test-related item which contains the following information: • A set of test inputs. These are data items received from an external source by the code under test. The external source can be hardware, software, or human.	C311.1	BTL-1

	• Execution conditions. These are conditions required for running the test, for example, a certain state of a database, or a configuration of a hardware device. • Expected outputs. These are the specified results to be produced by the code under test.		
13	13.Explain some of the quality metric attributes. A quality metric is a quantitative measurement of the degree to which an item possesses a given quality attribute. • Correctness— thedegreetowhichthesystemperformsitsintendedfunction. • Reliability—the degree to which the software is expected to perform its required functions under stated conditions for a stated period of time. • Usability—relates to the degree of effort needed to learn, operate, prepare input, and interpret output of the software. • Integrity—relates to the system's ability to withstand both intentional and accidental attacks. • Portability—relates to the ability of the software to be transferred from one environment to another. • Maintainability—the effort needed to make changes in the software. • Interoperability—the effort needed to link or couple one system to another.	C311.1	BTL-2
14	14.What is the role of software quality assurance (SQA) group? The softwarequalityassurance(SQA)groupisateamofpeoplewiththenecessary training and skills to ensure that all necessary actions are taken during the development process so that the resulting software conforms to established technical requirements.	C311.1	BTL-1
15	15. Define review. A review is a group meeting whose purpose is to evaluate a software artifact or a set of software artifacts.	C311.1	BTL-1
16	16.Define pre-condition and post-condition. ▢ A pre-condition is a condition that must be true in order for a software component to operate properly. A post-condition is a condition that must be true when a software component completes its operation properly	C311.1	BTL-1

17	17. What are interface description defects? These are defects that occur in the description of how the target software is to interface with external software, hardware, and users	C311.1	BTL-1
18	18. Define design defects. ☐ Design defects occur when system components, interactions between system components, interactions between the components and outside software/hardware, or users are incorrectly designed. This covers defects in the design of algorithms, control, logic, data elements, module interface descriptions, and external software/hardware/user interface descriptions.	C311.1	BTL-1
19	19.What do you mean by coding defects?	C311.1	BTL-1

	Coding defects are derived from errors in implementing the code. Some coding defects come from a failure to understand programming language constructs, and miscommunication with the designers. Others may have transcription or omission origins		
20	20. What do you mean by test harness? In order to test software, especially at the unit and integration levels, auxiliary code must be developed. This is called the test harness or scaffolding code.	C311.1	BTL-1
21	21. Define a fault model. A fault (defect) model can be described as a link between the error made (e.g., a missing requirement, a misunderstood design element, a typographical error),and the fault/defect in the software.	C311.1	BTL-1
22	22. What are the uses of hypotheses in software testing process? design test cases; design test procedures; assemble test sets; select the testing levels (unit, integration, etc.) appropriate for the tests; Evaluate the results of the tests.	C311.1	BTL-1
23	23. What is the role of a software tester in a software development organization? The tester's job is to reveal defects, find weak points, inconsistent behavior, and circumstances where the software does not work as expected.	C311.1	BTL-1
24	24. Define Software Engineering. Software Engineering is a discipline that produces error free software with in a time and budget.	C311.1	BTL-1
25	25. Define process in the context of software quality. **(Nov/Dec – 2018)** Process, in the software engineering domain, is a set of methods, practices, Standards, documents, activities, polices, and procedures that software engineers use to develop and maintain a software system and its associated artifacts, such as project and test plans, design documents, code, and manuals.	C311.1	BTL-1

26	26. Programmer A and Programmer B are working on a group of interfacing modules. Programmer A tends to be a poor communicator and does not get along well with Programmer B. Due to this situation, what types of defects are likely to surface in these interfacing modules? Communication defects.	C311.1	BTL-4
27	27. List the elements of the engineering disciplines. Basic principles, Processes, Standards, Measurements, Tools, Methods, Best practices, Code of ethics and Body of knowledge.	C311.1	BTL-1
28	28. Differentiate between testing and debugging. Testing as a dual purpose process reveal defects and to evaluate quality attributes. Debugging or fault localization is the process of locating the fault or defect repairing the code, and retesting the code.	C311.1	BTL-2

29	29. List the members of the critical groups in a testing process. Manager Developer/Tester User/Client	C311.1	BTL-1
30	30.What is the need of CMM (Capability Maturity Model)? These models allow an organization to evaluate its current software process and to capture an understanding of its state.	C311.1	BTL-1
31	31. List out the Software testing axioms. 1. It is impossible to test a program completely. 2. Software testing is a risk-based exercise. 3. Testing cannot show the absence of bugs. 4. The more bugs you find, the more bugs there are. 5. Not all bugs found will be fixed. 6. It is difficult to say when a bug is indeed a bug. 7. Specifications are never final. 8. Software testers are not the most popular members of a project. 9. Software testing is a disciplined and technical profession.	C311.1	BTL-2
32	32. Define Software Engineering Software Engineering is a discipline that produces error free software with in a time and budget.	C311.1	BTL-1
33	33. Define the term Testing. Testing is generally described as a group of procedures carried out to evaluate some aspect of a piece of software. Testing can be described as a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attributes.	C311.1	BTL-1

		C311.1	BTL-2
34	34 Define process in the context of software quality Process, in the software engineering domain, is a set of methods, practices, Standards, documents, activities, polices, and procedures that software engineers use to develop and maintain a software system and its associated artifacts, such as project and test plans, design documents, code, and manuals.		
35	35 List the members of the critical groups in a testing process (U.Q Nov/Dec 2008) • Manager • Developer/Tester • User/Client	C311.1	BTL-1
36	36. Define Faults (Defects). A fault is introduced into the software as the	C311.1	BTL-1

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

37	37 Define Test Cases.
	
	 A test case in a practical sense is attest related item which contains the following information.
	
	    • A set of test inputs. These are data items received from an external source by the code under test. The external source can be hardware, software, or human.
	
	    • Execution conditions. These are conditions required for running the test, for example, a certain state of a database, or a configuration of a hardware device.
	
	    • Expected outputs. These are the specified results to be produced by the code under test.
		C311.1	BTL-2

	38 Write short notes on Test, Test Set, and Test Suite.	C311.1	BTL-1
	A <u>Test</u> is a group of related test cases, or a group of related test cases and test procedure. A group of related test is sometimes referred to as a <u>test set</u> A group of related tests that are associated with a database, and are usually run together, is sometimes referred to as a <u>Test Suite.</u>		
38			
	39. Define Test Bed. A test bed is an environment that contains all the hardware and software needed to test a software component or a software system.	C311.1	BTL-1
39			
	40 Define Software Quality. Quality relates to the degree to which a system, system component, or process meets specified	C311.1	BTL-2
40			

	requirements. Quality relates to the degree to which a system, system component, or process meets Customer or user needs, or expectations.		
	41 Define SQA group. The software quality assurance (SQA) group is a team of people with the necessary training and skills to ensure that all necessary actions are taken during the development process so that the resulting software confirms to established technical requirements	C311.1	BTL-1
41			
	42 Explain the work of SQA group. Testers to develop quality related policies and quality assurance plans for each project. The group is also involved in measurement collection and analysis, record keeping, and Reporting. The SQA team members participate in reviews and audits, record and track Problems, and verify that corrections have been made.	C311.1	BTL-1
42			

		C311.1	BTL-2
43	43. Define reviews. A review is a group meeting whose purpose is to evaluate a software artifact or a set of Software artifacts. Review and audit is usually conducted by a SQA group		
44	44 List the sources of Defects or Origins of defects. Or list the classification of defect • Education • Communication • Oversight • Transcription • Process	C311.1	BTL-1
45	45 Define Test Oracle. Test Oracle is a document, or a piece of software that allows tester to determine whether a test has been passed or failed.	C311.1	BTL-1
46	46 Define failures. A failure is the inability of a software or component to perform its required within specified performance requirements	C311.1	BTL-2
47	47 List the levels of TMM. The testing maturity model or TMM contains five levels. They are Level1: Initial Level2: Phase definition Level3: Integration Level4: Management and Measurement Leval5: Optimization /Defect prevention and Quality Control	C311.1	BTL-1

			C311.1	BTL-1
48	48 List the elements of the engineering disciplines. • Basic principles • Processes • Standards • Measurements • Tools • Methods • Best practices • Code of ethics • Body of knowledge			
49	49 Define the term Debugging or fault localization. Debugging or fault localization is the process of • Locating the fault or defect • Repairing the code, and • Retesting the code		C311.1	BTL-1
50	50. List the Quality Attributes. • Correctness • Reliability • Usability • Integrity • Portability • Maintainability • Interoperability		C311.1	BTL-1

PART B

S. No.	Question	Course Outcome	Blooms Taxonomy
			Level
1	Explain in detail about Testing Maturity Model (TMM) levels and the test related activities that should be done for V-model architecture (Apr May 2018)	C311.1	BTL-2
2	Explain in detail processing and monitoring of the defects with defect repository (Apr May 2018) (PartC)	C311.1	BTL-2
3	Explain various software testing principles in detail and summarize the tester role in software development organization. (Apr May 2018)	C311.1	BTL-2
4	Explain various software testing principles in detail. **(Apr May 2017) (Nov/Dec – 2018)**	C311.1	BTL-2

5	What are the typical origins of defects? Explain the major classes of defects in the software artefects. **(Apr May 2017) (Nov/Dec – 2018)**	**C311.1**	**BTL-2**
6	Explain the challenges of a software tester.	**C311.1**	**BTL-2**
7	Explain in detail about requirement / specification defects.	**C311.1**	**BTL-2**
8	What do you mean by design defects? What are the different design defects? Explain.	**C311.1**	**BTL-2**
9	What are the different types of coding defects? Explain in detail Ref: Practical software Testing By Ilene Burnstein Pg.No: 48-51	**C311.1**	**BTL-3**
10	Explain in detail about different Testing Maturity Model (TMM) levels. Ref: Practical software Testing By Ilene Burnstein P.No: 8 to 16	**C311.1**	**BTL-2**
11	Elaborate on the principles of software testing and summarize the tester role in software development organization.(16) (Nov/Dec – 2016)	**C311.1**	**BTL-2**
12	(i) Describe about the components of software development process.(8) causing organizations to revise their approach to testing. (8)	**C311.1**	**BTL-1**
13	Give the internal structure of TMM and explain about its maturity goals at each level.(16)	**C311.1**	**BTL-1**
14	Give an Overview oft he Testing Maturity Model(TMM) & the test related activities that should be done.(16)	**C311.1**	**BTL-1**
15	Compare and contrast terms errors faults and failures using suitable examples.(8)	**C311.1**	**BTL-3**
16	Discuss the steps to be taken to monitor the defects with the help of defect repository?(16) (Nov/Dec – 2016)	**C311.1**	**BTL-3**
17	Describe about Tester Support of Developing a Defect Repository.(8) **(Nov/Dec – 2018)**	**C311.1**	**BTL-3**

UNIT II	TEST CASE DESIGN

Test case Design Strategies – Using Black Bod Approach to Test Case Design – Random Testing – Requirements based testing – Boundary Value Analysis – Equivalence Class Partitioning – Statebased testing – Cause-effect graphing – Compatibility testing – user documentation testing – domain testing – Using White Box Approach to Test design – Test Adequacy Criteria – static testing vs. structural testing – code functional testing – Coverage and Control Flow Graphs – Covering Code Logic – Paths – code complexity testing – Evaluating Test Adequacy Criteria

S. No.	Question	Course Outcome	Blooms Taxonomy Level
1	1. What are the factors affecting less than 100% degree of coverage? **(Apr/May – 2018)** The nature of the unit • Some statements/branches may not be reachable. • The unit may be simple, and not mission, or safety, critical, and so complete coverage is thought to be unnecessary. • The lack of resources • The time set aside for testing is not adequate to achieve complete coverage for all of the units. • There is a lack of tools to support complete coverage • Other project related issues such as timing, scheduling. And marketing constraints.	**C311.2**	**BTL-1**
2	2. Write the formula for Cyclomatic complexity? **(Apr/May – 2018) (Nov/Dec – 2016)** The complexity value is usually calculated from control flow graph (G) by the formula $V(G) = E-N+2$. Where the value E is the number of edges in the control flow graph and the value N is the number of nodes.	**C311.2**	**BTL-2**
3	3. Differentiate black box and white box testing. **(Apr/May – 2017) (Nov/Dec – 2018)** <table><tr><td>Black box testing</td><td>White box Testing</td></tr><tr><td>Black box testing, the tester is no Knowledge of its inner structure(i.e. how it works)The tester only has knowledge of what it does(Focus only input & output)</td><td>The White box approach focuses on the inner structure of the software to be tested.</td></tr></table>	**C311.2**	**BTL-1**

	z	White box approach is usually applied small size piece of software.		

	Black box testing sometimes called functional or specification testing.	White box sometimes called clear or glass box testing.	
4	4. Give some example methods for black box testing. **(Apr/May – 2017)** • Equivalence class partitioning (ECP) • Boundary value analysis (BVA) ☐ State Transition testing.(STT) ☐ Cause and Effect Graphing. ☐ Error guessing	**C311.2**	**BTL-2**
5	5. List the various iterations of Loop testing. **(Nov/Dec – 2016)** Zero iteration of the loop • One iteration of the loop • Two iterations of the loop • K iterations of the loop where k<n • n-1 iterations of the loop • n+1 iterations of the loop	**C311.2**	**BTL-1**
6	6. Define Smart Tester. Software must be tested before it is delivered to users. It is responsibility of the testers to design tests that reveal defects and can be used to evaluate software performance, usability and reliability. To achieve these goals, tester must select a finite no. of test cases (i/p, o/p, & conditions).	**C311.2**	**BTL-1**
7	7. What is the need of test adequacy criteria? The application scope of adequacy criteria also includes: Helping testers to select properties of a program to focus on during test; Helping testers to select a test data set for a program based on the selected properties; Supporting testers with the development of quantitative objectives for testing; Indicating to testers whether or not testing can be stopped for that program.	**C311.2**	**BTL-1**
8	8 What do you mean by a test data set? A test data set is statement, or branch, adequate if a test set T for program P causes all the statements, or branches, to be executed respectively.	**C311.2**	**BTL-1**

9	9. What are the basic testing strategies? Black box and white box testing.	**C311.2**	**BTL-1**
10	10. Define state. A state is an internal configuration of a system or component . It is defined interms of the values assumed at a particular time for the variables that characterize the system or component.	**C311.2**	**BTL-1**
11	11. What do you mean by a finite state machine? A finite-state machine is an abstract machine that can be represented by a state graph having a finite number of states and a finite number of transitions between states.	**C311.2**	**BTL-1**
12	12.What do you mean by path? A path is a sequence of control flow nodes usually beginning from the entry node of a graph through to the exit node.	**C311.2**	**BTL-1**
13	13.When will you say a variable is used in a statement? A variable is used in a statement when its value is utilized in a statement. The value of the variable is not changed.	**C311.2**	**BTL-1**
14	14.What is def-use path? A path from a variable definition to a use is called a def-use path.	**C311.2**	**BTL-1**
15	15 What is the need for loop testing? Loops are among the most frequently used control structures. Experienced software engineers realize that many defects are associated with loop constructs. These are often due to poor programming practices and lack of reviews. Therefore, special attention should be paid to loops during testing.	**C311.2**	**BTL-4**
16	16.What do you mean by mutation testing? Mutation testing is an approach to test data generation that requires knowledge of code structure, but it is classified as a fault-based testing approach. It considers the possible faults that could occur in a software component as the basis for test data generation and evaluation of testing effectiveness.	**C311.2**	**BTL-2**
17	17.What is the condition for a test set to be mutation adequate? A test set T is said to be mutation adequate for program P provided that for every in equivalent mutant Pi of P there is an element t in T such that Pi(t) is not equal to P(t).	**C311.2**	**BTL-1**
18	18. Write short notes on Random testing and Equivalence class portioning. Each software module or system has an input domain from which test input data is selected. If a tester randomly selects inputs from the domain, this is called random testing. In equivalence class partitioning the input and output is divided in to equal classes or partitions.	**C311.2**	**BTL-2**
19	19.Give some example methods for white box testing. Statement testing	**C311.2**	**BTL-2**

	Branch testing Path testing Data flow testing Mutation testing Loop testing		
20	20. Differentiate black box and white box testing in terms knowledge sources required to perform testing. 	Test Strategy	Knowledge Sources
---	---
Black box	1. Requirements document 2. Specifications 3. Domain Knowledge 4. Defect analysis data
White box	1. High level design 2. Detailed design 3. Control flow graphs 4. Cyclomatic complexity		**C311.2**	**BTL-2**
21	21.Define Error Guessing. The tester/developer is sometimes able to make an educated "guess' as to which type of defects may be present and design test cases to reveal them. Error Guessing is an ad-hoc approach to test design in most cases..	**C311.2**	**BTL-1**
22	22. Define COTS Components. **(Nov/Dec – 2018)** The reusable component may come from a code reuse library within their org or, as is most likely, from an outside vendor who specializes in the development of specific types of software components. Components produced by vendor org are known as commercial off-the shelf, or COTS, components.	**C311.2**	**BTL-1**
23	23. Define usage profiles and Certification. Usage profiles are characterizations of the population of intended uses of the software in its intended environment. Certification refers to third party assurance that a product , process, or service meets a specific set of requirements.	**C311.2**	**BTL-1**
24	24. What are the basic primes for all structured program? Sequential (e.g., Assignment statements) Condition (e.g., if/then/else statements) Iteration (e.g., while, for loops)	**C311.2**	**BTL-1**
25	25. What are the errors uncovered by black box testing? Incorrect or missing functions Interface errors Errors in data structures Performance errors Initialization or termination error	**C311.2**	**BTL-1**
26	26. What is statement coverage property? If the test set T is adequate for P, then T causes every executable statement of P to be executed.	**C311.2**	**BTL-1**

27	27.What are the assumptions taken by mutation testing? The competent programmer hypothesis and the coupling effect.	**C311.2**	**BTL-1**

28	28 What do you mean by a variable is defined in the statement? A variable is defined in a statement when its value is assigned or changed. .	**C311.2**	**BTL-1**
29	29. What are the logic elements analyzed for coverage analysis? The logic elements are: program statements, decisions/branches, conditions, combinations of decisions and conditions and paths.	**C311.2**	**BTL-1**
30	30 What is the need for coverage analysis? The concept of test data adequacy criteria, and the requirement that certain features or properties of the code are to be exercised by test cases, leads to an approach called "coverage analysis, "which is used to set testing goals and to develop and evaluate test data.	**C311.2**	**BTL-1**
31	31 Write short notes on Random testing and Equivalence class portioning. Each software module or system has an input domain from which test input data is selected. If a tester randomly selects inputs from the domain, this is called random testing. In equivalence class partitioning the input and output is divided in to equal classes or partitions.	**C311.2**	**BTL-1**
32	32 Define State. A state is an internal configuration of a system or component. It is defined in terms of the values assumed at a particular time for the variables that characterize the system or component.	**C311.2**	**BTL-2**
33	33 Define Finite-State machine. A finite-state machine is an abstract machine that can be represented by a state graph having a finite number of states and a finite number of transitions between states	**C311.2**	**BTL-1**
34	34 Define usage profiles and Certification. Usage profiles are characterizations of the population of intended uses of the software in its intended environment. Certification refers to third party assurance that a product,process, or service meets a specific set of requirements.	**C311.2**	**BTL-2**

	35	35 Write the application scope of adequacy criteria? • Helping testers to select properties of a program to focus on during test. • Helping testers to select a test data set for a program based	**C311.2**	**BTL-1**

		on the selected properties. • Supporting testers with the development of quantitative objectives for testing • Indicating to testers whether or not testing can be stopped for that program.		
	36	36 What are the factors affecting less than 100% degree of coverage? • The nature of the unit 　　Some statements/branches may not be reachable. 　　　The unit may be simple, and not mission, or safety, critical, and so complete coverage is thought to be unnecessary. • The lack of resources 　　The time set aside for testing is not adequate to achieve complete coverage for all of the units. There is a lack of tools to support complete coverage • Other project related issues such as timing, scheduling. And marketing constraints.	**C311.2**	**BTL-1**
	37	37 Write the formula for cyclomatic complexity? The complexity value is usually calculated from control flow graph(G) by the formula. V(G) = E-N+2 Where The value E is the number of edges in the control flow graph The value N is the number of nodes.	**C311.2**	**BTL-1**
	38	38 List the various iterations of Loop testing. 　a. Zero iteration of the loop 　b. One iteration of the loop 　c. Two iterations of the loop 　d. K iterations of the loop where k<n 　e. n-1 iterations of the loop 　f. n+1 iterations of the loop	**C311.2**	**BTL-1**

39	39 Define test set. A test set T is said to be mutation adequate for program p provided that for every in equivalent mutant pi of p there is an element t in T such that pi[t] is not equal to p[t].	**C311.2**	**BTL-1**

40	40 What are the errors uncovered by black box testing? Incorrect or missing functions Interface errors Errors in data structures Performance errors Initialization or termination error	**C311.2**	**BTL-1**

41	41 List the levels of Testing or Phases of testing.
	
	  a. Unit Test
	
	  b. Integration Test
	  c. System Test
	  d. Acceptance Test
		C311.2	**BTL-1**

42	42 What are the basic primes for all structured program.
	
	  a. Sequential (e.g., Assignment statements)
	  b. Condition (e.g., if/then/else statements)
	  c. Iteration (e.g., while, for loops)
	  The graphical representation of these three primes are
	  given
		C311.2	**BTL-1**

43	43. What is Random Testing Random testing is a black-box software testing technique where programs are tested by generating random, independent inputs. Results of the output are compared against software specifications to verify that the test output is pass or fail.	**C311.2**	**BTL-1**
44	44. What is complexity Testing? Compatibility Testing is a type of Software testingto check whether your software is capable of running on different hardware, operating systems, applications , network environments or Mobile devices. Compatibility Testing is a type of the Nonfunctional testing Types: Forward Testing Backward Testing	**C311.2**	**BTL-1**
45	45 What is Test adequacy criterion? An adequacy criterion specifies a particular software testing requirement, and hence determines test cases to satisfy the requirement. It can be an explicit specification for test case selection, such as a set of guidelines for the selection of test cases.	**C311.2**	**BTL-4**

46	46. State feature of Static and Dynamic testing? Static and dynamic testing complement to one another and each type has a unique approach to detecting bugs. Static testing is a system of White Box testing where developers verify or check code to find fault. This type of testing is completed without executing the applications that are currently developed.	**C311.2**	**BTL-2**
47	47. What is functional testing? FUNCTIONAL TESTING is a type of software testingwhereby the system is tested against the functionalrequirements/specifications. Functions (or features) are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application.	**C311.2**	**BTL-1**
48	48. What are the different types of functional testing? In the types of functional testing following testing types should be cover: Unit Testing. Smoke testing. Sanity testing. Integration Testing. Interface Testing. System Testing. Regression Testing. UAT.	**C311.2**	**BTL-2**
49	49. What is functional automation testing? Functional testing is a form of automated testing that deals with how applications functions, or, in other words, its relation to the users and especially to the rest of the system. Traditionally, functional testing is implemented by a team of testers, independent of the developers.	**C311.2**	**BTL-2**
50	50. Is functional testing Black Box? Black Box Testing. BLACK BOX TESTING, also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or nonfunctional, though usually functional.	**C311.2**	**BTL-2**
51	51. What are the functional testing tools? Unified Functional Testing (UFT) is an automation tool introduced by HP (Hewlett Packard) for functional and regression testing. It's also known as the QTP (Quick Test Professional). The UFT is a combination of HP QTP (GUI testing tool) and HP Service Test (API testing tool)	**C311.2**	**BTL-1**
52	52. Is it possible to achieve 100 automation? Yes, it is very much possible. It is a tough target but definitely achievable. Mostly100% automation would be easily achievable when the application as a whole is in production or nearing completion. Complete automation would make sense when there won't be frequent changes occurring in the application	**C311.2**	**BTL-1**

S. No.	Question	Course Outcome	Blooms Taxonomy Level
53	53. What is alpha and beta testing? Alpha testing is the last testing done by the test teams at the development site after the acceptance testing and before releasing the software for beta test. Alpha testing can also be done by the potential users or customers of the application. But still, this is a form of in-house acceptance testing	**C311.2**	**BTL-1**
54	54.What is end to end testing? End-to-End Testing is a methodology used to test whether the flow of an application is performing as designed from start to finish. The purpose of carrying out end-to-end tests is to identify system dependencies and to ensure that the right information is passed between various system components and systems.	**C311.2**	**BTL-1**

PART B

S. No.	Question	Course Outcome	Blooms Taxonomy Level
1	Demonstrate black box test cases using equivalence class partitioning and boundary value analysis to test a module for payroll system (Apr May 2018) (Nov/Dec – 2016)	**C311.2**	**BTL-3**
2	Explain some of the black box testing approaches in detail	**C311.2**	**BTL-2**
3	Explain in detail about various white box testing approaches	**C311.2**	**BTL-2**
4	Explain in detail about equivalence class partitioning **(Apr May 2017) (Nov/Dec – 2018)**	**C311.2**	**BTL-2**
5	Explain about boundary value analysis in detail **(Apr May 2017) (Nov/Dec – 2018)**	**C311.2**	**BTL-2**
6	Explain in detail about state transition testing	**C311.2**	**BTL-2**
7	Explain in detail about test adequacy criteria	**C311.2**	**BTL-2**
8	Write a note on the following: (i)Positive and Negative Testing(5) (ii) Decision Tables(5) (iii) State based testing(6)	**C311.2**	**BTL-1**
9	Write short notes on the list given below: (i)Compatibility testing.(5) (ii)Documentation testing.(5) (iii) Domain testing.(6)	**C311.2**	**BTL-1**

10	Show how black box testing is performed in COTS components? (16)	C311.2	BTL3
11	Demonstrate the various black box test cases using equivalence class partitioning and boundary values analysis to test a module for ATM system.(16)	C311.2	BTL-2
12	Explain the various axioms that allow testers to valuate Test Adequacy Criteria.(16) **(Apr May 2017)**	C311.2	BTL5
13	What inference can you make from random testing, requirement based testing and domain testing explain?(16)	C311.2	BTL3
14	Explain the various axioms that allow testers to evaluate Test Adequacy Criteria.(16)	C311.2	BTL 4
15	(i)Discuss in detail about code coverage testing.(8) **(Nov/Dec – 2016)** (ii)Explain various white box testing with suitable test cases.(8) **(Nov/Dec – 2016)**	C311.2	BTL-2
16	Explain the significance of Control flow graph and Cyclomatic complexity in white box testing with a pseudo code for sum of positive numbers. Also mention the independent paths with test cases.(16)	C311.2	BTL 5
17	Explain about state transition testing . (Apr May 2018)	C311.2	BTL-2
18	Explain mutation testing with an example **(Apr May 2017)**	C311.2	BTL-2
19	How data flow testing aid in identifying defects in variable declaration and its use **(Apr May 2017)**	C311.2	BTL-2
20	Suppose you are testing defect coin problem artefacts. Identify the causes of various defects. What steps could have been taken to prevent the various classes of defects.	C311.2	BTL5
21	Discuss in detail about static testing and structural testing.Also mention the difference between these two testing concepts. **(Nov/Dec – 2018)**		

UNIT III   <u>**LEVELS OF TESTING**</u>

The need for Levers of Testing – Unit Test – Unit Test Planning – Designing the Unit Tests – The Test Harness – Running the Unit tests and Recording results – Integration tests – Designing Integration Tests – Integration Test Planning – Scenario testing – Defect bash elimination System Testing – Acceptance testing – Performance testing – Regression Testing – Internationalization testing – Adhoc testing – Alpha, Beta Tests – Testing OO systems – Usability and Accessibility testing – Configuration testing – Compatibility testing – Testing the documentation – Website testing

S. No.	Question	Course Outcome	Blooms Taxonomy Level
1	1.**What is Bottom up integration Testing? And its advantages. (Apr/May – 2018)** **Bottom-up testing** • Integrate individual components in levels until the complete system is created • Advantages and disadvantages • Architectural validation • Top-down integration testing is better at discovering errors in the system architecture • System demonstration • Top-down integration testing allows a limited demonstration at an early stage in the development • Test implementation • Often easier with bottom-up integration testing	C311.3	BTL-1
	• Test observation • Problems with both approaches. Extra code may be required to observe tests		
2	**2. Give example for security testing?** **(Apr/May – 2018)** It Evaluates system characteristics that relate to the availability, integrity and confidentiality of system data and services. Security Testing examples: password checking, legal and illegal entry with passwords, password expiration, encryption, browsing, trap doors, viruses.	C311.3	BTL-2
3	**3.Define integration testing?** **(Apr/May – 2017)** One unit at a time is integrated into a set of previously integrated modules which have passed a set of integration tests.	C311.3	BTL-1

4	4. What are the different types of system types? (Apr/May – 2017) • Functional testing • Performance testing • Stress testing • Configuration testing • Security testing • Recovery testing	C311.3	BTL-1
5	5.What is the need for different levels of testing? (Nov/Dec – 2016) Execution-based software testing, especially for large systems, is usually carried out at different levels. At each level there are specific testing goals .At the system level the system as a whole is tested and a principle goal is to evaluate attributes such as usability, reliability, and performance. To make sure that all the requirements are fulfilled, different levels of testing are needed.	C311.3	BTL-1
6	6.Define alpha, beta and acceptance tests. (Nov/Dec – 2016) When software is being developed for a specific client, acceptance tests are carried out after system testing. Alpha test: This test takes place at the developer's site. Beta Test: Beta tests ends the software to a cross-section of users who install it and use it under real world working conditions.	C311.3	BTL-1
7	7.What are the difference levels of testing? (Nov/Dec – 2018) The major phases of testing: unit test, integration test, system test, and some type of acceptance test.	C311.3	BTL-1
8	8.What is the importance of acceptance testing? During acceptance test the development organization must show that the software meets all of the client's requirements. Very often final payments for system development depend on the quality of the software as observed during the acceptance test.	C311.3	BTL-1
9	9.What is meant by software unit? A unit is the smallest possible testable software component.	C311.3	BTL-1

10	10.What are the characteristics of software unit? performs a single cohesive function; can be compiled separately; is a task in a work breakdown structure (from the manager's point of view); Contains code that can fit on a single page or screen.	C311.3	BTL-1
11	11.What are the steps required to perform unit testing? plan the general approach to unit testing; Design the test cases, and test procedures (these will be attached to the test plan); Define relationships between the tests; Prepare the auxiliary code necessary for unit test.	C311.3	BTL-1

12	**12.What do you mean by test harness?** The auxiliary code developed to support testing of units and components is called a test harness. The harness consists of drivers that call the target code and stubs that represent modules it calls.	**C311.3**	**BTL-1**
13	**13.What are the reasons for unit failure?** A fault in the test case specification (the input or the output was not specified correctly); A fault in test procedure execution (the test should be rerun); A fault in the test environment (perhaps a database was not set up properly); A fault in the unit design (the code correctly adheres to the design specification, but the latter is incorrect).	**C311.3**	**BTL-1**
14	**14. What is the need for test summary report?** This is a valuable document for the groups responsible for integrationandsystemtests.Itisalsoavaluablecomponentoftheproject history. Its value lies in the useful data it provides for test process improvement and defect prevention.	**C311.3**	**BTL-1**
15	**15.What are the goals of integration test?** To detect defects that occur on the interfaces of units; To assemble the individual units into working subsystems and finally a complete system that is ready for system test.	**C311.3**	**BTL-1**
16	**16.What do you mean by clusters?** A cluster consists of classes that are related, for example, they may work together (co-operate) to support a required functionality for the complete system	**C311.3**	**BTL-1**
17	**17.What are the documents required for integration test planning?** Requirements document, the user manual, and usage scenarios. These documents contain structure charts, state charts, data dictionaries, cross-reference tables, module interface descriptions, dataflow descriptions, messages and event descriptions, all necessary to plan integration tests.	**C311.3**	**BTL-1**
18	**18.Write notes on cluster test plan.** The plan includes the following items: Clusters this cluster is dependent on; A natural language description of the functionality of the cluster to be tested;	**C311.3**	**BTL-2**

	List of classes in the cluster; A set of cluster test cases.		
19	**19.Define functional testing.** Functional tests are black box in nature. The focus is on the inputs and proper outputs for each function. Improper and illegal inputs must also be handled by the system. System behavior under the latter circumstances tests must be observed. All functions must be tested	**C311.3**	**BTL-1**

20	**20.What do you mean by load generator?** An important tool for implementing system tests is a load generator. A load generator is essential for testing quality requirements such as performance and stress. A load is a series of inputs that simulates a group of transactions.	**C311.3**	**BTL-1**
21	**21.Define functional testing.** Functional tests are black box in nature. The focus is on the inputs and proper outputs for each function. Improper and illegal inputs must also be handled by the system. System behavior under the latter circumstances tests must be observed. All functions must be tested.	**C311.3**	**BTL-1**
22	**22.What is the goal of performance testing?** The goal of system performance tests is to see if the software meets the performance requirements. Testers also learn from performance test whether there are any hardware or software factors that impact on the system's performance. Performance testing allows testers to tune the system; that is, to optimize the allocation of system resources.	**C311.3**	**BTL-1**
23	**23 What are the resources required for performance testing?** A source of transactions to drive the experiments. An experimental test bed that includes hardware and software the system-under-test interacts with. Instrumentation or probes that help to collect the performance data. A set of tools to collect, store, process, and interpret the data.	**C311.3**	**BTL-1**
24	**24 What do you mean by regression testing? (Nov/Dec – 2018)** Regression testing is not a level of testing, but it is there testing of software that occurs when changes are made to ensure that the new version of the software has retained the capabilities of the old version and that no new defects have been introduced due to the changes.	**C311.3**	**BTL-1**
25	**25.List the levels of Testing or Phases of testing. (Nov/Dec – 2018)** Unit Test Integration Test System Test Acceptance Test	**C311.3**	**BTL-1**
26	**26.List the phases of unit test planning**. Unit test planning having set of development phases. Phase1: Describe unit test approach and risks. Phase 2: Identify unit features	**C311.3**	**BTL-1**

	to be tested. Phase 3: Add levels of detail to the plan.		

27	**27. What are the steps for top down integration?** Main control module is used as a test driver and stubs are substituted for all components directly subordinate to the main module. Depending on integration approach (Depth or breadth first) subordinate stubs are replaced one at a time with actual components. Tests are conducted as each component is integrated. The completion of each set of tests another stub is replaced with real component	**C311.3**	**BTL-1**
28	**28. Define stress Testing.** When a system is tested with a load that causes it to allocate its resources in maximum amounts .It is important because it can reveal defects in real-time and other types of systems which it will crash. This is sometimes called "breaking the system".	**C311.3**	**BTL-1**
29	**29 What are the two major requirements in the Performance testing?** Functional Requirement: User describe what functions the software should perform. We test for compliance of the requirement at the system level with the functional based system test. Quality Requirement: They are nonfunctional in nature but describe quality levels expected for the software.	**C311.3**	**BTL-1**
30	**30. What are the Integration strategies?** Top_ Down: In this strategy integration of the module begins with testing the upper level modules. Bottom_ Up: In this strategy integration of the module begins with testing the lowest level modules.	**C311.3**	**BTL-1**
31	**31.Define Test incident report**. The tester must determine from the test whether the unit has passed or failed the test. If the test is failed, the nature of the problem should be recorded in what is sometimes called a test incident report.	**C311.3**	**BTL-1**
32	**32.Define test case.** A use case is a pattern, scenario, or exemplar of usage. It describes a typical interaction between the software system under development and a user.	**C311.3**	**BTL-1**
33	**33. List the phases of unit test planning.** Unit test planning having set of development phases. Phase1: Describe unit test approach and risks. Phase 2: Identify unit features to be tested. Phase 3: Add levels of detail to the plan.	**C311.3**	**BTL-1**
34	**34. List the work of test planner.** • Identifies test risks. • Describes techniques to be used for designing the test cases for the units.	**C311.3**	**BTL-2**

	• Describe techniques to be used for data validation and recording of test results. • Describe the requirement for test harness and other software that interfaces with the unit to be tested, for ex, any special objects needed for testing object oriented.		
35	35. List the issues in class testing • Testing a class instance (an object) can validate a class in isolation. • When individually validated classes are used to create more complex classes in an application system, the entire subsystem must be tested as whole before it can be considered to be validated(integration testing). • Inheritance introduce problems that are not found in traditional software. • Test cases designed for base class are not applicable to derived class always (especially, when derived class is used in different context). Thus, most testing methods require some kind of adaptation in order to function properly in an OO environment	**C311.3**	**BTL-1**
36	36.Define test Harness. The auxiliary code developed into support testing of units and components is called a test harness. The harness consists of drivers that call the target code and stubs that represent modules it calls.	**C311.3**	**BTL-1**
37	37.Define Test incident report. The tester must determine from the test whether the unit has passed or failed the test. If the test is failed, the nature of the problem should be recorded in what is sometimes called a test incident report.	**C311.3**	**BTL-1**
38	**38 Define Summary report.** The causes of the failure should be recorded in the test summary report, which is the summary of testing activities for all the units covered by the unit test plan.	**C311.3**	**BTL-1**

39	**39 Goals of Integration test.**
	    • To detects defects that occur on the interface of the units.
	
	    • To assemble the individual units into working subsystems and finally a completed system that ready for system test.
		C311.3	**BTL-1**

40	**40 What are the Integration strategies?** ☐ Top_ Down: In this strategy integration of the module begins with testing the upper level modules. Bottom_ Up: In this strategy integration of the module begins with testing the lowest level modules.	**C311.3**	**BTL-1**

41	**41. Define load generator and Load.** An important tool for implementing system tests is a load generator. A load generator is essential for testing quality requirements such as performance and stress A load is a series of inputs that simulates a group of transactions. A transaction is a unit of work seen from the system user's view. A transaction consist of a set of operation that may be perform by a person , s/w system or device that is outside the system.	**C311.3**	**BTL-1**
42	**42, What are the two major requirements in the Performance testing.** • Functional Requirement: User describe what functions the software should perform. We test for compliance of the requirement at the system level with the functional based system test. • Quality Requirement: They are nonfunctional in nature but describe quality levels expected for the software.	**C311.3**	**BTL-1**
43	**43, What are the steps for top down integration?** • Main control module is used as a test driver and stubs are substituted for all components directly subordinate to the main module. • Depending on integration approach (Depth or breadth first) subordinate stubs are replaced one at a time with actual components. • Tests are conducted as each component is integrated. • The completion of each set of tests another stub is replaced with real component • Regression testing may be conducted to ensure that new errors have not been introduced.	**C311.3**	**BTL-1**
44	**44. What is AD HOC TESTING?** AD HOC TESTING, also known as Random Testingor Monkey Testing, is a method of software testingwithout any planning and documentation. The testsare conducted informally and randomly without any formal expected results. The tester improvises the steps and arbitrarily executes them (like a monkey typing while dancing).	**C311.3**	**BTL-1**

45	45. Why Ad hoc testing is done? Adhoc testing is an informal testing type with an aim to break the system. Thistesting is usually an unplanned activity. ... Ad hoc Testing does not follow any structured way of testing and it is randomly done on any part of application. Main aim of this testing is to find defects by random checking	**C311.3**	**BTL-1**
46	**46. What is Smoke testing?** **SMOKE TESTING**, also known as "Build Verification **Testing**", is a type of **software testing** that comprises of a non-exhaustive set of **tests** that aim at ensuring that the most important functions work. The result of this **testing** is used to decide if a build is stable enough to proceed with further **testing**.	**C311.3**	**BTL-1**
47	**47. What is 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. **Compatibility Testing** is a type of the Nonfunctional **testing**	**C311.3**	**BTL-1**

48	48. What is application compatibility testing? **Compatibility** is a non- functional **testing** to ensure customer satisfaction. It is to determine whether your software **application** or product is proficient enough to run in different browsers, database, hardware, operating system, mobile devices, and networks.	**C311.3**	**BTL-1**
49	49. Is compatibility testing functional or nonfunctional? **Non functional testing** ensures that a system/application meets the specified performance requirements. In **non functional** software **testing**, by performance we do not only mean response time, but several other factors such as security, scalability and usability of the application as well	**C311.3**	**BTL-1**
50	50. What is responsive testing? A **responsive** web design involves creating a flexible web page that is accessible from any device, starting from a mobile phone to a tablet. ... Software testers may find it challenging to perform **responsive** design **testing** as a variety of factors are to be looked into during the **testing** phase	**C311.3**	**BTL-2**

PART B

S. No.	Question	Course Outcome	Blooms Taxonomy Level
1	What do you mean by unit testing? Explain in detail about the process of unit testing and unit test planning **(Apr May 2018) (Nov/Dec – 2018)**	**C311.3**	BTL-2
2	Write the importance of security testing and explain the consequences of security breaches, also write the various areas which has to be focused on during security testing. **(Apr May 2018)**	**C311.3**	BTL-2 BTL-2
3	Write notes on configuration testing and its objectives. **(Apr May 2018)**	**C311.3**	BTL-2

4	State the need for integration testing in procedural code **(Apr May 2018) (Nov/Dec – 2016)**	**C311.3**	**BTL-1**
5	Explain in detail about test harness. Also write notes on integration test.	**C311.3**	**BTL-2**
6	Explain various system testing approaches in detail. **(Nov/Dec – 2018)**	**C311.3**	**BTL-2**
7	Write notes on regression testing, alpha and beta acceptance testing strategies	**C311.3**	**BTL-2**
8	Write notes on configuration testing and compatibility testing How Would you identify the hardware and software for	**C311.3**	**BTL-2**

	configuration testing? **(Nov/Dec – 2016)**		
9	Write notes on usability, accessibility and website testing approaches	**C311.3**	**BTL-2**
10	Give the most effective ad hoc testing techniques.	**C311.3**	**BTL-2**
11	Show the approaches you use to do website testing. **(Nov/Dec – 2016)** Can you judge on the reason for system testing	**C311.3**	**BTL3**
12	Analyze on when to do the regression testing and smoke testing?	**C311.3**	**BTL-2**
13	Compare and contrast Alpha and Beta Testing with suitable example.	**C311.3**	**BTL 5**
14	Prepare the role of test data generators in testing Object Oriented System and explain Object Oriented Testing.	**C311.3**	**BTL3**
15	Show the test cases applied for acceptance testing	**C311.3**	**BTL 5**
16	Explain the significance of control flow graph and Cyclomatic complexity in white box testing with a pseudo code for sum of positive numbers. Also mention the independent paths with test cases.**(Apr. May 2017)**	**C311.3**	**BTL3**
17	Explain the black box testing techniques with example .**(Apr. May 2017)**	**C311.3**	**BTL-2**
18	**Case Study: Several kinds of tests for a web application. How to improve regression testing (Nov/Dec – 2018)**		

	UNIT IV	TEST MANAGEMENT		

People and organizational issues in testing – Organization structures for testing teams – testing services – Test Planning – Test Plan Components – Test Plan Attachments – Locating Test Items – test management – test process – Reporting Test Results – The role of three groups in Test Planning and Policy Development – Introducing the test specialist – Skills needed by a test specialist – Building a Testing Group.

S. No.	Question	Course Outcome	Blooms Taxonomy Level
1	1. Define work breakdown structure? (Apr/May – 2018) A Work Breakdown Structure is a hierarchical or treelike representation of all the tasks that are required to complete a project. And the elements are 1. Project startup 2. Management coordination 3. Tool selection 4. Test planning 5. Test design 6. Test development 7. Test execution 8. Test measurement, and monitoring 9. Test analysis and reporting 10. Test process improvement	C311.4	BTL-1
2	2.What is Test item transmittal report? (Apr/May – 2018) It identifies the test items being transmitted for testing in the event that separate development and test groups are involved or in the event that a formal beginning of test execution is desired	C311.4	BTL-1
3	3.What are the responsibilities of a test specialist? (Apr/May – 2017) Their primary responsibility is to ensure that testing is effective and productive, and that quality issues are addressed.	C311.4	BTL-1
4	4.What is the role of test manager? **(Apr/May – 2017) (Nov/Dec – 2016)** The test manager is usually responsible for test policy making, customer interaction, test planning, test documentation, controlling and monitoring of tests, training, test tool acquisition, participation in inspections and walkthroughs, reviewing test work, the test repository, and staffing issues such as hiring, firing, and evaluation of the test team members.	C311.4	BTL-1

5	5.Write the approaches to test cost Estimation? **(Apr/May – 2017)** The COCOMO model and heuristics • Use of test cost drivers • Test tasks • Tester/developer ratios	C311.4	BTL-2

	☐ Expert judgment		
6	6 .What are the three critical groups in testing planning and test plan policy? **(Nov/Dec – 2016)** • Managers: Task forces, policies, standards, planning Resource allocation, support for education and training, Interact with users/Clients • Developers/Testers: Apply Black box and White box methods, test at all levels, assist with test planning, Participate in task forces. • Users/Clients: Specify requirement clearly, Support with operational profile, Participate in acceptance test planning.	C311.4	BTL-1
7	7.Define Goal. A goal can be described as a statement of intent or a statement of an accomplishment that an individual or an organization wants to achieve.	C311.4	BTL-1
8	8.Define policy. A Policy can be defined as a high-level statement of principle or course of action that is used to govern a set of activities in an org.	C311.4	BTL-1
9	9.Define Plan. A plan is a document that provides a framework or approach for achieving a set of goals.	C311.4	BTL-1
10	10.List the Test plan components. • Test plan identifier • Introduction • Items to be tested • Features to be tested • Approach • Pass/fail criteria • Suspension and resumption criteria	C311.4	BTL-1
11	11.Mention some of the personal and managerial skills required by a test specialist. • Organizational, and planning skills; • The ability to keep track of, and pay attention to, details; • The determination to discover and solve problems; • The ability to work with others and be able to resolve conflicts; • The ability to mentor and train others; • The ability to work with users and clients; • Strong written and oral communication skills; • The ability to work in a variety of environments; • The ability to think creatively.	C311.4	BTL-2

12	12.How a test group can be formed? Upper management must support the decision to establish a test group and commit resources to the group. Using interviews the test group members can be identified. The skill level as well as the education required have to be decided before the interview.	C311.4	BTL-1
13	13.What is the role of test lead? The test lead assists the test manager and works with a team of test engineers on individual projects. He or she may be responsible for duties such as test planning, staff supervision, and status reporting. The test lead also participates in test design, test	C311.4	BTL-1

	execution and reporting, technical reviews, customer interaction, and tool training.		
14	14.What is the role of test engineer? The test engineers design, develop, and execute tests, develop test harnesses, and set up test laboratories and environments. They also give input to test planning and support maintenance of the test and defect repositories.	C311.4	BTL-1
15	15.Write notes on junior test engineer. The junior test engineers are usually new hires. They gain experience by participating in test design, test execution, and test harnessdevelopment. They may also be asked to review user manuals and user help facilities defect and maintain the test and defect repositories.	C311.4	BTL-1
16	16.Define milestones. Milestones are tangible events that are expected to occur at a certain time in the project's lifetime. Managers use them to determine project status.	C311.4	BTL-2
17	17.What are the components of test planner? • Overall test objectives • What to test (scope of the tests) ☐ Who will test? • Howtotest? • When to test? • When to stop testing?	C311.4	BTL-1
18	18.What do you mean by test plan identifier? Each test plan should have a unique identifier so that it can be associated with a specific project and become a part of the project history.	C311.4	BTL-1

19	19.What are the technical skills required by a test specialist? • Strong coding skills and an understanding of code structure and behavior; • A good understanding of testing principles and practices; • A good understanding of basic testing strategies, methods, and techniques; • A knowledge of process issues; • Knowledge of how networks, databases, and operating systems are organized and how they work.	C311.4	BTL-1
20	20.What are features? Features may be described as distinguishing characteristics of a software component or system	C311.4	BTL-1
21	21.What is item pass/fail criteria? Given a test item and a test case, the tester must have a set of criteria to decide on whether the test has been passed or failed upon execution	C311.4	BTL-1
22	22.What do you mean by failure? A failure occurs when the actual output produced by the software does not agree with what was expected, under the conditions	C311.4	BTL-1

	specified by the test.		
23	23. Give some examples of test deliverables. Execution-basedtestinghasasetofdeliverablesthatincludesthetestplan along with its associated test design specifications, test harness, test procedures, and test cases	C311.4	BTL-1
24	24.What is cost driver? A cost driver can be described as a process or product factor that has an impact on overall project costs.	C311.4	BTL-1
25	25.What is meant by test procedure? A procedure in general is a sequence of steps required to carry out a specific task.	C311.4	BTL-1
26	26.Write the WBS elements for testing. • Project startup • Management coordination • Tool selection • Test planning • Test design • Test development • Test execution • Test measurement, and monitoring • Test analysis and reporting • Test process improvement	C311.4	BTL-2

27	27. Write short notes on Cost driver. A Cost driver can be described as a process or product factor that has an impact on overall project costs. Cost drivers for project the include Product attributes such as the required level of reliability, Hardware attributes such as memory constraints, Personnel attributes such as experience level and Project attributes such as tools and methods.	**C311.4**	**BTL-2**
28	28 Write the WBS elements for testing. 1. Project startup 2. Management coordination 3. Tool selection 4. Test planning 5. Test design 6. Test development 7. Test execution 8. Test measurement, and monitoring 9. Test analysis and reporting 10. Test process improvement.	**C311.4**	**BTL-2**
29	29. Define Test incident Report. The tester should record in attest incident report (sometimes called a problem report) any event that occurs during the execution of the tests that is unexpected, unexplainable, and that requires a follow- up investigation.	**C311.4**	**BTL-1**
30	30 Write the test term hierarchy? Test Manager Test leader Test Engineer Junior Test Engineer	**C311.4**	**BTL-2**
31	31. Write notes on risk in testing process.	**C311.4**	**BTL-2**

	Testing software with a high degree of criticality, complexity, or a tight delivery deadline all impose risks that may have negative impacts on project goals. These risks should be: (i) identified, (ii) evaluated in terms of their probability of occurrence, (iii) prioritized, and (iv) contingency plans should be developed that can be activated if the risk occurs. People and organizational issues in testing – Organization structures for testing teams – testing services – Test Planning – Test Plan Components – Test Plan Attachments – Locating Test Items – test management – test process – Reporting Test Results – The role of three groups in Test Planning and Policy Development – Introducing the test specialist – Skills needed by a test specialist – Building a Testing Group		

		C311.4	BTL-1

	32,Write the different types of goals?
	
	   i. Business goal: To increase market share 10% in the next 2 years in the area of financial software
	
	   ii. Technical Goal: To reduce defects by 2% per year over the next 3 years.
	
	   iii. Business/technical Goal: To reduce hotline calls by 5% over the next 2 years
	
	   iv. Political Goal: To increase the number of women and minorities in high management positions by 15% in the next 3 years.		

32			

	33,Define Goal and Policy
	      A goal can be described as (i) a statement of intent or (ii) a statement of a accomplishment that an individual or an org wants to achieve.
	
	      A Policy can be defined as a high-level statement of principle or course of action that is used to govern a set of activities in an org.
		C311.4	BTL-1

33			

	34,List the Test plan components. /
	
	Duties of component wise testing teams (Nov/Dec – 2018)
	   • Test plan identifier
	   • Introduction
	   • Items to be tested
	   • Features to be tested
		C311.4	BTL-1

34			

	• Approach • Pass/fail criteria • Suspension and resumption criteria • Test deliverables • Testing Tasks • Test environment • Responsibilities • Staffing and training needs • Scheduling • Risks and contingencies ☐ Testing costs • Approvals.		
35	35,Define a Work Breakdown Structure.(WBS) A Work Breakdown Structure (WBS) is a hierarchical or treelike representation of all the tasks that are required to complete a project	**C311.4**	**BTL-1**
36	36,Write the approaches to test cost Estimation? • The COCOMO model and heuristics • Use of test cost drivers ☐ Test tasks • Tester/developer ratios • Expert judgment	**C311.4**	**BTL-2**
37	37. What is test effectiveness? Let us look at what Test effectiveness is, Test effectiveness of a technique or a system or a team is the ability to find defects and isolate them, from a product or deliverable. Test effectiveness is to ensure quality and close the two quality gaps, namely producer's quality gap and customer's quality gap.	**C311.4**	**BTL-1**
38	38,Write short notes on Cost driver. A Cost driver can be described as a process or product factor that has an impact on overall project costs. Cost drivers for project the include • Product attributes such as the required level of reliability • Hardware attributes such as memory constraints. • Personnel attributes such as experience level. ☐ Project attributes such as tools and methods.	**C311.4**	**BTL-1**

			C311.4	BTL-1
39	39,Write the WBS elements for testing. 1. Project startup 2. Management coordination 3. Tool selection 4. Test planning 5. Test design			
	6. Test development 7. Test execution 8. Test measurement, and monitoring 9. Test analysis and reporting 10. Test process improvement			
40	40,What is the function of Test Item Transmittal Report or Locating Test Items Suppose a tester is ready to run tests on the data described in the test plan. We needs to be able to locate the item and have knowledge of its current status. This is the function of the Test Item Transmittal Report. Each Test Item Transmittal Report has a unique identifier.		C311.4	BTL-1
41	41,What is the information present in the Test Item Transmittal Report or Locating Test Items 1) Version/revision number of the item 2) Location of the item 3) Person responsible for the item (the developer) 4) References tyo item documentation and test plan it is related to. 5) Status of the item 6) Approvals – space for signatures of staff who approve the transmittal.		C311.4	BTL-1
42	42,What is the need of Test Incident Report **(Nov/Dec – 2018)** The tester should record in attest incident report (sometimes called a problem report) any event that occurs during the execution of the tests that is unexpected , unexplainable, and that requires a follow- up investigation.		C311.4	BTL-2
			C311.4	BTL-1

43	43,What are the Three critical groups in testing planning and test plan policy ? • Managers: Task forces, policies, standards, planning Resource allocation, support for education and training, Interact with users/Clients • Developers/Testers :Apply Black box and White box methods, test at all levels, Assst with test planning, Participate in task forces. • Users/Clients :Specify requirement clearly, Support with operational profile, Participate		

	in acceptance test planning		

44	44, What are the skills needed by a test specialist?
	• Personal and managerial Skills
	    Organizational, and planning skills, work with others, resolve conflicts, mentor and train others, written /oral communication skills, think creatively.
	• Technical Skills
	    General software engineering principles and practices, understanding of
	
	    testing principles and practices, ability to plan, design, and execute test cases, knowledge of networks, database, and operating System.
		C311.4	**BTL-1**

45	45. What does a test plan consists of? A test plan is a document that defines the strategy that will be used to verify that the product or system is developed according to its specifications and requirements. It describes the scope of testing, testing techniques to be used, resources required for testing and the schedule of intended test activities.	**C311.4**	**BTL-1**

46	46, How do you write a test plan document?
	    Write the introduction. ...
	    Define your objectives. ...
	    Write a section on required resources. ...
	    Write a section on risks and dependencies. ...
	    Write a section on what you are going to test. ...
	    Write a section on what you will not be testing. ...
	    List your strategy. ...
	    Develop pass/fail criteria.
		C311.4	**BTL-1**

47	47. What is Test Plan? . A TEST PLAN is a document describing software testing scope and activities. It is the basis for formally testing any software/product in a project. ISTQB Definition. test plan: A document describing the scope, approach, resources and schedule of intended test activities.	**C311.4**	**BTL-2**
48	48, What is difference between test strategy and test plan? Test strategy is a high level document which defines the approach for softwaretesting. It is basically derived from the Business Requirement document. Test strategy is developed by project manager or business analyst. It is kind of static document which sets the standards for testing so not updated often	**C311.4**	**BTL-1**
49	49, What is test plan test strategy? A Test Strategy document is a high level document and normally developed by project manager. This document defines "Software Testing Approach" to achievetesting objectives. ... Some companies include the "Test Approach" or "Strategy"	**C311.4**	**BTL-1**
	inside the Test Plan, which is fine and it is usually the case for small projects		
50	50, What are test metrics? In software testing, Metric is a quantitative measure of the degree to which a system, system component, or process possesses a given attribute. In other words,metrics helps estimating the progress, quality and health of a software testingeffort.	**C311.4**	**BTL-1**
51	51, What is KPI in testing? A Key Performance Indicator (or KPI) is usually used to evaluate the software process efficiency evaluation. The important parameters and their usage are analysed and the outcome of the measurement is used to trigger any process improvements.	**C311.4**	**BTL-1**

PART B

S. No.	Question	Course Outcome	Blooms Taxonomy Level
1	(i) Describe about the testing team structure for single product companies **(Apr May 2018) (Nov/Dec – 2016)** (ii)What are the skills needed for a test specialist. **(Nov/Dec – 2016)**	**C311.4**	**BTL-6** **BTL-1**
2	Name the reports of test results and the contents available in each test reports **(Apr May 2018)**	**C311.4**	**BTL-2**
3	Analyze the various steps in forming the test group. / How will you build a testing group discuss with an example. **(Nov/Dec – 2018)**	**C311.4**	**BTL-6** **BTL-1**
4	(i) Demonstrate on various stages of test plan. (ii) Illustrate the role of testing.	**C311.4**	**BTL-2** **BTL-3**

5	(i) Develop the challenges and issues faced in testing service organization also write how we can eliminate challenges. (ii) Can you list the components of test plan in detail. **(Apr May 2018) (Nov/Dec – 2016)** **Illustrate the various components of Test plan with an example (Nov/Dec – 2018)**	**C311.4**	**BTL-4** **BTL-1** **BTL-2**
6	Demonstrate the test management based on standards, infrastructure, people and product.	**C311.4**	**BTL-2**
7	Describe with example test people management. (8)	**C311.4**	BTL-4
8	Analyze on few typical resources that are considered when test planning.	**C311.4**	BTL-4
9	Point out the five stages in a test plan process.	**C311.4**	BTL-4
10	Examine purpose of Test Transmitted report and the test log	**C311.4**	**BTL-1**

11	Analyze the role of manager in support of test group	**C311.4**	BTL-4
12	List the various skills needed by a test specialist. **(Apr May 2018)**	**C311.4**	**BTL-1**
13	How would you estimate the measurements for monitoring error, faults and failures?	**C311.4**	**BTL-2**
14	Write the reason to create work break down structure	**C311.4**	**BTL-2**
15	Point out the five stages in a test plan process./ Explain the concepts of test planning in detail. Also mention the way of defining test plan **(Nov/Dec – 2018)**	**C311.4**	**BTL-1**
16	Compare and contrast the role of debugging goals and policies in testing. **(Nov/Dec – 2016)**	**C311.4**	**BTL-2**

Software test automation – skill needed for automation – scope of automation – design and architecture for automation – requirements for a test tool – challenges in automation – Test metrics and measurements – project, progress and productivity metrics
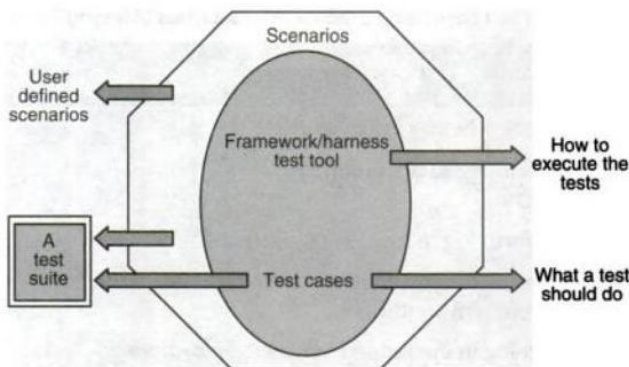
S. No.	Question	Course Outcome	Blooms Taxonomy Level
1	**1. What is Walk throughs?** **(Apr/May – 2018)** • Type of technical review where the producer of the reviewed material serves as the review leader and actually guides the progression of the review (as a review reader) • Traditionally applied to design and code • In the case of code walkthrough, test inputs may be selected and review participants then literally walk through the design or code • Checklist and preparation steps may be eliminated	C311.5	BTL-1
2	**2 What are the general goals for the reviewers** **(Apr/May – 2018)** The general goals for the reviewers • identify problem components or components in the software artifact that need improvement; • identify components of the software artifact that do not need improvement; • identify specific errors or defects in the software artifact (defect detection); • ensure that the artifact conforms to organizational standards. the many benefits of a review program are: • higher-quality software; • increased productivity (shorter rework time); • closer adherence to project schedules (improved process control); • increased awareness of quality issues; • teaching tool for junior staff; • opportunity to identify reusable software artifacts;	C311.5	BTL-1
3	**3. What are stress and load tools?** **(Apr/May – 2017)** Stress and load tools induce stresses and loads to the software being tested. A word processor running as the only application on the system, with all available memory and disk space, probably works just fine.	C311.5	BTL-1

4	4.What are calculated metrics? (Apr/May – 2017) Calculated Metrics are derived from the data gathered in Base Metrics. These Metrics are generally tracked by the test lead/manager for Test Reporting purpose.	C311.5	BTL-1
5	5.What are the benefits of testing tools and automation? (Nov/Dec – 2016) Speed, Efficiency, Accuracy and Precision, Relentlessness.	C311.5	BTL-1
6	6.Define Base line. (Nov/Dec – 2016) Base lines are formally reviewed and agreed upon versions of software artifacts, from which all changes are measured. They serve as the basis for further development and can be changed only through formal change procedures.	C311.5	BTL-2
7	7.What is the use of viewers and monitors test tool? A viewer or monitor test tool allows you to see details of the software's operation that you wouldn't normally be able to see.	C311.5	BTL-1
8	8.What are Drivers tool? Drivers are tools used to control and operate the software being tested. One of the simplest examples of a driver is a batch file, a simple list of programs or commands that are executed sequentially	C311.5	BTL-1
9	9. What are stubs? Stubs are essentially the opposite of drivers in that they don't control or operate the software being tested; they instead receive or respond to data that the software sends.	C311.5	BTL-1
10	10.What are Interference Injectors and Noise Generators? Another class of tools is interference injectors and noise generators. They're similar to stress and load tools but are more random in what they do. The Stress tool, for example, has an executor mode that randomly changes the available resources	C311.5	BTL-1
11	11.What are programmed macros? Programmed macros are a step up in evolution from the simple record and playback variety. Rather than create programmed macros by recording your actions as you run the test for the first time, create them by programming simple instructions for the playback system to follow.	C311.5	BTL-1
12	12.How verification can be performed using test automation tools? Screen captures Control values File and other output	C311.5	BTL-1

13	**13.What are Software Testing Metrics?** A Metric is a quantitative measure of the degree to which a system, system component, or process possesses a given attribute. Software Metrics are used to measure the quality of the project. Simply, Metric is a unit used for describing an attribute. Metric is a scale for measurement.	**C311.5**	**BTL-1**

14	**14.What is Software Test Measurement?** Measurement is the quantitative indication of extent, amount, dimension, capacity, or size of some attribute of a product or process.	**C311.5**	**BTL-1**
15	**15.Give some examples of test metrics.** • How many defects are existed within the module? • How many test cases are executed per person? • What is the Test coverage %?	**C311.5**	**BTL-2**
16	**16.What is the need for test metrics?** Test Metrics are used to, Take the decision for next phase of activities such as, estimate the cost & schedule of future projects. Understand the kind of improvement required to success the project Take decision on process or technology to be modified etc.	**C311.5**	**BTL-1**
17	**17.What are the types of manual test metrics?** Base metrics Calculated metrics	**C311.5**	**BTL-1**
18	**18.What are base metrics?** Base Metrics are the Metrics which are derived from the data gathered by the Test Analyst during the test case development and execution. This data will be tracked throughout the Test Life cycle. I.e. collecting the data like, Total no. of test cases developed for a project (or) no. of test cases need to be executed (or) no. of test cases passed/failed/blocked etc.	**C311.5**	**BTL-1**
19	**19.Tell about test case execution productivity?** It determines the number of Test cases / Scripts that can be prepared per person days of effort.	**C311.5**	**BTL-1**
20	**20.Define test automation. / What is the need of test automation (Nov/Dec – 2018)** In software testing, test automation is the use of special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes. It reduces human effort and reduce human error.	**C311.5**	**BTL-1**

21	**21.Classify the skills needed for automation.**	C311.5	BTL-2
	The skills needed for automation is classified into four levels in three generation.		

	Automation—first generation	Automation—second generation	Automation—third generation	
	Skills for test case automation	Skills for test case automation	Skills for test case automation	Skills for framework
	Scripting languages	Scripting languages	Scripting languages	Programming languages
	Record-playback tools usage	Programming languages	Programming languages	Design and architecture skills for framework creation
		Knowledge of data generation techniques	Design and architecture of the product under test	Generic test requirements for multiple products
		Usage of the product under test	Usage of the framework	

22	**22.Draw the framework for test automation.**	C311.5	BTL-3

23	$Q_o = [C^T : A^T C^T : \cdots (A^T)^{n-1} C^T]$	C311.5	BTL-1
	23.What are configuration file modules?		
	A configuration file contains a set of variables that are used in automation. A configuration file is important for running the test cases for various execution conditions and for running the test for various input and output conditions and states.		

24	**24.Define Review.**	C311.5	BTL-1
	Review is a group meeting whose purpose is to evaluate a software artifact or a set of software artifacts.		

25	**25.Define SCM (Software Configuration management).**	C311.5	BTL-1
	Software Configuration Management is a set of activities carried out for identifying, organizing and controlling changes throughout the lifecycle of computer software.		

26	**26.Define Project Controlling.** It consists of developing and applying a set of corrective actions to get a project on track when monitoring shows a deviation from what was planned.	**C311.5**	**BTL-1**
27	**27.What are the benefits of a Review program?** • Higher quality software • Increased productivity • Increased awareness of quality issues • Reduced maintenance costs • Higher customer satisfaction	**C311.5**	**BTL-1**
28	**28. What are the various types of Reviews?** • Inspections • Walkthroughs	**C311.5**	**BTL-1**
29	**29. What is Inspections?** It is a type of review that is formal in nature and requires prereview preparation on the part of the review team. The Inspection leader prepares is the checklist of items that serves as the agenda for the review.	**C311.5**	**BTL-1**
30	**30. List out the members present in the Review Team.** • SQA(Software Quality Assurance) staff • Testers • Developers • Users /Clients. • Specialists.	**C311.5**	**BTL-1**
31	**31. What are the advantages of automation?** Automation saves time, reliable, helps immediate testing, better utilization of global resources.	**C311.5**	**BTL-1**
32	**32. What is a "test suite"?** A test suite is a set of test cases that are automated and scenarios that are associated with the test cases.	**C311.5**	**BTL-1**
33	33. Define baseline testing. Definition: Baseline testing refers to the validation of the documents and specifications on which test cases are designed. ... Many problems are discovered and solved during baseline testing. Description: Baseline testing is a type of nonfunctional testing which is generally performed by testing engineers.	**C311.5**	**BTL-1**
34	34. What is the difference between benchmark and baseline? Baseline and benchmark are similar but distinct activities. Figuratively, a baselineis a "line in the sand" for an organization whereby it measures important performance characteristics for future reference. ... a benchmark is about assessing the relative performance of an application	**C311.5**	**BTL-1**

35	35. What is a performance baseline? The Performance Measurement Baseline is a time–phased schedule of all the work to be performed, the budgeted cost for this work, and the organizational elements that produce the deliverables from this work.	**C311.5**	**BTL-1**

36	36. Which tools are used for manual testing?
	    Selenium (Web Application Testing)
	    Appium (Mobile Testing)
	    JMeter (Load Testing)
	    Jenkins (Continuous Testing)
	    TestLink (Test Management)
	    Mantis (Bug-Tracking & Project Management)
	    Postman (API Testing)
	    Firebug / Firepath (Online Debugging)
		C311.5	**BTL-1**

37	37, What is the difference between manual and automated testing? In manual testing (as the name suggests), test cases are executed manually (by a human, that is) without any support from tools or scripts. But with automated testing, test cases are executed with the assistance of tools, scripts, and software.Testing is an integral part of any successful software project	**C311.5**	**BTL-1**
38	38. What is the meaning of penetration testing tools? A penetration testing tool is a tool used for testing the security of a Web application. ... Fuzzing refers to a testing technique that is highly automated, which covers several boundary cases by means of invalid data as the application input to make sure that exploitable vulnerabilities are absent.	**C311.5**	**BTL-2**
39	39, Why manual testing is important? Manual testing is important because it's good to test the application randomly/throughly. In initial stage of of the application we can not do the ad-hoc testing by execution of scripts. We need to do this Manually. Manual testing is important till application is in regression phase or stable	**C311.5**	**BTL-1**
40	40, Is it possible to do performance testing manually? In order to execute performance test manually, in this case, many active sessions of the same application should be open for testing. ... Usually, the test is performed by deploying the application on the server, accessing the application from various client machines and allowing multiple threads to run	**C311.5**	**BTL-1**

41	41, Can Automation Testing replace manual testing? However, test automation does not have the ability to fully replace manual testing. ... However, you cannot expect test automation to perform all the work done manually by a tester. The testing process of a new software application canbe divided into two major phases	**C311.5**	**BTL-1**

42	42. List the the challenges in automation?
	        1) Testing the complete application: ...
	        2) Misunderstanding of company processes: ...
	        3) Relationship with developers: ...
	        4) Regression testing: ...
	        5) Lack of skilled testers: ...
	        6) Testing always under time constraint:
		C311.5	**BTL-1**

43	43, What is UI automation testing? Automated tests that drive your application through its user interface (UI) are known as coded UI tests (CUITs) in Visual Studio. These tests include functionaltesting of the UI controls. They let you verify that the whole application, including itsuser interface, is functioning correctly	**C311.5**	**BTL-1**

44	44, Write some tips to improve automation testing?
	
	    Decide What to Automate.
	    Prioritize & Divide Tasks.
	    Prepare Test Cases & Scenario Beforehand.
	    Identify the Process.
	    Create Tests that Don't Affect UI Changes.
	    Utilize Quality Test Data.
		C311.5	**BTL-1**

	Utilize Standardized Test Tools, Frameworks & Techniques. Don't Automate Every Test.		

45	45, List some Benefits of Automation Testing.
	
	    Fast: Runs tests significantly faster than human users.
	    Repeatable: Testers can test how the website or software reacts after repeated execution of the same operation.
	     Reusable: Tests can be re-used on different versions of the software.
		C311.5	**BTL-1**

46	46, What skills are needed to be a good test automation tester? Following are the technical skills a manual tester should master to become a brilliant automation testing engineer: Test architecture. Test design. Performance testing. Configuration management. Manual testing agility & interaction. Communication between teams. Troubleshooting. Agile, DevOps, and continuous delivery	**C311.5**	**BTL-1**
47	47, What is test coverage metrics? Test coverage (also referred to by some as code coverage) is one of many metricsthat are commonly used to give a statistical representation of the state of the code written for a certain piece of software	**C311.5**	**BTL-2**
48	48, What is test effectiveness? Let us look at what Test effectiveness is, Test effectiveness of a technique or a system or a team is the ability to find defects and isolate them, from a product or deliverable. Test effectiveness is to ensure quality and close the two quality gaps, namely producer's quality gap and customer's quality gap 49, What is defect density in software testing? Defect Density is the number of defects confirmed in software/module during a specific period of operation or development divided by the size of thesoftware/module. It enables one to decide if a piece of software is ready to be released. Defect density is counted per thousand lines of code also known as KLOC.	**C311.5**	**BTL-1**
49	49, What is KPI in testing? A Key Performance Indicator (or KPI) is usually used to evaluate the software process efficiency evaluation. The important parameters and their usage are analysed and the outcome of the measurement is used to trigger any process improvements	**C311.5**	**BTL-1**
50	**What is Code Walk Through review?** Code Walkthrough is a form of peer review in which a programmer leads the review process and the other team members ask questions and spot possible errors against development standards and other issues. The meeting is usually led by the author of the document under review and attended by other members of the team.	**C311.5**	**BTL-1**

S. No.	Question	Course Outcome	Blooms Taxonomy Level
	What is Progress metrics? (Nov/Dec – 2018) In software testing, Metric is a quantitative measure of the degree to which a system, system component, or process possesses a given attribute. In other words, metrics helps estimating the progress, quality and health of a software testing effort.	C311.5	BTL-1
51			

Part B & C

S. No.	Question	Course Outcome	Blooms Taxonomy Level
1	Discuss the types of review. Explain various components of review plans. **(Apr May 2018)**	C311.5	BTL-2
2	Explain various requirements for test tool.	C311.5	BTL-2
3	(i) Explain the design and architecture for automation **(Nov/Dec – 2016)**	C311.5	BTL-2
	(ii) List and discuss metrics that can be used for detection prevention and how	C311.5	BTL-2
4	Narrate and formulate about the metrics of parameters to be considered for evaluating the software quality. **(Apr May 2018)**	C311.5	BTL-2 BTL-5
5	Explain in detail about skills needed for automation and give its challenges.	C311.5	BTL-2
6	How metrics are classified? Demonstrate project metrics.	C311.5	BTL-2
7	a.Outline the challenges in automation. **(Nov/Dec – 2016)** b. What is the need for metrics in testing? Analyze about Productivity metrics. **(Nov/Dec – 2016)**	C311.5	BTL-1 BTL-4
8	Briefly explain the test tool selection procedure. **(Nov/Dec – 2018)**	C311.5	BTL-2
9	Discuss the significance of various measurements in the testing process.	C311.5	BTL-3
10	(i)List the requirements for test tool. Explain with suitable examples. (8) (ii)Why testing in metrics? Analyze about Productivity metrics. (8)	C311.5	BTL-3

11	What are the key benefits in using metrics in product development and testing. (8) What are the steps involved in a metrics program. Briefly explain each step. (8)	**C311.5**	**BTL-2**
12	What is the purpose of progress metrics? Describe in detail	**C311.5**	**BTL-1**
13	How would you classify the measurements in productivity metrics. Summarize it.	**C311.5**	**BTL-2**
14	What are metrics and measurements? illustrate the types of product metrics	**C311.5**	**BTL-1**
15	Explain the various generations of automation and the required skills for each. **(Apr May 2017)**	**C311.5**	**BTL-2**
16	Explain the different types of Test defect metrics under progress metrics based on what they measure and what area they focus on. **(Apr May 2017)**	**C311.5**	**BTL-2**
17	Write short notes on: **(Nov/Dec – 2018)** Classification of automation testing Scope of an automatuion	**C311.5**	**BTL-2**