



Roll No.

**PRESIDENCY UNIVERSITY  
BENGALURU**

**SCHOOL OF ENGINEERING**

**TEST 1**

**Sem & AY:** Odd Sem. 2019-20

**Date:** 30.09.2019

**Course Code:** CSE 403

**Time:** 1:00PM to 2:00 PM

**Course Name:** SOFTWARE TESTING AND QUALITY ASSURANCE

**Max Marks:** 40

**Program & Sem:** B.TECH & VII OE

**Weightage:** 20 %

**Instructions:**

- i. All questions are compulsory
- ii. Write answers in brief
- iii. Draw figures wherever necessary

**Part A [Memory Recall Questions]**

**Answer both the Questions. Each question carries six marks. (2Qx6M=12M)**

1. Define the following: (C.O.NO. 1) [Knowledge]
  - a) Quality
  - b) Quality Assurance
  - c) Quality Control
2. Distinguish between Software Verification and Software Validation (C.O.NO.1) [Knowledge]

**Part B [Thought Provoking Questions]**

**Answer both the Questions. Each question carries eight marks. (2Qx8M=16M)**

3. With neat diagram, discuss Rapid Application Model (RAD) along with its strengths and weaknesses. (C.O.NO.1) [Comprehension]
4. List and describe the four quadrants involved in spiral model. (C.O.NO.1) [Comprehension]

**Part C [Problem Solving Questions]**

**Answer the Question. The Question carries twelve marks. (1Qx12M=12M)**

5. Explain in detail, phases involved in the development of Software Project by listing the activities in each phase (C.O.NO. 1) [Knowledge]





## SCHOOL OF ENGINEERING

Semester: VII

Course Code: CSE 403

Course Name: Software Testing and Quality Assurance

Date: 30/09/2019

Time: 1:00 to 2:00 PM

Max Marks: 40

Weightage: 20 %

### Extract of question distribution [outcome wise & level wise]

Q.NO	C.O.NO	Unit/Module Number/Unit /Module Title	Memory recall type [Marks allotted] Bloom's Levels			Thought provoking type [Marks allotted] Bloom's Levels			Problem Solving type [Marks allotted]			Total Marks
			K			C			A			
1	CO1	Module-1		6								6
2	CO1	Module-1		6								6
3	CO1	Module-1		8								8
4	CO1	Module-1		8								8
5	CO1	Module-1		12								12
	Total Marks	40		40								40

K = Knowledge Level    C = Comprehension Level, A = Application Level



Note: While setting all types of questions the general guideline is that about 60%

Of the questions must be such that even a below average students must be able to attempt, About 20% of the questions must be such that only above average students must be able to attempt and finally 20% of the questions must be such that only the bright students must be able to attempt.

[I hereby certify that All the questions are set as per the above guide lines. Mr. Ramakrishna ]

Reviewers' Comments



## SCHOOL OF ENGINEERING

### SOLUTION

Semester: VII

Course Code: CSE 403

Course Name: Software Testing and Quality Assurance

Date: 30/09/2019

Time: 1:00 to 2:00 PM

Max Marks: 40

Weightage: 20 %

#### Part A

(2Q x6 M =12 Marks)

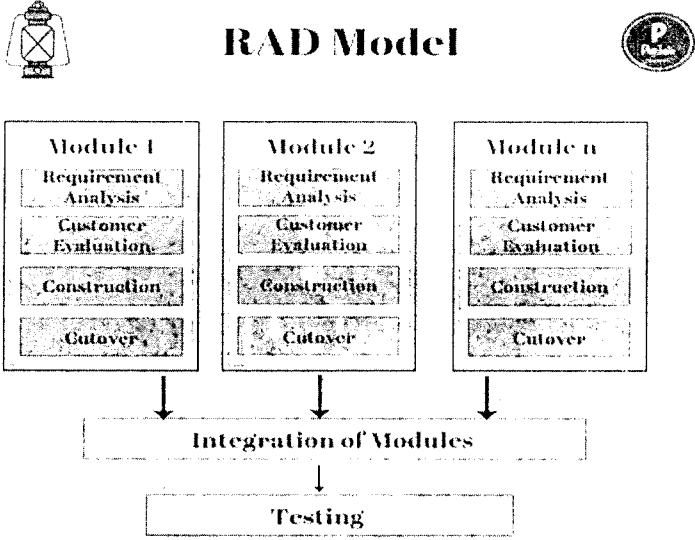
Q No	Solution	Scheme of Marking	Max. Time required for each Question
------	----------	-------------------	--------------------------------------



1	<p><b>Quality</b> is meeting the requirement, expectation, and needs of the customer is free from the defects, lacks and substantial variants. There are standards needs to follow to satisfy the customer requirements.</p> <p><b>Quality Assurance</b> is known as QA and focuses on preventing defect. Quality Assurance ensures that the approaches, techniques, methods and processes are designed for the projects are implemented correctly.</p> <p>Quality assurance activities monitor and verify that the processes used to manage and create the deliverables have been followed and are operative.</p> <p>Quality Assurance is a proactive process and is Prevention in nature. It recognizes flaws in the process. Quality Assurance has to complete before Quality Control.</p> <p><b>Quality Control</b> is known as QC and focuses on identifying a defect. QC ensures that the approaches, techniques, methods and processes are designed in the project are following correctly. QC activities monitor and verify that the project deliverables meet the defined quality standards.</p> <p>Quality Control is a reactive process and is detection in nature. It recognizes the defects. Quality Control has to complete after Quality assurance</p>	<p>1a) 1M 1b) &amp; 1c) 5M</p>	5 Mins												
2	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: left;">SOFTWARE VERIFICATION</th> <th style="width: 50%; text-align: left;">SOFTWARE VALIDATION</th> </tr> </thead> <tbody> <tr> <td>1. It means, "Are we building any System/Product in a right manner?"</td> <td>1. It means, "Are we building right System/Product?"</td> </tr> <tr> <td>2. Software Verification is the process of evaluating System/Products in the development phase to find whether they meet the specified requirements or not.</td> <td>2. Software Validation is the process of evaluating software at the end of development process in order to determine whether software Product/System meets the customer expectations and requirements or not.</td> </tr> <tr> <td>3. Software Verification process involves : a) Reviews. b) Meetings. c) Inspections.</td> <td>3. Software Validation involves : a) Black Box Testing. b) White Box Testing. c) Grey Box Testing.</td> </tr> <tr> <td>4. Verification of software is carried out by quality assurance team.</td> <td>4. Validation of software is carried out by the testing team.</td> </tr> <tr> <td>5. Execution of code is not done in case of Software Verification and is carried out before Validation process.</td> <td>5. Code is executed in case of Software Validation and it is carried out after the Software Verification process.</td> </tr> </tbody> </table>	SOFTWARE VERIFICATION	SOFTWARE VALIDATION	1. It means, "Are we building any System/Product in a right manner?"	1. It means, "Are we building right System/Product?"	2. Software Verification is the process of evaluating System/Products in the development phase to find whether they meet the specified requirements or not.	2. Software Validation is the process of evaluating software at the end of development process in order to determine whether software Product/System meets the customer expectations and requirements or not.	3. Software Verification process involves : a) Reviews. b) Meetings. c) Inspections.	3. Software Validation involves : a) Black Box Testing. b) White Box Testing. c) Grey Box Testing.	4. Verification of software is carried out by quality assurance team.	4. Validation of software is carried out by the testing team.	5. Execution of code is not done in case of Software Verification and is carried out before Validation process.	5. Code is executed in case of Software Validation and it is carried out after the Software Verification process.	6 M for 5 differences	5 Mins
SOFTWARE VERIFICATION	SOFTWARE VALIDATION														
1. It means, "Are we building any System/Product in a right manner?"	1. It means, "Are we building right System/Product?"														
2. Software Verification is the process of evaluating System/Products in the development phase to find whether they meet the specified requirements or not.	2. Software Validation is the process of evaluating software at the end of development process in order to determine whether software Product/System meets the customer expectations and requirements or not.														
3. Software Verification process involves : a) Reviews. b) Meetings. c) Inspections.	3. Software Validation involves : a) Black Box Testing. b) White Box Testing. c) Grey Box Testing.														
4. Verification of software is carried out by quality assurance team.	4. Validation of software is carried out by the testing team.														
5. Execution of code is not done in case of Software Verification and is carried out before Validation process.	5. Code is executed in case of Software Validation and it is carried out after the Software Verification process.														





Q No	Solution	Scheme of Marking	Max. Time required for each Question
3	<p>Requirements planning phase (a workshop utilizing structured discussion of business problems)</p> <p>User description phase – automated tools capture information from users</p> <p>Construction phase – productivity tools, such as code generators, screen generators. etc. inside a time-box. (“Do until done”)</p> <p>Cutover phase -- installation of the system, user acceptance testing and user training</p> <div style="text-align: center;">  <p><b>RAD Model</b></p> </div> <p><b>RAD Strengths:</b></p> <ul style="list-style-type: none"> <li>Reduced cycle time and improved productivity with fewer people means lower costs</li> <li>Time-box approach mitigates cost and schedule risk</li> <li>Customer involved throughout the complete cycle minimizes risk of not achieving customer satisfaction and business needs</li> <li>Focus moves from documentation to code (WYSIWYG).</li> <li>Uses modeling concepts to capture information about business, data, and processes.</li> </ul> <p><b>RAD Weaknesses:</b></p> <ul style="list-style-type: none"> <li>Accelerated development process must give quick responses to the user</li> <li>Risk of never achieving closure</li> <li>Hard to use with legacy systems</li> <li>Requires a system that can be modularized</li> <li>Developers and customers must be committed to rapid-fire activities in an abbreviated time frame.</li> </ul>	<p>RAD: 1M</p> <p>Diagram: 3M</p> <p>Strength:2M</p> <p>Weakness:2M</p>	10 Mins



4	<p>Four Quadrants in Spiral Model:  Determine  Evaluate  Plan  Develop</p> <p><b>Determine:</b>  Objectives: functionality, performance, hardware/software interface, critical success factors, etc.</p> <p><b>Evaluate:</b>  Alternatives: build, reuse, buy, sub-contract, etc.  Constraints: cost, schedule, interface, etc.</p> <p><b>Plan:</b>  Typical activities  Develop project plan  Develop configuration management plan  Develop a test plan  Develop an installation plan</p> <p><b>Develop:</b>  <b>Typical activities:</b>  Create a design  Review design  Develop code  Inspect code  Test product</p>	Each Quadrant 2M	10 Mins
---	--	------------------	---------

**Part C**

(1Q x12 M = 12 Marks)

Q No	Solution	Scheme of Marking	Max. Time required for each Question
5	<p><b>A software is developed by a series of phases as follows:</b></p> <ol style="list-style-type: none"> <li>1) <b>Requirements gathering and analysis</b></li> <li>2) <b>Planning</b></li> <li>3) <b>Design</b></li> <li>4) <b>Development/Coding/Implementation</b></li> <li>5) <b>Testing</b></li> <li>6) <b>Deployment and Maintenance</b></li> </ol> <p><b>Requirements gathering and analysis</b>  In this phase, the specific requirements of the software to be built are gathered and documented.  The requirements get documented in the form of a System Requirements Specification (SRS) documented.  SRS document acts as a bridge between the customer and software designers</p>	2M for Each phase	20 Mins



## Components of SRS Document

- Functional Requirements
- Non - Functional Requirements
- Static Requirements
- Execution Constraints eg. Response time, throughput time
- Standards to be followed
- Security requirements
- Company Policies
- Interface with other external agents ie. persons, software or hardware

### Planning:

The purpose of Planning phase is to come up with a schedule, scope and resource requirements for a release.

A plan explains requirements will be met and by what time.

It needs to take into consideration the requirements- what will be met and what will not be met.

### Activities of Planning Phase of SDLC

- Define business problem and scope
- Produce detailed project schedule
- Confirm project feasibility
  - Economic, organizational, technical, resource, and schedule
- Staff the project (resource management)
- Launch project → official announcement

### Design:

Design Phase's purpose is to figure out how to satisfy requirements enumerated in the SRS document.

It produces a representation (BluePrint) that will be used by the following phase, the development phase. This representation serves two purposes:

- a) possible to verify all requirements are satisfied
- b) Gives info to development phases to proceed with coding.

Design is split in High-Level Design (HDD) and Low-Level Design (SDD)

This phase produces System Design Description (SDD) that will be used by development teams to produce the programs that realize the design

## Design

- It is the most creative and challenging phase of SDLC.
- It defines the final system and refers to the technical specifications.
- DFDs are used to show the flow of system.
- Two phases: 1. Logical Design and Physical Design
- Logical Design: Specifies user needs.
- Physical Design: Tells the programmer what the candidate system must do.



**Development or Coding:**

Design act as a blueprint for actual coding to proceed.

This phase comprises coding the programs in chosen programming Language.

In addition, this phase also involves the creation of product documentation.

**Testing:**

As the programs are coded, they are also debugged and executed. After the coding is completed, the product is subjected to testing.

Testing is the process of exercising the software product in pre-defined ways to check if the behavior is the same as expected behavior.

Software testing is a process. to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product.

Software testing is a set of processes aimed at investigating, evaluating and ascertaining the completeness and quality of computer software. Software testing ensures the compliance of a software product in relation with regulatory, business, technical, functional and user requirements.

**Deployment and Maintenance:**

Once the software is tested, it is given to clients who deploy it in their environments.

As users starts using the product in their environment they may observe discrepancies between the actual behavior and what they were given to expect. Such defects need to be corrected.

The product now enters the maintenance phase, where the product is maintained or changed to satisfy customers expectations that arises from environmental changes.

Maintenance can be one among the following:

Corrective (Fixing customer related problems)

Adaptive (Making the S/w run on new version of OS or DB)

Preventive(Changing to code to avoid security threat)







Roll No.																				
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

**PRESIDENCY UNIVERSITY  
BENGALURU**

**SCHOOL OF ENGINEERING**

**TEST – 2**

**Sem & AY:** Odd Sem 2019-20

**Date:** 18.11.2019

**Course Code:** CSE 403

**Time:** 1.00 PM to 2.00 PM

**Course Name:** SOFTWARE TESTING AND QUALITY ASSURANCE

**Max Marks:** 40

**Program & Sem:** B.Tech & VII (OE)

**Weightage:** 20%

**Instructions:**

- (i) Read the question properly and answer accordingly.
- (ii) Question paper consists of 3 parts.

**Part A [Memory Recall Questions]**

**Answer all the Questions. Each Question carries four marks.**

**(3Qx4M=12M)**

1. Explain the methods in static testing by humans. **[CO2] [K]**
2. Discuss the two types of integration Testing. **[CO2] [K]**
3. Differentiate between equivalence class and boundary value testing.

**CO2] [K]**

**Part B [Thought Provoking Questions]**

**Answer both the Questions. Each Question carries eight marks.**

**(2Qx8M=16M)**

4.a. For a program with the condition statement:  $If(x < y) \text{ AND } (a > b) \text{ THEN}$   
Calculate the condition coverage percentage **(4 M) [CO2] [C]**

b. Design boundary value test cases for a Pizza delivery app.

Pizza values 1 to 10 is considered valid. A success message is shown.

While value 11 to 99 are considered invalid for order and an error message will appear, "Only 10 Pizza can be ordered". **(4 M) [CO2] [A]**

5. a. Calculate the cyclomatic complexity for the below piece of code.

(4 M) [CO1] [C]

```
if A=10 then
  if B > C then
    A=B
  else A=C
endif
endif
Print A
Print B
Print C
```

b. If you are testing for an input box accepting numbers from 1 to 1000. How will you design test case using equivalence partitioning.

(4 M) [CO2] [C]

### Part C [Problem Solving Questions]

Answer the Question. The Question carry twelve marks.

(1Qx12M=12M)

6. Design path testing cases for the function given below.

```
Function fn_delete_element (int value, int array_size, int array[])
{
  1 int i;
  location = array_size + 1;

  2 for i = 1 to array_size
  3 if ( array[i] == value )
  4 location = i;
  end if;
  end for;

  5 for i = location to array_size
  6 array[i] = array[i+1];
  end for;
  7 array_size --;
}
```

[CO2] [A]



## SCHOOL OF ENGINEERING

Semester: VII

Course Code: CSE403

Course Name: Software Testing & Quality Assurance

Date: 18 November 2019

Max Marks: 40

Weightage: 20%

### Extract of question distribution [outcome wise & level wise]

Q.NO	C.O.NO	Unit/Module Number/Unit /Module Title	Memory recall type			Thought provoking type			Problem Solving type			Total Marks
			[Marks allotted]	Bloom's Levels		[Marks allotted]	Bloom's Levels		[Marks allotted]			
			K			C			A			
1	CO2	M2/ White Box Testing	4									4
2	CO2	M3/ Integration testing	4									4
3	CO2	M2/ Black box testing	4									4
4.a	CO2	M2/ Black box testing				4						4
4.b.	CO2	M3/Integration testing						4				4
5.a	CO2	M2/ white box testing				4						4

5.b	CO2	M2/ white box testing				4					4
6	CO2	M2/static testing							12		12
<b>Total</b>			<b>12</b>			<b>12</b>			<b>16</b>		<b>40</b>

K =Knowledge Level C = Comprehension Level, A = Application Level

Note: While setting all types of questions the general guideline is that about 60%

Of the questions must be such that even a below average students must be able to attempt, About 20% of the questions must be such that only above average students must be able to attempt and finally 20% of the questions must be such that only the bright students must be able to attempt.



## SCHOOL OF ENGINEERING

### SOLUTION

Semester: VII

Course Code: CSE403

Course Name: Software Testing and Quality Assurance

Date: 18 November 2019

Max Marks: 40

Weightage: 20%

#### Part A

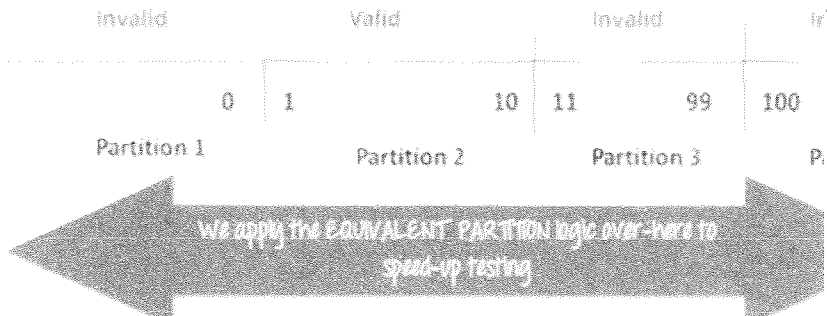
(3 Q x 4M = 12 Marks)

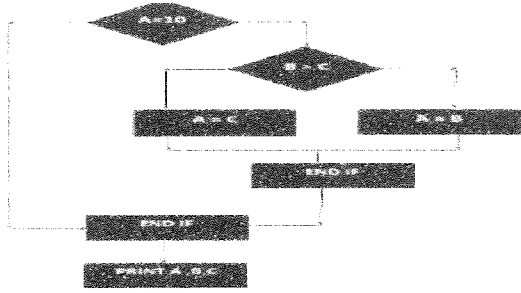
Q No	Solution	Scheme of Marking	Max. Time required for each Question
1	<ul style="list-style-type: none"> <li>• Desk checking of the code</li> <li>• Code walkthrough</li> <li>• Code Review</li> <li>• Code Inspection</li> </ul>	4 marks. 1 mark for each.	5 mins
2	<ul style="list-style-type: none"> <li>• <b>Internal Interfaces-</b> Provide communication across 2 modules within a project or product, internally not exposed to the customer or external developers.</li> <li>• <b>External/Exported Interfaces-</b> Visible outside the product to third party developers and solution providers.</li> </ul>	4 marks . 2 mark for each type.	5 mins

3	<ul style="list-style-type: none"> <li>• <b>Equivalence Class Testing:</b> It is used to minimize the number of possible test cases to an optimum level while maintains reasonable test coverage.</li> <li>• <b>Boundary Value Testing:</b> Boundary value testing is focused on the values at boundaries. This technique determines whether a certain range of values are acceptable by the system or not. It is very useful in reducing the number of test cases. It is most suitable for the systems where an input is within certain ranges.</li> </ul>	4 marks for differentiation	5 mins
---	---	-----------------------------	--------

**Part B**

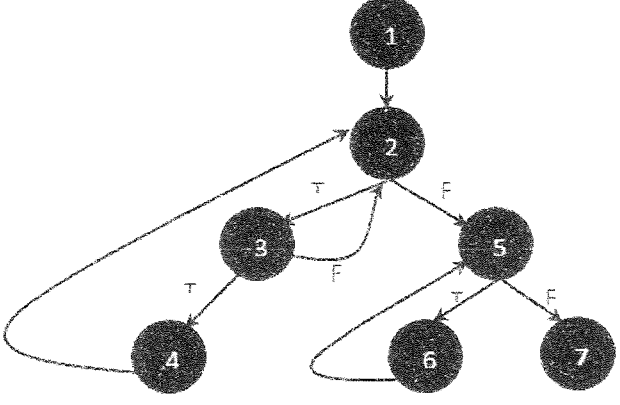
(2 Q x 8M = 16 Marks)

Q No	Solution	Scheme of Marking	Max. Time required for each Question
4.a	<p><b>4 possible combinations</b></p> <p>TT FF TF FT</p> <p>Eg. Consider x=3,Y=4 (x&lt;y)=TRUE (a&gt;b)=FALSE Conditioncoverage=</p> $\text{Condition Coverage} = \frac{\text{Number of Executed Operands}}{\text{Total Number of Operands}}$ <p>1/4 =25%</p>	4 marks	8 mins
4.b		4 marks	7 mins
5.a	<p>Cyclomatic complexity-E-N+2 E= No.of edges N= No.of Nodes,</p>	4 marks	8 mins

	<p>From graph <math>8-7+2=3</math></p>  <p>The characteristic computation is calculated using the above control flow diagram and shows below. Additionally, the right edge (loop) inside the equivalent graph is 8-7+2=3.</p>		
5.b	<ul style="list-style-type: none"> <li>#1) One input data class with all valid inputs. Pick a single value from range 1 to 1000 as a valid test case. If you select other values between 1 and 1000 the result is going to be the same. So one test case for valid input data should be sufficient.</li> <li>#2) Input data class with all values below the lower limit. I.e. any value below 1, as an invalid input data test case.</li> <li>#3) Input data with any value greater than 1000 to represent the third invalid input class.</li> <li>We have selected one representative from every input class to design our test cases. Test case values are selected in such a way that largest number of attributes of equivalence class can be exercised.</li> </ul>	4 marks	7min

Part C

(1Q x 12M = 12 Marks)

Q No	Solution	Scheme of Marking	Max. Time required for each Question
6	 <p>Path 1: 1 - 2 - 5 - 7</p> <p>Path 2: 1 - 2 - 5 - 6 - 7</p> <p>Path 3: 1 - 2 - 3 - 2 - 5 - 6 - 7</p>	12MARKS	15 min

	Path 4: 1 - 2 - 3 - 4 - 2 - 5 - 6 - 7		
--	---------------------------------------	--	--







Roll No																			
---------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

**PRESIDENCY UNIVERSITY  
BENGALURU**

**SCHOOL OF ENGINEERING**

**END TERM FINAL EXAMINATION**

**Semester:** Odd Semester: 2019 - 20

**Course Code:** CSE 403

**Course Name:** SOFTWARE TESTING AND QUALITY ASSURANCE

**Program & Sem:** B.Tech (CSE/ECE/EEE/MEC), & VII (OE-II)

**Date:** 26 December 2019

**Time:** 9:30 AM to 12:30 PM

**Max Marks:** 80

**Weightage:** 40%

**Instructions:**

- (i) Read the all questions carefully and answer accordingly.
- (ii) Draw diagrams where necessary

**Part A [Memory Recall Questions]**

**Answer all the Questions. Each Question carries 10 marks.**

**(2Qx10M=20M)**

1. a. Fill in the blanks with appropriate terms [5 Marks] (C.O.No.1-3) [Knowledge]
  - i. \_\_\_\_\_ is an informal manual test that programmers can use to verify coding and algorithm logic before a program launch
  - ii. \_\_\_\_\_ is a form of peer review in which a programmer leads the review process
  - iii. \_\_\_\_\_ is systematic examination (often as peer review) of computer source code
  - iv. \_\_\_\_\_ is the most formal type of review, which is a kind of static testing to avoid the defect multiplication at a later stage
  - v. \_\_\_\_\_ is a Level of Testing where smallest part of individual component
- b. List five coverage metric types with their formulae. [5 Marks]
2. a. List the phases of software development [5 Marks] (C.O.No.1-3) [Knowledge]
- b. List types of performance testing [5 Marks]

**Part B [Thought Provoking Questions]**

**Answer all the Questions. Each Question carries 10 marks.**

**(3Qx10M=30M)**

3. Explain types of regression testing with examples. (C.O.No.3) [Knowledge]
4. Explain Defect lifecycle scenario with an appropriate state diagram. (C.O.No.2) [Knowledge]
5. Explain ad-hoc testing. Describe types of Ad-hoc testing. (C.O.No.2) [Knowledge]

### Part C [Problem Solving Questions]

Answer all the Questions. Each Question carries 10 marks.

(3Qx10M=30M)

6. The triangle program accepts three integers, a, b and c as input. These are taken to be the sides of a triangle. The integers a, b and c must satisfy the following conditions

C1:  $1 \leq a \leq 200$

C2:  $1 \leq b \leq 200$

C3:  $1 \leq c \leq 200$

C4:  $a < b+c$

C5:  $b < a+c$

C6:  $c < a+b$

The output of the program may be either of: Equilateral Triangle, Isosceles Triangle,

Scalene or "Not a Triangle". Design BVA test cases (C.O.No.1) [Knowledge]

7. a. Differentiate between alpha testing and beta testing

b. Explain types of integration testing.

(C.O.No.1) [Knowledge]

8. a. Draw path graph and compute Cyclomatic complexity

```
insertion_procedure (int a[], int p [], int N)
{
  Int i,j,k;
  for (i=0; i<=N; i++)
  p[i] = i;
  for (i=2; i<=N; i++)
  {
    k=p[i];j=1;
    while (a[p[j-1]] > a[k])
    {
      p[j] = p[j-1];
      j--;
    }
    p[j] = k;
  }
}
```

- b. An exam has a pass boundary at 50 percent, merit at 75 percent and distinction at 85 percent. Design BVA test cases (C.O.No.1) [Knowledge]



## SCHOOL OF ENGINEERING

### END TERM FINAL EXAMINATION

#### Extract of question distribution [outcome wise & level wise]

Q.NO	C.O.NO (% age of CO)	Unit/Module Number/Unit /Module Title	Memory recall type	Thought provoking type	Problem Solving type [Marks allotted]	Total Marks
			[Marks allotted]	[Marks allotted]		
			Bloom's Levels	Bloom's Levels		
			K	C	A	
1	1	1 to 4	10			
2	2	1 to 4	10			
3	2	4		10		
4	2	4		10		
5	2	4		10		
6	2				10	
7	2				10	
8	2				10	
	Total Marks		20	30	30	80

K = Knowledge Level C = Comprehension Level, A = Application Level

Note: While setting all types of questions the general guideline is that about 60%

Of the questions must be such that even a below average students must be able to attempt, About 20% of the questions must be such that only above average students must be able to attempt and finally 20% of the questions must be such that only the bright students must be able to attempt.

I hereby certify that all the questions are set as per the above guidelines.

Faculty Signature:

Reviewer Comment:

## Format of Answer Scheme



## SCHOOL OF ENGINEERING

### SOLUTION

**Semester:** Odd Sem. 2019-20  
**Course Code:** CSE403  
**Course Name:** Software Testing & Quality Assurance  
**Program & Sem:** B.Tech. VII Sem

**Date:** xx.12.2019  
**Time:** 3 HRS  
**Max Marks:** 80  
**Weightage:** 40%

#### Part A

(2Q x 10M = 20Marks)

Q No	Solution	Scheme of Marking	Max. Time required for each Question
1 a	Desk checking Code Walkthrough Code review Code Inspection Unit testing	1 mark each	5 min
b	Statement Coverage Decision Coverage Branch Coverage Toggle Coverage FSM Coverage	1 mark each	5 min
2 A	Requirements gathering and analysis Planning Design Development/Coding/Implementation Testing Deployment and Maintenance	1 mark each	5 min
b	Load Testing Stress Testing Stability Testing Volume Testing Scalability Testing	1 mark each	5 min

Part B

(3Q x 10M = 30 Marks)

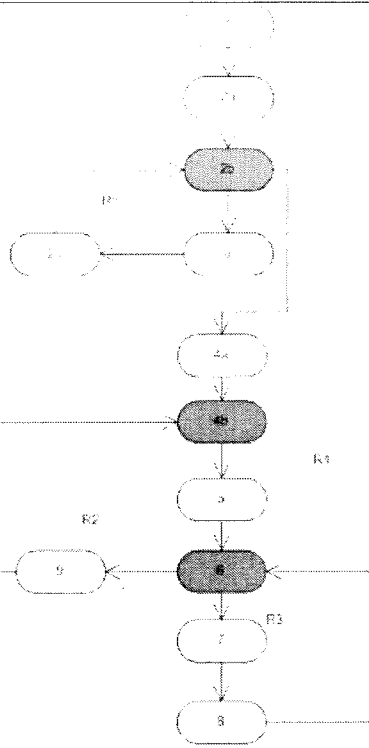
Q No	Solution	Scheme of Marking	Max. Time required for each Question
3	<p>Corrective</p> <p>Reset all</p> <p>Selective</p> <p>Progressive</p> <p>Complete</p> <p>Partial</p> <p>Unit</p>	<p>7 for listing and explanation</p> <p>3 marks for suitable examples</p>	20 min
4	<p><b>Types of Adhoc testing</b></p> <p>There are different types of Adhoc testing and they are listed as below:</p> <p><b>Buddy Testing</b> Two buddies mutually work on identifying defects in the same module. Mostly one buddy will be from development team and another person will be from testing team. Buddy testing helps the testers develop better test cases and development team can also make design changes early. This testing usually happens after ... .. completion.</p> <p><b>Pair testing</b> Two testers are assigned modules, share ideas and work on the same machines to find defects. One person can execute the tests and another person can take notes on the findings. Roles of the persons can be a tester and scribe during testing.</p> <p><i>Comparison Buddy and Pair Testing:</i> Buddy testing is combination of unit and ... .. together with developers and testers but Pair testing is done only with the testers with different knowledge levels. (Experienced and non-experienced to share their ideas and views)</p> <p><b>Random Testing</b> Randomly test the product or application without test cases with a goal to break the system.</p>	<p>Diagram 5 marks</p> <p>Explanation 5 Marks</p>	20 min
5	<p><b>Buddy Testing</b> Two buddies mutually work on identifying defects in the same module. Mostly one buddy will be from development team and another person will be from testing team. Buddy testing helps the testers develop better test cases and development team can also make design changes early. This testing usually happens after ... .. completion.</p> <p><b>Pair testing</b> Two testers are assigned modules, share ideas and work on the same machines to find defects. One person can execute the tests and another person can take notes on the findings. Roles of the persons can be a tester and scribe during testing.</p> <p><i>Comparison Buddy and Pair Testing:</i> Buddy testing is combination of unit and ... .. together with developers and testers but Pair testing is done only with the testers with different knowledge levels. (Experienced and non-experienced to share their ideas and views)</p> <p><b>Random Testing</b> Randomly test the product or application without test cases with a goal to break the system.</p>	<p>Diagram 5 marks</p> <p>Explanation 5 Marks</p>	20 min

Part C

(3Q x 10M = 30Marks)

Q No	Solution	Scheme of Marking	Max. Time required for each Question					
6	<table border="0"> <tr> <td>Test Case ID</td> <td>Side "a"</td> <td>Side "b"</td> <td>Side "c"</td> <td>Expected Output</td> </tr> </table>	Test Case ID	Side "a"	Side "b"	Side "c"	Expected Output	10 Marks	20 min
Test Case ID	Side "a"	Side "b"	Side "c"	Expected Output				

	<p>1            100        100        1            Isosceles Triangle</p> <p>2            100        100        2            Isosceles Triangle</p> <p>3            100        100        100        Equilateral Triangle</p> <p>4            100        100        199        Isosceles Triangle</p> <p>5            100        100        200        Not a Triangle</p> <p>6            100        1            100        Isosceles Triangle</p> <p>7            100        2            100        Isosceles Triangle</p> <p>8            100        100        100        Equilateral Triangle</p> <p>9            100        199        100        Isosceles Triangle</p> <p>10           100        200        100        Not a Triangle</p> <p>11           1            100        100        Isosceles Triangle</p> <p>12           2            100        100        Isosceles Triangle</p> <p>13           100        100        100        Equilateral Triangle</p> <p>14           199        100        100        Isosceles Triangle</p> <p>15           200        100        100        Not a Triangle</p>			
7 A	<p><b>Alpha Testing</b></p> <p>Alpha testing performed by Testers who are usually internal employees of the organization</p> <p>Alpha Testing performed at developer's site</p> <p>Reliability and Security Testing are not performed in-depth Alpha Testing</p> <p>Alpha testing involves both the white box and black box techniques</p> <p>Alpha testing requires a lab environment or testing environment</p> <p>Long execution cycle may be required for Alpha testing</p>	<p><b>Beta Testing</b></p> <p>Beta testing is performed by Clients or End Users who are not employees of the organization</p> <p>Beta testing is performed at a client location or end user of the product</p> <p>Reliability, Security, Robustness are checked during Beta Testing</p> <p>Beta Testing typically uses <u>Black Box Testing</u></p> <p>Beta testing doesn't require any lab environment or testing environment. The software is made available to the public and is said to be real time environment</p> <p>Only a few weeks of execution are required for Beta testing</p>	5 marks	20 min
b	<p>Top Down Integration</p> <p>Bottom up Integration</p> <p>Bi-directional Integration</p> <p>System Integration</p>		5 Marks	
8 A			10 marks	20 min



Count the number of regions on the graph: 4  
 No. of predicates (red on graph) + 1 : 3 + 1 = 4  
 No of edges – no. of nodes + 2: 14 – 12 + 2 = 4

- 49, 50 - for pass
- 74, 75 - for merit
- 84, 85 - for distinction
- 0 - for lower limit boundary value
- 101 - for upper limit boundary value

b

