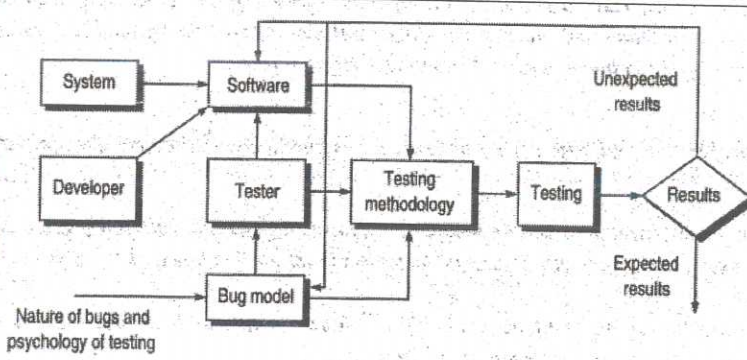


## Scoring Indicators

COURSE NAME: Software Testing

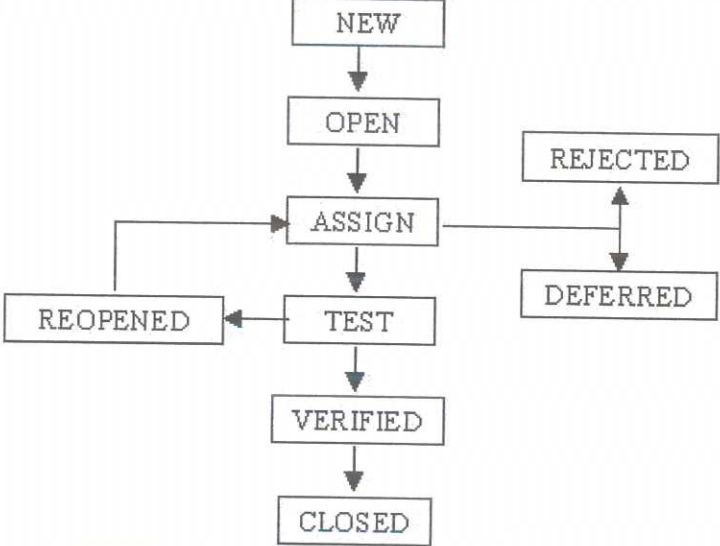
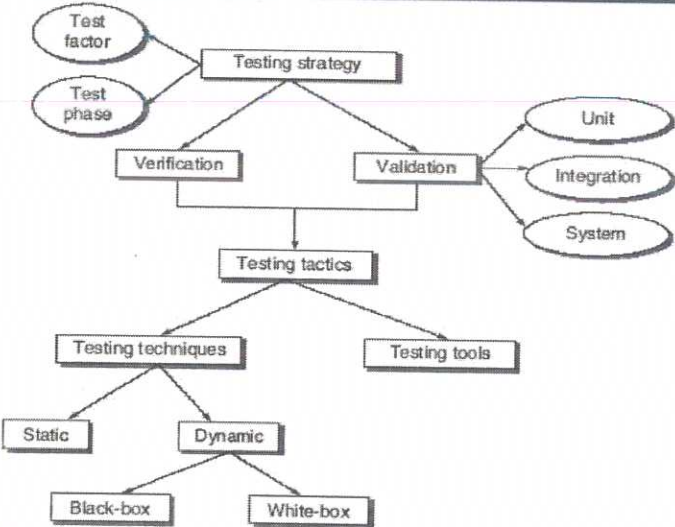
COURSECODE: 6131C (21)

QID: 2102240141

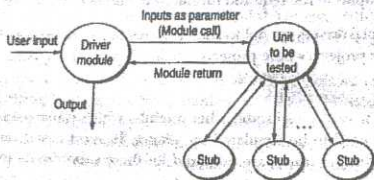
| Q No          | Scoring Indicators   | Split score                      | Sub Total | Total score |
|---------------|--|----------------------------------|-----------|-------------|
| <b>PART A</b> |  |                                  |           | <b>9</b>    |
| I. 1          | Reduced maintenance cost, improved testing process, bug prevention (any two)   | 1/2+1/2                          | 1         |             |
| I. 2          | Whenever a development team member makes a mistake in any phase of SDLC, errors are produced. It might be a typographical error, a misleading specification etc.   | 1                                | 1         |             |
| I. 3          | $6n+1 = 13$  | 1                                | 1         |             |
| I. 4          | In white box testing, structure means logic of the program which has been implemented in the language code.  | 1                                | 1         |             |
| I. 5          | Software inspection, walkthroughs, technical reviews   | 1/2+1/2                          | 1         |             |
| I. 6          | Organizes, moderates and follows up the walkthrough activities   | 1                                | 1         |             |
| I. 7          | The module under testing may also call some other module, which is not ready at the time of testing. Therefore, these modules need to be simulated for testing. In most cases, dummy modules instead of actual modules, which are not ready, are prepared for these subordinate modules. These dummy modules are called stubs.   | 1                                | 1         |             |
| I. 8          | Static and dynamic   | 1/2+1/2                          | 1         |             |
| I. 9          | Selenium, Jira   | 1/2+1/2                          | 1         |             |
| <b>PART B</b> |  |                                  |           | <b>24</b>   |
| II. 1         |  <p style="text-align: center;">Figure 1: Software Testing Models</p>  | Block diagram-2<br>Explanation-1 | 3         |             |
| II. 2         | <p>Failure: It means the inability of the system or a component of the system to perform a required function according to its specification.</p> <p>Error: Whenever a development team member makes a mistake in any phase of SDLC, errors are produced. It might be a typographical error, a misleading specification, a misunderstanding of what a subroutine does, etc.</p> | 1 1/2 x 2                        | 3         |             |

| II. 3         | <ul style="list-style-type: none"> <li>Select and rank test factors</li> <li>Identify system development phases</li> <li>Identify risks associated with the system under development</li> </ul>   | 1x3    | 3               |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
|---------------|---|--------|-----------------|---|-----------|---|---|------------|---|---|-----------|----|---|------------|---|---|---------------|---|---|--------------|---|---|-----------------|---|---|---|---|---|---|---|---|---|----|---|------|--------------|---|---|-----------------|---|---|---|-----|---|---|---|---|---|---|---|----|---|---|---|-----|---|---|---|------|---|---|---|-----|--|---|--|
| II. 4         | <p><b>Solution</b></p> <p>(a) Test cases using BVC Since there are two variables, <math>a</math> and <math>b</math>, the total number of test cases will be <math>4n + 1 = 9</math>. The set of boundary values is shown below:</p> <table border="1" data-bbox="531 465 802 656"> <thead> <tr> <th></th> <th>a</th> <th>b</th> </tr> </thead> <tbody> <tr> <td>Min value</td> <td>1</td> <td>1</td> </tr> <tr> <td>Min+ value</td> <td>2</td> <td>2</td> </tr> <tr> <td>Max value</td> <td>10</td> <td>5</td> </tr> <tr> <td>Max- value</td> <td>9</td> <td>4</td> </tr> <tr> <td>Nominal value</td> <td>5</td> <td>3</td> </tr> </tbody> </table> <p>Using these values, test cases can be designed as shown:</p> <table border="1" data-bbox="456 701 866 835"> <thead> <tr> <th>Test Case ID</th> <th>a</th> <th>b</th> <th>Expected Output</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>3</td> <td>1</td> </tr> <tr> <td>2</td> <td>2</td> <td>3</td> <td>8</td> </tr> <tr> <td>3</td> <td>10</td> <td>3</td> <td>1000</td> </tr> </tbody> </table><br><table border="1" data-bbox="376 976 1042 1335"> <thead> <tr> <th>Test Case ID</th> <th>a</th> <th>b</th> <th>Expected Output</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>9</td> <td>3</td> <td>729</td> </tr> <tr> <td>5</td> <td>5</td> <td>1</td> <td>5</td> </tr> <tr> <td>6</td> <td>5</td> <td>2</td> <td>25</td> </tr> <tr> <td>7</td> <td>5</td> <td>4</td> <td>625</td> </tr> <tr> <td>8</td> <td>5</td> <td>5</td> <td>3125</td> </tr> <tr> <td>9</td> <td>5</td> <td>3</td> <td>125</td> </tr> </tbody> </table> |        | a               | b | Min value | 1 | 1 | Min+ value | 2 | 2 | Max value | 10 | 5 | Max- value | 9 | 4 | Nominal value | 5 | 3 | Test Case ID | a | b | Expected Output | 1 | 1 | 3 | 1 | 2 | 2 | 3 | 8 | 3 | 10 | 3 | 1000 | Test Case ID | a | b | Expected Output | 4 | 9 | 3 | 729 | 5 | 5 | 1 | 5 | 6 | 5 | 2 | 25 | 7 | 5 | 4 | 625 | 8 | 5 | 5 | 3125 | 9 | 5 | 3 | 125 | Boundary values-1<br><br>Test cases -2 | 3 |  |
|               | a   | b      |                 |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| Min value     | 1   | 1      |                 |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| Min+ value    | 2   | 2      |                 |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| Max value     | 10  | 5      |                 |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| Max- value    | 9   | 4      |                 |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| Nominal value | 5   | 3      |                 |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| Test Case ID  | a   | b      | Expected Output |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| 1             | 1   | 3      | 1               |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| 2             | 2   | 3      | 8               |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| 3             | 10  | 3      | 1000            |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| Test Case ID  | a   | b      | Expected Output |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| 4             | 9   | 3      | 729             |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| 5             | 5   | 1      | 5               |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| 6             | 5   | 2      | 25              |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| 7             | 5   | 4      | 625             |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| 8             | 5   | 5      | 3125            |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| 9             | 5   | 3      | 125             |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| II. 5         | Planning<br>Overview<br>Individual preparation<br>Inspection meeting<br>Rework<br>Follow up   | 1x3    | 3               |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| II. 6         | Organization --> Preparation-> walk through-> rework and follow up  | 3      | 3               |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |
| II. 7         | <b>Alpha testing</b> is a type of acceptance testing, which is performed to identify all possible bugs/issues before releasing the product to the end-user. Alpha test is a preliminary software field test carried out by a team of users to find out the bugs that were not found previously by other tests. Alpha testing is to simulate a real user environment by carrying out tasks and operations that actual user might perform. Alpha testing implies a meeting with a software vendor and client to ensure that   | 1 ½ x2 | 3               |   |           |   |   |            |   |   |           |    |   |            |   |   |               |   |   |              |   |   |                 |   |   |   |   |   |   |   |   |   |    |   |      |              |   |   |                 |   |   |   |     |   |   |   |   |   |   |   |    |   |   |   |     |   |   |   |      |   |   |   |     |  |   |  |

|       |   |     |   |  |
|-------|---|-----|---|--|
|       | <p>the developers appropriately meet the client's requirements in terms of the performance, functionality, and durability of the software.</p> <p>Alpha testing needs lab environment, and usually, the testers are an internal employee of the organization. This testing is called alpha because it is done early on, near the end of the software development, but before beta testing.</p> <p><b>Beta Testing</b> is a type of acceptance testing; it is the final test before shipping a product to the customers. Beta testing of a product is implemented by "real users" of the software application in a "real environment." In this phase of testing, the software is released to a limited number of end-users of the product to obtain feedback on the product quality. It allows the real customers an opportunity to provide inputs into the design, functionality, and usability of the product. These inputs are essential for the success of the product. Beta testing reduces product failure risks and increases the quality of the product through customer validation. Direct feedback from customers is a significant advantage of beta testing. This testing helps to tests the software in a real environment. The experiences of the previous users are forwarded back to the developers who make final changes before releasing the software product.</p> |     |   |  |
| II. 8 | <ul style="list-style-type: none"> <li>• Consider building a tool instead of buying one, if possible</li> <li>• Test the tool on application prototype</li> <li>• Not all the test should be automated</li> <li>• Select the tools according to organizational needs</li> </ul> <p>Use proven test script development techniques</p>  | 1x3 | 3 |  |
| II.9  | <ul style="list-style-type: none"> <li>• Reduction of testing effort</li> <li>• Reduces the testers involvement in executing tests</li> <li>• Facilitates regression testing</li> <li>• Avoids human mistakes</li> <li>• Reduces overall cost of software</li> <li>• Simulated testing</li> <li>• Internal testing</li> <li>• Test enablers</li> <li>• Test case design</li> </ul>  | 1x3 | 3 |  |

| II.10         | Basic unit for testing<br>Implications of inheritance<br>Polymorphism<br>White-box testing<br>Black-box testing<br>Integration strategies   | 1x3                         | 3  |                 |                                  |                 |                                  |   |    |    |    |              |          |   |   |    |    |               |    |   |   |   |   |   |   |   |    |    |    |               |    |                                |   |   |
|---------------|---|-----------------------------|----|-----------------|----------------------------------|-----------------|----------------------------------|---|----|----|----|--------------|----------|---|---|----|----|---------------|----|---|---|---|---|---|---|---|----|----|----|---------------|----|--------------------------------|---|---|
| <b>PART C</b> |   |                             |    |                 |                                  |                 |                                  |   |    |    |    |              |          |   |   |    |    |               |    |   |   |   |   |   |   |   |    |    |    |               |    |                                |   |   |
| III. 1        |  <pre> graph TD     NEW[NEW] --&gt; OPEN[OPEN]     OPEN --&gt; ASSIGN[ASSIGN]     ASSIGN --&gt; REJECTED[REJECTED]     REJECTED &lt;--&gt; DEFERRED[DEFERRED]     DEFERRED --&gt; ASSIGN     ASSIGN --&gt; TEST[TEST]     TEST --&gt; REOPENED[REOPENED]     REOPENED --&gt; ASSIGN     TEST --&gt; VERIFIED[VERIFIED]     VERIFIED --&gt; CLOSED[CLOSED]       </pre>   | Diagram-4<br>Explanation -3 | 7  | 7               |                                  |                 |                                  |   |    |    |    |              |          |   |   |    |    |               |    |   |   |   |   |   |   |   |    |    |    |               |    |                                |   |   |
| III. 2        | <div style="background-color: #333; color: white; padding: 5px; text-align: center; font-weight: bold;">Software Testing Methodology</div>  <pre> graph TD     TS[Testing strategy] --- TF((Test factor))     TS --- TP((Test phase))     TS --- V[Verification]     TS --- Val[Validation]     V --- TT[Testing tactics]     Val --- U((Unit))     Val --- I((Integration))     Val --- S((System))     TT --- Tn[Testing techniques]     TT --- Tt[Testing tools]     Tn --- St[Static]     Tn --- Dn[Dynamic]     Dn --- Bb[Black-box]     Dn --- Wb[White-box]       </pre> | Diagram-4<br>Explanation -3 | 7  | 7               |                                  |                 |                                  |   |    |    |    |              |          |   |   |    |    |               |    |   |   |   |   |   |   |   |    |    |    |               |    |                                |   |   |
| III. 3        | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Test case ID</th> <th>x</th> <th>y</th> <th>z</th> <th>Expected result</th> <th>Classes covered by the test case</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>13</td> <td>25</td> <td>36</td> <td>Z is largest</td> <td>I1,I2,I3</td> </tr> <tr> <td>2</td> <td>0</td> <td>13</td> <td>45</td> <td>Invalid input</td> <td>I4</td> </tr> <tr> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>7</td> <td>33</td> <td>21</td> <td>51</td> <td>Invalid input</td> <td>I9</td> </tr> </tbody> </table>  | Test case ID                | x  | y               | z                                | Expected result | Classes covered by the test case | 1 | 13 | 25 | 36 | Z is largest | I1,I2,I3 | 2 | 0 | 13 | 45 | Invalid input | I4 | - | - | - | - | - | - | 7 | 33 | 21 | 51 | Invalid input | I9 | Test classes-3<br>Test cases-4 | 7 | 7 |
| Test case ID  | x   | y                           | z  | Expected result | Classes covered by the test case |                 |                                  |   |    |    |    |              |          |   |   |    |    |               |    |   |   |   |   |   |   |   |    |    |    |               |    |                                |   |   |
| 1             | 13  | 25                          | 36 | Z is largest    | I1,I2,I3                         |                 |                                  |   |    |    |    |              |          |   |   |    |    |               |    |   |   |   |   |   |   |   |    |    |    |               |    |                                |   |   |
| 2             | 0   | 13                          | 45 | Invalid input   | I4                               |                 |                                  |   |    |    |    |              |          |   |   |    |    |               |    |   |   |   |   |   |   |   |    |    |    |               |    |                                |   |   |
| -             | -   | -                           | -  | -               | -                                |                 |                                  |   |    |    |    |              |          |   |   |    |    |               |    |   |   |   |   |   |   |   |    |    |    |               |    |                                |   |   |
| 7             | 33  | 21                          | 51 | Invalid input   | I9                               |                 |                                  |   |    |    |    |              |          |   |   |    |    |               |    |   |   |   |   |   |   |   |    |    |    |               |    |                                |   |   |

| III. 4         | <p>The decision table for the program is shown below:</p> <table border="1" data-bbox="368 309 1062 528"> <thead> <tr> <th colspan="5">ENTRY</th> </tr> <tr> <th></th> <th></th> <th>Rule 1</th> <th>Rule 2</th> <th>Rule 3</th> </tr> </thead> <tbody> <tr> <td>Condition Stub</td> <td>C1: Working hours &gt; 8</td> <td>T</td> <td>F</td> <td>F</td> </tr> <tr> <td></td> <td>C2: Holidays or Sundays</td> <td>F</td> <td>T</td> <td>F</td> </tr> <tr> <td>Action Stub</td> <td>A1: Normal salary</td> <td></td> <td>X</td> <td></td> </tr> <tr> <td></td> <td>A2: 1.25 of salary</td> <td></td> <td></td> <td>X</td> </tr> <tr> <td></td> <td>A3: 2.00 of salary</td> <td>X</td> <td></td> <td></td> </tr> </tbody> </table> <p>The test cases derived from the decision table are given below:</p> <table border="1" data-bbox="363 566 1062 723"> <thead> <tr> <th>Test Case ID</th> <th>Working Hour</th> <th>Day</th> <th>Expected Result</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>8</td> <td>Monday</td> <td>Normal salary</td> </tr> <tr> <td>2</td> <td>10</td> <td>Tuesday</td> <td>1.25 of salary</td> </tr> <tr> <td>3</td> <td>12</td> <td>Sunday</td> <td>2.00 of salary</td> </tr> </tbody> </table>  | ENTRY   |                 |        |  |  |  |  | Rule 1 | Rule 2 | Rule 3 | Condition Stub | C1: Working hours > 8 | T | F | F |  | C2: Holidays or Sundays | F | T | F | Action Stub | A1: Normal salary |  | X |  |  | A2: 1.25 of salary |  |  | X |  | A3: 2.00 of salary | X |  |  | Test Case ID | Working Hour | Day | Expected Result | 1 | 8 | Monday | Normal salary | 2 | 10 | Tuesday | 1.25 of salary | 3 | 12 | Sunday | 2.00 of salary | Entry -3<br>Test cases -4 | 7 | 7 |
|----------------|---|---------|-----------------|--------|--|--|--|--|--------|--------|--------|----------------|-----------------------|---|---|---|--|-------------------------|---|---|---|-------------|-------------------|--|---|--|--|--------------------|--|--|---|--|--------------------|---|--|--|--------------|--------------|-----|-----------------|---|---|--------|---------------|---|----|---------|----------------|---|----|--------|----------------|---------------------------|---|---|
| ENTRY          |   |         |                 |        |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
|                |   | Rule 1  | Rule 2          | Rule 3 |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
| Condition Stub | C1: Working hours > 8   | T       | F               | F      |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
|                | C2: Holidays or Sundays   | F       | T               | F      |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
| Action Stub    | A1: Normal salary   |         | X               |        |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
|                | A2: 1.25 of salary  |         |                 | X      |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
|                | A3: 2.00 of salary  | X       |                 |        |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
| Test Case ID   | Working Hour  | Day     | Expected Result |        |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
| 1              | 8   | Monday  | Normal salary   |        |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
| 2              | 10  | Tuesday | 1.25 of salary  |        |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
| 3              | 12  | Sunday  | 2.00 of salary  |        |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
| III. 5         | <p>Completeness- without executing all the test cases , we strive to touch the completeness of testing domain.</p> <p>Nonredundancy- reuse the redundant test cases</p>   | 3 ½ x 2 | 7               | 7      |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
| III. 6         | <p><b>1 NEED OF WHITE-BOX TESTING</b></p> <p>white-box testing really necessary? Can't we write the code and simply test the software using black-box testing techniques? The supporting reasons for white box testing are given below:</p> <ol style="list-style-type: none"> <li>In fact, white-box testing techniques are used for testing the module for initial stage testing. Black-box testing is the second stage for testing the software. Though test cases for black box testing can be designed earlier than for white-box testing, they cannot be executed until the code is produced and checked using white-box testing techniques. Thus, white-box testing is not an alternative but an essential stage.</li> <li>Since white-box testing is complementary to black-box testing, there are categories of bugs that can be revealed by white-box testing, but not through black-box testing. There may be portions in the code, which are not checked when executing functional test cases, but these will be executed and tested by white-box testing.</li> <li>Errors that come from the design phase will also be reflected in the code and therefore, we must execute white-box test cases for code verification (unit verification).</li> <li>We often believe that a logical path is not likely to be executed but, in fact, it may be executed on a regular basis. White-box testing explores these paths too.</li> <li>Some typographical errors are not observed and go undetected and are not covered by black box testing techniques. White-box testing techniques help detect these errors.</li> </ol> | 7       | 7               | 7      |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
| III. 7         | <ul style="list-style-type: none"> <li>It minimizes the resources required to regression test a new version</li> <li>It is achieved by minimizing the number of test cases applied to the new version</li> <li>It is needed because a regression test suite grows with each version, resulting in broken ,obsolete, uncontrollable and redundant test cases</li> <li>It analyses the relationship between the test cases and the software elements they cover</li> <li>It uses the information about changes to select test cases.</li> </ul>   | 7       | 7               | 7      |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |
| III. 8         | <ul style="list-style-type: none"> <li>Checks if bug has been addressed</li> <li>Finds other related bugs</li> <li>Check effect on other parts of program</li> </ul>  | 7       | 7               | 7      |  |  |  |  |        |        |        |                |                       |   |   |   |  |                         |   |   |   |             |                   |  |   |  |  |                    |  |  |   |  |                    |   |  |  |              |              |     |                 |   |   |        |               |   |    |         |                |   |    |        |                |                           |   |   |

|                |  |   |   |   |
|----------------|--|---|---|---|
| <p>III. 9</p>  | <p><b>Benefits of Designing Stubs and Drivers</b></p> <p>The benefit of designing drivers and stubs (see Fig. 7.5) is that a unit will work/ behave in the stub environment as in the actual software environment. Now a unit test case can be written against the interface and the unit will still work properly once the drivers and stubs have been placed correctly.</p> <p>The benefits of designing stubs and drivers are:</p> <ul style="list-style-type: none"> <li>▪ Stubs allow the programmer to call a method in the code being developed, even if the method does not have the desired behaviour yet.</li> <li>▪ By using stubs and drivers effectively, we can cut down our total debugging and testing time by testing small parts of a program individually, helping us to narrow down problems before they expand.</li> <li>▪ Stubs and drivers can also be an effective tool for demonstrating progress in a business environment. For example, if you are to implement four specific methods in a class by the end of the week, you can insert stubs for any method and write a short driver program so that you can demonstrate to your manager or client that the requirement has been met. As you continue, you can replace the stubs and drivers with the real code to finish your program.</li> </ul>  <p>The diagram illustrates the interaction between a Driver module and a Unit to be tested. The Driver module receives 'User input' and sends 'Inputs as parameter (Module call)' to the Unit to be tested. The Unit to be tested returns 'Module return' to the Driver module, which then produces 'Output'. The Unit to be tested is supported by multiple 'Stub' components.</p> <p>Figure 7.5 Drivers and Stubs</p> | 7 | 7 | 7 |
| <p>III. 10</p> | <ul style="list-style-type: none"> <li>• Integration testing exposes inconsistency between the modules such as improper call or return sequences.</li> <li>• Data can be lost across an interface.</li> <li>• One module when combined with another module may not give the desired result.</li> <li>• Data types and their valid ranges may mismatch between the modules</li> </ul>   | 7 | 7 | 7 |

|         |   |   |   |   |
|---------|---|---|---|---|
| III. 11 | <p><b>16.4 CHALLENGES IN TESTING FOR WEB-BASED SOFTWARE</b></p> <p>As discussed earlier, web-based software are different as compared to traditional systems. Therefore, we must understand the environment of their creation and then test them. Keeping in view the environment and behaviour of web-based systems, we face many challenges while testing them. These challenges become issues and guidelines when we perform testing of these systems. Some of the challenges and quality issues for web-based system are discussed here:</p> <p><b>Diversity and complexity</b> Web applications interact with many components that run on diverse hardware and software platforms. They are written in diverse languages and they are based on different programming approaches such as procedural, OO, and hybrid languages such as Java Server Pages (JSPs). The client side includes browsers, HTML, embedded scripting languages, and applets. The server side includes CGI, JSPs, Java Servlets, and .NET technologies. They all interact with diverse back-end engines and other components that are found on the web server or other servers.</p> <p><b>Dynamic environment</b> The key aspect of web-applications is its dynamic nature. The dynamic aspects are caused by uncertainty in the program behaviour, changes in application requirements, rapidly evolving web technology itself, and other factors. The dynamic nature of web software creates challenges for the analysis, testing, and maintenance for these systems. For example, it is difficult to determine statically the application's control flow because the control flow is highly dependent on user input and sometimes in terms of trends in user behaviour over time or user location. Not knowing which page an application is likely to display hinders statically modelling the control flow with accuracy and efficiency.</p> <p><b>Very short development time.</b> Clients of web-based systems impose very short development time, compared to other software or information systems projects (e.g., an e-business system, sports website, etc).</p> <p><b>Continuous evolution.</b> Demand for more functionality and capacity after the system has been designed and deployed to meet the extended scope and demands, that is, scalability issues.</p> <p><b>Compatibility and interoperability</b> As discussed, there may also be compatibility issues that make web testing a difficult task. Web applications often are affected by factors that may cause incompatibility and interoperability issues. The problem of incompatibility may exist on both the client as well as the server side. The server components can be distributed to different operating systems. Various versions of browsers running under a variety of operating systems can be there on the client side. Graphics and other objects on a website have to be tested on multiple browsers. If more than one browser will be supported, then the graphics have to be visually checked for differences in the physical appearance. The code that executes from the browser also has to be tested. There are different versions of HTML. They are similar in some ways but they have different tags which may produce different features.</p> | 7 | 7 | 7 |
| III. 12 | <p>Selenium is an open-source testing tool that helps automate tests on web browsers. It has many features, including:</p> <ul style="list-style-type: none"> <li>• Multi-browser support: Selenium can test web applications across different browsers and platforms.</li> <li>• Multi-language compatibility: Selenium can be used in multiple languages.</li> <li>• Dynamic web elements: Selenium can handle dynamic web elements.</li> <li>• Performance and speed: Selenium can be fast and performant.</li> <li>• Portability: Selenium can work with different operating systems.</li> <li>• Reusability and extras: Selenium can be reused and include extras.</li> <li>• Relative locators: Selenium 4 introduced relative locators,</li> </ul>   | 7 | 7 | 7 |

|  |   |  |  |  |
|--|---|--|--|--|
|  | also known as friendly locators, to help locate elements based on their relationship to other elements. |  |  |  |
|--|---|--|--|--|