Lab Manual

Software Engineering (Pr)

COT-216 & IT-313
Software Engineering
COT-206

1. **Introduction**
   Introduction to Software crisis & Software processes; Software life cycle models – Build & Fix, waterfall prototype evolutionary, spiral model.

2. **Requirement Analysis & Specifications**
   Problem Analysis – DFD, Data dictionaries, ER diagrams, object diagrams; approaches to problems analysis; SRS; specifying behavioral & non-behavioral requirements.

3. **Software Design**
   What is design? Modularity, strategy of design, function oriented design, object oriented design.

4. **Software Metrics**
   Introduction, size metrics, data structure metrics, information flow metrics, entropy-based measures, metric analysis.

5. **Software Reliability**
   Importance, Software reliability & Hardware reliability, failures & faults, reliability concepts, reliability models – macro, basic, logarithmic Poisson, calendar time component, micro models; estimating number of residual errors; reliability allocation.

6. **Software Testing**
   Introduction, Functional testing, structural testing, activities during testing, debugging, testing tools.

7. **Software Maintenance**
   Introduction, types of maintenance, maintenance process, maintenance models, reverse engineering, re-engineering.

**BOOKS**
B Tech 5th Semester (Information Technology)

IT-303  Software Engineering  

L  T  P
3  1

1. Introduction

Introduction to Software crisis & Software processes; Software life cycle models – Build & Fix, waterfall, incremental, prototype evolutionary, spiral model, Unified process. [1]

2. Requirement Analysis & Specifications

Requirements engineering, types of requirements, feasibility study, requirement elicitation, analysis, documentation, validation, management, case study [1]

3. Software Project Planning

Size estimation, cost estimation, COCOMO, COCOMO II, Putnam model, risk management [1]

4. Software Design

Design, modularity, strategy of design, function oriented design, object oriented design. [1]

5. Software Metrics

Introduction, token count, data structure metrics, IF metrics, O-O metrics, size metrics, data structure metrics, information flow metrics, entropy-based measures, metric analysis. [1]

6. Software Reliability

Basic concepts, software quality, reliability models, Capability Maturity Models, ISO 9000 [1]

7. Software Testing

Introduction, functional testing, structural testing, levels of testing, debugging, testing tools. [1]

8. Software Maintenance

Introduction, types of maintenance, maintenance models, reverse engineering, re-engineering. [1]

BOOKS:
Lab Instructions

Whether an experiment contains one or several practicals/programs

Several practicals/programs

• Lab Teacher forms groups of the students based on
  Total number of students in a group = \( \frac{\text{no. of students}}{\text{no. of practicals}} \)
  • Assign all practicals/programs among all groups

One practical/program

All Students need to perform the practical/program

• Teacher decides whether the completed practicals/programs can be appropriately described using flow chart, algorithm, query statement, etc.
  • Teacher issues necessary instructions to the students for writing practicals/programs accordingly

• Students write experiments in practical files and get them signed by the lab teacher
  • Students make entries in the list of contents of the practical files and get them signed by the lab teacher

Whether practical has been verified and signed by the lab teacher?

• If a student has not completed a practical, he/she is expected to complete it at his/her own with the help of his/her fellow students in his/her hostel
  • The student completes the practical file and submits it to the concerned teacher in his/her office or mail box on next working day

• In case of an experiment containing several practicals, a lab teacher needs to think whether a practical performed by the students in one group needs to be repeated by the other groups in lab on the same day?
  
  OR

  A practical performed by the students in one group needs to be repeated as assignments to be completed by the students of other groups in their hostels? Here, an assignment includes both executing a program on computer and also writing the same in practical file.
  
  OR

  A practical performed by the students in one group needs to be repeated as assignments, only writing practicals in their practical files, for the students of other groups in their hostels?

• Teacher issues necessary instructions to the students accordingly.
PART-I

Experiment 1 (Role of Software)

Objective  To identify the role of the software in today’s world across a few significant domains related to day to day life.

Background  Software has made the world a global village today. The impact of software spans across almost all aspect of human life. All organizations, Institutions and companies are leveraging the potentials of software in automating the critical functions and eliminating manual interventions. Software is also a predominant area for trade and export especially for the countries like India. Domains like health care, Airlines , financial Services, Insurance , retails, Education, and many more have exploited software and still there a lot of the scope for software to create impact and add values in multiple dimensions.

Problem Description  In the context of this background, indentify the areas (or application or systems) how software has been leveraged extensively in the following domains,

1. Health Care
2. Airlines
3. Banking Insurance
4. Retail
5. Education

Estimated time: 60 minutes

Experiment 2 (Software Crisis)

Objective  To identify the problem related to software crisis for a given scenario

Background  In the early years of computers applications, the focus of the development and innovation were on hardware. Software was largely views as an afterthought. Computer Programming was an art. Programmers did not follow any disciplined or formalized approaches. This way of doing things was adequate for a while, until the sophisticated of computer applications outgrow. Software soon took over and more functions which were done manually. A software houses begin to develop for widespread distribution. Software development projects produced thousands of source
program statement. With the increase in the size and complexity of the software, following situation resulted is collectively termed as software crisis.

1. Time Slippage
2. Cost Slippage
3. Failure at customer Site
4. Intractable Error after delivery

### Problem Description
In the context of this background, for each of the scenario mentioned below, identify the most appropriate problem related to software crisis and mention the same in the table provided.

#### Scenario A
Railways reservation software was delivered to the customer and was installed in one of the metro station at 12.00 AM (mid-night) as per the plan. The system worked quite fine till the next day 12.00 PM(noon). The system crashed at 12.00 PM and the railways authorities could not continue using software for reservation till 02.00 M. It took two hours to fix the defect in the software in the software.

#### Scenario B
A polar satellite launch vehicle was scheduled for the launch on August 15th. The auto-pilot of the rocket to be delivered for integration of the rocket on may 15th. The design and development of the software for the auto-pilot more effort because of which the auto-pilot was delivered for the integration on June 15th (delayed by a month). The rocket was launched on Sep 15th (delayed by a month).

#### Scenario C
Software for financial systems was delivered to the customer. Customer informed the development team about a mal-function in the system. As the software was huge and complex, the development team could not identify the defect in the software.

#### Scenario D
Due to the defect in the software for the baggage handling system. There was also of & 2M of revenues for the airport authorities.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Situation (as given A to D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>

**Estimated time:** 60 minutes

### Experiment 3 (Software Development Model)

**Objective**
To identify the suitable software development model for the given scenario.

**Background**
“ABC” is a new enterprise recently started by a few young entrepreneurs, intended to develop robotic systems using IT based solutions. “TENZ” is a well-
established automobile manufacturing plant, interested in automating some of the operations of the plant.

As first step towards automation, TENZ has decided to deploy a robotic arm to pick and place large number of bolts from one tray of a conveyer belt to another tray of a different conveyer belt. Moving one bolt at a time. TENZ has asked ABC for demonstration the robotic arm for the said purpose as a proof of concept. This is the first project being executed by ABC. TENZ would like to have a first experience of the robotic arm before signing the MOU (Memorandum of Understanding) with ABC. After MOU sign-off, TENZ would provide the complete set of requirement of the robot.

**Problem Description**

Identify the most appropriate SDLC model for this scenario.

**Estimated time:** 30 minutes

**Experiment 4 (Requirement Development)**

**Objective**

To identify the various requirement development activities viz. elicitation, analysis, specification and verification for the given scenarios.

**Background**

Requirement engineering produces a specification of what a system should do. The intention of requirement engineering is to provide a clear definition of requirement of the systems. This phase is a very important phase because, if the customer requirements are not clearly understood, the ambiguity can get into the other phase of the development. To avoid such issues, requirement has to be elicited using the right elicitation techniques, to be analyzed effectively, specified clearly and verified thoroughly.

All activities are collectively termed as requirement development activities.

**Problem Description**

Identify the requirement development activities associated with each of the following scenarios,

a. Joe is creating an online survey questionnaire for requesting user feedback on the desired features of the application to be developed.
b. Mark is preparing a formal document which includes all of the desired features identified by the survey.
c. Jack identified an incomplete requirement statement.
d. Jones is identifying all security related requirement and separating them from the performance related requirements.
e. Merlin a team member is sent to client to observe the business case and collect typical user requirements.
f. Leo is team member is working on requirement and ensuring that requirement collected should not be vague and unclear.
g. Lee is conducting a facilitated meeting with the stakeholder to capture the
h. Amit a team member is distributing questionnaires to stack holder for gathering user requirements.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Requirement Development Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td></td>
</tr>
<tr>
<td>g</td>
<td></td>
</tr>
<tr>
<td>h</td>
<td></td>
</tr>
</tbody>
</table>

Estimated time: 60 minutes

Experiment 5 (Requirement Elicitation)

Objective
To identify the various elicitation techniques and their usage for the Banking case study.

Background
Requirement elicitation is the process of seeking, discovering, acquiring and elaborating requirement. This includes learning and understanding the needs of the users. This activity is communication centric and iterative in the nature. The techniques used here are the important to get stack holder consensus on the requirements.

Problem Description
KHL is a leading global bank that provides standard banking services to its customers spanning across the globe. The head office is located in London and the bank has presence in more than 20 countries with client base of nearly 500,000. Tuning with times and ever increasing clients and transactions, the bank has specialized branches for specific customer segments like consumer, corporate and the SME’s. KHL Bank aims to be one stop shop for its customers to address their changing financial needs. KHL bank offers various banking products and services across its customer segments including Core Banking and Wealth Management amongst other services.

KHL Bank is well known among its clients for world-class processes and speed of execution of transactions as part of core banking. Currently, KHL bank has made a proposal for investing around $200 million in setting-up 24x7 banking support facilities for the customers. The bank has decided to leverage IT for automating several of the business processes including:

- Managing Accounts
- Transaction Management
The aim of this proposed banking system is to create a paperless bank thereby moving towards e-banking. FinSoft, a newly established software company has the vision of providing software solutions in the financial sector. Managing Director (MD) of KHL bank has approached FinSoft for the computerization of the bank so that there is no more manual way of doing transactions in any of its branches. As part of automation, the KHL bank users are to be provided with ATM facility, e-banking facility over internet and phone banking facility over land lines and cellular networks. FinSoft is doing such a project for the first time. Requirements development team in FinSoft has planned for carrying out the requirement elicitation for this project.

In the context of the case study, for the following scenarios identify the most appropriate requirements elicitation techniques (Brainstorming, Workshops, Questionnaire, Task Analysis, Observation, Prototyping, Scenario identification).

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Requirement elicitation technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interrogative conservation with Managers, Cashiers, Clerks and other Staff for arriving at the requirement for automating transactions.</td>
<td></td>
</tr>
<tr>
<td>Formal and planned requirement discussion in a conference to room conducted among managers of diversified branches facilitate by anchor.</td>
<td></td>
</tr>
<tr>
<td>Survey form circulated among the users (account holders) who visit the bank, to ease their interactions with bank</td>
<td></td>
</tr>
<tr>
<td>Analysis for understanding mode of transactions- Checks, Cash, DD, MT, Gold, etc.</td>
<td></td>
</tr>
<tr>
<td>Ethnographers deployed for understanding the users interactions with bank officials.</td>
<td></td>
</tr>
<tr>
<td>UI design of e-banking portal, ATM, Computer Systems</td>
<td></td>
</tr>
<tr>
<td>Understanding the process involved in each transaction like withdraw, deposit, fund transfer etc.</td>
<td></td>
</tr>
</tbody>
</table>

**Estimated time:** 60 minutes

**Experiment 6 (Requirement Classification)**

**Objective**

To Classify the requirement into functional and non-functional requirements.

**Background**

Functional requirements (FRs) specify the software functionality that the developer must build into the product to enable users accomplish their tasks, thereby satisfying the business requirements. Non functional requirement as the name suggest, are those requirements which are not directly concerned with the
specific functions delivered by the system. Many non-functional requirements (NFRs) related to the system as a whole rather than to individual functional requirements. While failure to meet an individual functional may degrade the system, failure to meet a non-functional system requirement may make whole system unusable. NFR’s are of different types namely usability requirements, reliability requirements etc.

**Problem Description**

Classify the following requirement by selecting the appropriate option.

1. ATM machine shall validate PIN of the user during login along with biometric verification.
2. “Peak transaction Volume(s)- 20,000 calls in a busy hour, average duration 20 Secs, grade of services 99.98%”
3. “Brahe System Shall sounds the alarm for 10 seconds at frequency of 100H when the brake is applied”.
4. “Mean Time to failure (MTTF) -There should be no more than three Severity-1 outage per month”.

**Estimated time:** 30 minutes

**Experiment 7 (Requirements)**

**Objective**

Identify the elements in software Requirements Specification document.

**Background**

The output of requirement analysis place is the Software requirements Specification it should specify what a system should do. Atypical SRS usually consists of the system overview (both current and proposed). Objectives of the proposed system such that it is proven to be significant development over the existing system, business analysis of the client functional and non-functional requirement, glossary, etc.

**Problem Description**

Article Systems (AS), a giant books franchise has approached IT company ABCLtd. To help them automate their business processes and shift them online so all of its outlets across the globe would be under one portal. AMC Ltd, agrees to take on the project and both the companies sit down and strike a deal. After detailed discussion and analysis, ABC Ltd. decides to follows the waterfall model for development of the project. This discussion covered topics such as the business analysis and operations of AS, The expectations of the client (AS) from ABC Ltd, their needs and requirements from the software, their current existing system including its limitations and what the proposed system would do to address these issues. Taking into account all the input, the concerned team at ABC designed and developed the SRS.

Which of the following are typically expected to be found in the SRS created by ABC Ltd. Towards the AS Project?
System Overview
Class Diagram
source Code
Use Cases
Flow Charts for algorithms realizing the functionality of the system
Code review Comments
test results
The Difference in the scope of the current system to be proposed system.
Non-Functional requirements
Features of new portal to be developed

Experiment 8 (Requirement Verification)

Objective
To verify the requirements against the quality attributes.

Background
Software requirements specification formally captures the requirements of the software to be developed. Hence it is important that requirements are free from defects like incorrect or conflicting requirements.

Problem Description
Identify the requirements in the given SRS(Premium University Placement Portal) for following issues,
1. Incorrect requirements
2. Ambiguous requirements
3. Missing requirements
4. Conflicting requirements
5. Incomplete requirements

Estimated time: 60 minutes

Experiment 9 (Software Design)

Objective
Identify the elements and relationship by analyzing the class diagram of Easy Shop Retail Application case study.

Background
Software Design describes how software is decomposed and organized into components and the interfaces between those components. It also describes the components at a level of detail that enable their construction.

Problem Description
1. Study the class diagram of Easy shop retail application depicted in the “CCFP4.0-SE Easy Shop Manager-Phase1-ClassDiagram.docx” and identify
the Following for the class “Customer”.
1. Class Name
2. Private attributes
3. Public attributes
4. Private methods
5. Public Methods

2. In above document, identify the relationship between the elements given below:

<table>
<thead>
<tr>
<th>Class 1</th>
<th>Class 2</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Apparels</td>
<td></td>
</tr>
<tr>
<td>Retail Outlet</td>
<td>Item</td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>Regular Customer</td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>Address</td>
<td></td>
</tr>
<tr>
<td>PurchaseBill</td>
<td>LineItem</td>
<td></td>
</tr>
<tr>
<td>Tax</td>
<td>PurchaseBill</td>
<td></td>
</tr>
</tbody>
</table>

Estimated time: 60 minutes

Experiment 10 (Software Design Principles)

Objective
Identify the design principle that is being violated in relation to the given scenario.

Background
A good object oriented design not only meets the specified requirements but also addresses implicit requirements. There are five design principles which address most of the implicit requirements:

Software Design Principles
1. Abstraction: Focus on solving a problem by considering the relevant details and ignoring the irrelevant.
2. Encapsulation: Wrapping the internal details, thereby making these details inaccessible. Encapsulation separates interface and implementation, specifying only the public interface to the clients, hiding the details of implementation.
3. Decomposition and Modularization: Dividing the problem into smaller, independent, interactive subtasks for placing different functionalities in different components.
4. Coupling & Cohesion: Coupling is the degree to which modules are dependent on each other. Cohesion is the degree to which a module has a single, well defined task or responsibility. A good design is one with loose coupling and strong cohesion.
5. Sufficiency, Completeness and Primitiveness: Design should ensure the completeness and sufficiency with respect to the given specifications in a very simple way as possible.
Problem Description
Which of the following design principle(s) have been violated in the following scenarios?
1. Abstraction
2. Decomposition and Modularization
3. Coupling & Cohesion
4. Encapsulation
5. Sufficiency, Completeness and Primitiveness
6. All

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
<th>Principle Being Violated</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Important information of a module is directly accessible by other modules</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Too many global variables in the program after implementing the design</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Code breaks in unexpected places</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Unfulfilled requirements in the code after the design has been implemented</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Cyclic dependency among classes</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Huge class doing too many unrelated operations</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Several un-related functionalities/tasks are carried out by a single module</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>All data of all classes in public</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Design resulting in spaghetti code</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>An algorithm documented as part of design is not understandable by the programmers</td>
<td></td>
</tr>
</tbody>
</table>

Estimated time: 60 minutes

Experiment 11 (Integration Testing)

Objective
To identify the usage of stubs or drivers in the context of an integration testing scenario.

Background
Integration testing is carried out after the completion of unit testing and before the software is delivered for system testing. In top down integration testing, dummy stubs are required for bottom level modules. Similarly in bottom up testing, dummy drivers are required for top level modules.

Problem Description
Consider the scenario of development of software for Travel Management System (TMS) is in progress. The TMS software has 3 major modules namely Ticket_Booking_Module, Hotel_Booking_Module and Taxi_Booking_Module. The Ticket_Booking_Module has 3 sub modules namely Enquiry_Module, Booking_Module and Update_Module. The enquiry
module uses Date_Validation_Unit, Ticket_Validation_Unit and Place_Validation_Unit.

In the context of the given scenario, identify the usage of stub or driver for the following situations.

1. Except the Ticket_validation_Unit, the coding and unit testing of all other modules, sub modules and units of TMS are completed. The top-down integration is in progress for the TMS software. To carry out the integration testing, which among the following is necessary?

   - A Stub for Ticket_Validation_Unit
   - A Driver For Ticket_Validation_Unit
   - A Stub for Enquiry_Module
   - A Driver for Enquiry_Module
   - A Stub For Ticket_Booking_Module
   - A Driver For Ticket_Booking_Module

2. The coding and unit testing of all the module, sub modules and units of TMS are completed except the Update_Module (coding and testing for Edit_Module, Cancel_Module and View_Module are also completed). The bottom-up integration is to be started for the TMS software. Mention any stub or driver needed to carry out the integration testing?

3. Except the Taxi_Booking_Module, the coding and unit testing of all other modules, sub modules and units of TMS are completed. The top-down integration is to be started for the TMS software. Mention any stub or driver needed to carry out the integration testing?

Estimated time: 60 minutes
Experiment 12 (Performance Testing)

Objective
Identify the different types of performance testing

Background
Performance testing tests the non-functional requirements of the system. The different types of performance testing are load testing, stress testing, endurance testing and spike testing.

Problem Description
Identify the type of performance testing for the following:

1. A spacecraft is expected to function for nearly 8 years in space. The orbit control system of the spacecraft is a real-time embedded system. Before the launch, the embedded software is to be tested to ensure that it is capable of working for 8 years in the space. Identify the suitable performance testing category to be carried out to ensure that the spacecraft will be functioning for 8 years in the space as required.

2. Global Education Centre (GEC) at Infosys Mysore provides the training for fresh entrants. GEC uses an automated tool for conducting objective type test for the trainees. At a time, a maximum of 2000 trainees are expected to take the test. Before the tool is deployed, testing of the tool was carried out to ensure that it is capable of supporting 2000 simultaneous users. Indicate the performance testing category?

4. A university uses its web-based portal for publishing the results of the students. When the results of an examination were announced on the website recently on a pre-planned date, the website crashed. Which type of performance testing should have been done during website development to avoid this unpleasant situation?

4. During unexpected terrorist attack, one of the popular websites crashed as many people logged into the website in a short span of time to know the consequences of terrorist attack and for immediate guidelines from the security personnel. After analyzing the situation, the maintenance team of that website came to know that it was the consequences of unexpected load on the system which had never happened previously. Which type of performance testing should have been done during website development to avoid this unpleasant situation?

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Performance Testing Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td></td>
</tr>
<tr>
<td>Scenario 2</td>
<td></td>
</tr>
<tr>
<td>Scenario 3</td>
<td></td>
</tr>
<tr>
<td>Scenario 4</td>
<td></td>
</tr>
</tbody>
</table>

Estimated time: 60 minutes
Experiment 13 (Regression Testing)

Objective
To identify the usage of regression testing.

Background
Enhancements are introduction of new features to the software and might be released in different versions. Whenever a version is released, regression testing should be done on the system to ensure that the existing features have not been disturbed.

Problem Description
Consider the scenario of development of software for Travel Management System (TMS) discussed in previous assignment. TMS has been developed by Infosys and released to its customer Advance Travel Solutions Ltd. (ATSL). Integration testing, system testing and acceptance testing were carried out before releasing the final build to the customer. However, as per the customer feedback during the first month of usage of the software, some minor changes are required in the Enquiry Module of the TMS. The customer has approached Infosys with the minor changes for upgrading the software. The development team of Infosys has incorporated those changes, and delivered the software to testing team to test the upgraded software.

Which among the following statement is true?
1. Since minor changes are there, integration of the Enquiry Module and quick system testing on Enquiry module should be done.
2. The incorporation of minor changes would have introduced new bugs into other modules, so regression testing should be carried out.
3. Since the acceptance testing is already carried out, it is enough if the team performs sanity testing on the Enquire module.
4. No need of testing any module.

Estimated time: 30 minutes

Experiment 14 (Defect Classification)

Objective
To classify the given defects into different defect types.

Background
Defect detection activities like reviews and testing help in identifying the defects in the artifacts (deliverables). These defects must be classified into various buckets before carrying out the root cause analysis. Following are some the defect Categories.
1. Logical
2. User interface
3. Maintainability
4. Standards

Problem Description
In the context of the above defect categories, classify the following statements under the defect categories and mention in the table given below.
1. Divide by Zero Error is not guarded
2. Usage of 3.14 in the statement Circle_Area = 3.14 * Radius * Radius;
3. 3500 lines of code in a single function
4. A pointer is declared but not initialized. It is used in the program for storing a value.
5. A program designed to handle 1000 simultaneous users, crashed when 1001 the user logged in.
6. A “while” loop never exits
7. User interface displays “MALFUNCTION 54” when something goes wrong in the back-end
8. No documentation (comments) for the source code
9. Hungarian Notation not followed while coding, even though the coding guidelines mandate to use Hungarian Notation
10. Pressing of “Tab” key moves the cursor in different fields of a web form randomly.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Defect Category</th>
<th>Defect Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Estimated time: **60 minutes**

**Experiment 15 (Software Metrics)**

**Objective**
To understand usage of software metrics.

**Background**
There are some metrics which are fundamental and the rest can be derived from these. Examples of basic (fundamental) measures are Size, Effort, Defect, Schedule. If the fundamental measures are known, then we can derive others. For example if size and effort are known, we can get Productivity (=size/effort). If the total numbers of defects are known we can get the Quality (=defect/size) and so on.

**Problem Description**
Online loan system has two modules for the two basic services, namely Car loan service and House loan service.
The two modules have been named as Car_Loan_Module and House_Loan_Module. Car_Loan_Module has 2000 lines of uncommented source code. House_Loan_Module has 3000 lines of uncommented source code. Car_Loan_Module was completely implemented by Mike. House_Loan_Module was completely implemented by John. Mike took 100 person hours to implement Car_Loan_Module. John took 200 person hours to implement House_Loan_Module. Mike’s module had 5 defects. John’s module had 6 defects. With respect to the context given, which among the following is an INCORRECT statement?

Choose one:
1. John’s Quality is better than Mike’s Quality
2. John’s Productivity is more than Mike’s Productivity
3. John introduced more defects than Mike
4. John’s Effort is more than Mike’s Effort

Estimated time: 60 minutes

PART-II

Project Work

Objective
Understand Importance of SDLC approach & various processes.

Problem Description
Software Development Life cycle involve multiple phases like Requirement analysis, Design, Coding, Testing & Maintenance. In each phase certain activities are performed to deliver the deliverables or outcome eg. from Requirement Analysis phase outcome is SRS document which is input to the Design phase.

Project group is assigned with a project and asked to perform the complete cycle of SDLC. Following are some instruction related to the project work,

1. Project group has to collect the required information related to the assigned system/project.
2. Project group has to develop the project and deliver the deliverables on the timelines given by the Teached/Lab Instructor.
3. On the completion of duration, each project group has to submit a project report. The Teacher/Lab Instructor will evaluate the report.
4. The Implementation of project will be done on C language.

Estimated time: 15-30 hours