SOFTWARE ENGINEERING LABORATORY

LAB MANUAL

Academic Year	**:**	**2018 - 2019**
Course Code	**:**	**ACS107**
Regulations	**:**	**IARE - R16**
Semester	**:**	**IV**
Branch	**:**	**IT**

Prepared by

Mr. A. PRAVEEN
Assistant Professor

Mrs.B.DHANALAXMI
Assistant Professor

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad - 500 043

IARE
Institute of Aeronautical Engineering

1. PROGRAM OUTCOMES:

B.TECH - PROGRAM OUTCOMES (POs)	
PO-1	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems (**Engineering knowledge**).
PO-2	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences (**Problem analysis**).
PO-3	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations (**Design/development of solutions**).
PO-4	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions (**Conduct investigations of complex problems**).
PO-5	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations (**Modern tool usage**).
PO-6	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice (**The engineer and society**).
PO-7	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development (**Environment and sustainability**).
PO-8	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice (**Ethics**).
PO-9	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings (**Individual and team work**).
PO-10	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions (**Communication**).
PO-11	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments (**Project management and finance**).
PO-12	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change (**Life-long learning**).

2. PROGRAM SPECIFIC OUTCOMES

PROGRAM SPECIFIC OUTCOMES (PSO's)	
PSO-1	**Professional Skills:** The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexity.
PSO-2	**Software Engineering Practices:** The ability to apply standard practices and strategies in software service management using open-ended programming environments with agility to deliver a quality product for business success.
PSO-3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

3. ATTAINMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

S No	Experiment	Program Outcomes Attained	Program Specific Outcomes Attained
WEEK-1	**ROLE OF SOFTWARE** **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 are 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, identify 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	PO-2, PO-3	PSO-1

WEEK-2	**SOFTWARE CRISIS**
	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 PM. 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.
		PO-2, PO-3	PSO-1

	Scenario D: Due to the defect in the software for the baggage handling system. There was also a loss of $2M of revenues for the airport authorities. 	Scenario	Situation (as given A to D)
---	---
A	
B	
C	
D				

WEEK-3	**REQUIREMENT DEVELOPMENT** **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 requirements. h. Amit a team member is distributing questionnaires to stack holder for gathering user requirements.	**PO-2**	

WEEK-4	**REQUIREMENT CLASSIFICATION AND VERIFICATION** A. **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. Nonfunctional 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 di 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 bio-metric verification. 2. "Peak transaction-20,000calls inVolume(s) a busy hour, average duration 20 Secs, grade of services 99.98%. 3. "Brahe System sounds the alarmShallfor10seconds at frequency of 100H when the brake is applied". 4. "Mean Time Failure (MTTF) to -There should be no more than three Severity-1 outage per month".	**PO-3**	**PSO-1**

	B. **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		
WEEK-5	**SOFTWARE DESIGN PRINCIPLES** **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 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. 2. Decomposition and Modularization: Dividing the problem into smaller, independent, interactive subtasks for placing different functionalities in different components 3. 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. 4. 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	**PO-3**	**PSO-1**
WEEK-6	**INTEGRATION TESTING** **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 Travel_Management_System Ticket_Booking_Module Hotel_Booking_Module Taxi_Booking_Module Enquiry_Module Booking_Module Update_Module View_Module Edit_Module Cancle_Module Data_Validation_Unit Place_Validation_Unit Ticket_Validation_Unit Source_Validation_Unit Destination_Validation_Unit	**PO-3**	**PSO-1**

	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? 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.		
WEEK-7	**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 space craft 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 space craft 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? 3. 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 web site crashed. Which type of performance testing should have been done during web-site development to avoid this unpleasant situation? 4. During unexpected terrorist attack, one of the popular websites crashed as many people logged into the web-site 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 web-site development to avoid this unpleasant situation?	**PO-2, PO-3**	

WEEK-8	**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? a. Since minor changes are there, integration of the Enquiry Module and quick system testing on Enquiry module should be done. b. The incorporation of minor changes would have introduced new bugs into other modules, so regression testing should be carried out. c. Since the acceptance testing is already carried out, it is enough if the team performs sanity testing on the Enquire module. d. No need of testing any module.	**PO-2,PO-3**	
WEEK-9	**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, and 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 Carloan 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. Jonh'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. 2. John's productivity is more than Mike. 3. John introduced more defects than Mike. 4. John's effort is more than Mike.	**PO-3**	

4. MAPPING COURSE OBJECTIVES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Objectives	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
I		✓	✓									✓	✓		
II		✓	✓									✓	✓		
III					✓									✓	
1V		✓							✓					✓	

5. SYLLABUS:

SOFTWARE ENGINEERING LABORATORY

IV Semester: IT							
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**	

Course Code	Category	L	T	P	C	CIA	SEE	Total
ACS107	Core	-	-	3	2	30	70	100

Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 27	Total Classes: 27

OBJECTIVES:
The course should enable the students to:
I.   Select suitable software development process model for the given scenario.
II. Classify the requirements and prepare software requirement documents for analyzing the projects.
III. Understand the different design techniques and their implementation.
IV. Apply various testing methodologies for validating design models.

WEEK-l	ROLE OF SOFTWARE

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 are   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, identify 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 |

| **WEEK-2** | **SOFTWARE CRISIS** |

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.

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 PM. 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 loss of $2M of revenues for the airport authorities.

Scenario	Situation (as given A to D)
A	
B	
C	
D	

| **WEEK-3** | **REQUIREMENT DEVELOPMENT** |

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 requirements.

    h. Amit a team member is distributing questionnaires to stack holder for gathering user requirements.

Scenario	Requirement Development Activities
A	
B	
C	
D	
E	
F	
G	
H	

WEEK-4	REQUIREMENT CLASSIFICATION AND VERIFICATION

C. **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. Nonfunctional 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 di 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 bio-metric verification.

    2. "Peak transaction-20,000calls in Volume(s)abusyhour, average duration 20 Secs, grade of services 99.98%.

    3. "Brahe System sounds the alarmShallfor10seconds at frequency of 100H when the brake is applied".

    4. "Mean Time Failure (MTTF) to -There should be no more than three Severity-1 out age per month".

D. **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

WEEK-5	SOFTWARE DESIGN PRINCIPLES

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
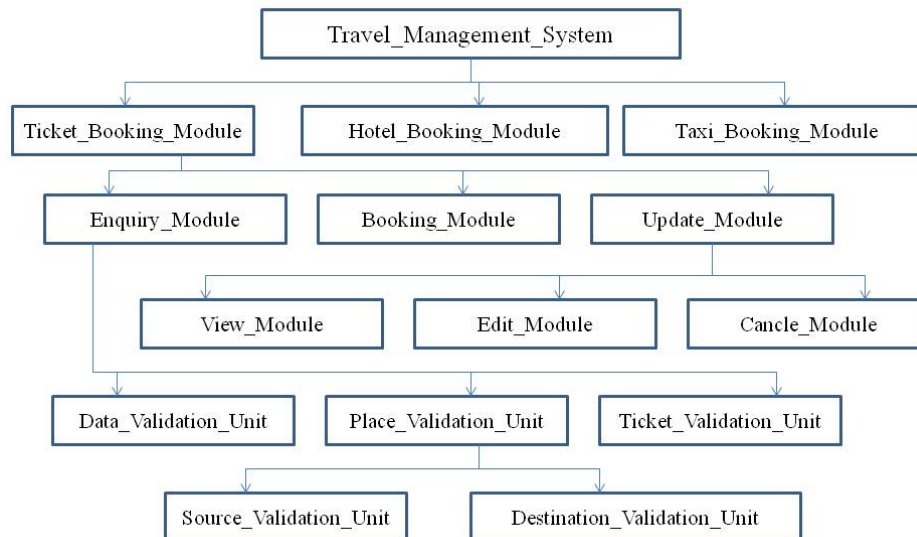5. Sufficiency, Completeness and Primitiveness
6. All

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

WEEK-6	INTEGRATION TESTING

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.

WEEK-7	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 space craft 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 space craft 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?
3. 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 web site crashed. Which type of performance testing should have been done during web-site development to avoid this unpleasant situation?

4. During unexpected terrorist attack, one of the popular websites crashed as many people logged into the web-site 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 web-site development to avoid this unpleasant situation?

Scenarios	Performance Testing Type
Scenario 1	
Scenario 2	
Scenario 3	
Scenario 4	

WEEK-8	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.

WEEK-9	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, and 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. Jonh'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
2. John's productivity is more than Mike

3. John introduced more defects than Mike
4. John's effort is more than Mike

REFERENCE BOOKS:

1. Roger S. Pressman, "Software Engineering: A Practitioner's Approach", Tata McGraw Hill International Edition, 7th Edition, 2009.
2. Ian Somerville, "Software Engineering", Pearson Education, 8th Edition, 2008.

WEB REFERENCES:

1. www.tutorialspoint.com
2. www.webopedia.com
3. http://vlabs.iitkgp.ernet.in/se/

SOFTWARE AND HARDWARE REQUIREMENTS FOR A BATCH OF 36 STUDENTS:

HARDWARE:
Desktop Computer Systems: 36 Nos

SOFTWARE:
Application Software: C Programming compiler /Java Development Kit

6. INDEX:

SOFTWARE ENGINEERING LABORATORY

WEEK-1	ROLE OF SOFTWARE

OBJECTIVE: To identify the role of software in today's world across a few significant domains.

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 are 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, identify 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

Summary: Identified the role of software across multiple domains related to day to day life.

Viva Questions:
1. Define software.
2. Give different example software's
3. Give the Importance of software in human life
4. List any Future expected software's example
5. What do you mean by changing nature of software

WEEK-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.

Scenario	Situation (as given A to D)
A	Failure at Customer Site
B	Time Slippage
C	Intractable Error after Delivery
D	Cost Slippage

Summary: mapped the problems related to software crisis to the given scenarios.

Viva Questions:
1. What do you mean by software crisis
2. Give an example of an error in any software
3. List any software failure feature
4. What do you mean by a project
5. Differentiate between a process and a project

WEEK-3	REQUIREMENT DEVELOPMENT

OBJECTIVE: To identify the various requirements 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 requirements.
   h. Amit a team member is distributing questionnaires to stack holder for gathering user requirements.

Scenario	Requirement Development Activities
A	Survey or Questionnaire
B	Process Modeling
C	Observation
D	Process Modeling
E	Process Modeling
F	Interface Analysis
G	Interface Analysis
H	Observation

Summary: identified the requirement development activities for the given scenarios

Viva Questions:
1. Define a requirement
2. Give an example for a requirement
3. List the classifications of the requirements
4. Define non functional requirements
5. List the categories of functional requirements

WEEK-4	REQUIREMENT CLASSIFICATION AND VERIFICATION

OBJECTIVE: a) To classify the requirements into functional and non-functional requirements
　　　　　b) To verify the requirements against quality attributes

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. Nonfunctional 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 di 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 bio-metric verification.
2. "Peak transaction-20,000calls inVolume(s)abusyhour,average duration 20 Secs, grade of services 99.98%.
3. "Brahe Systemsoundsthe alarmShallfor10seconds at frequency of 100H when the brake is applied".
4. "Mean Time Failure (MTTF)to -There should be no more than three Severity-1 outage per month".

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
   Brahe - misspelled
   "Brahe Systemsoundsthe alarmShallfor10seconds at frequency of 100H when the brake is applied".

2. Ambiguous requirements
   ATM (No need of Biometric only one login required)
   ATM machine shall validate PIN of the user during login along with bio-metric verification.

3. Missing requirements
   ATM machine shall validate PIN of the user during login along with bio-metric verification.

4. Conflicting requirements
   "Mean Time Failure (MTTF)to -There should be no more than three Severity-1 outage per month".

5. Incomplete requirements
   Peak , MTTF
   "Peak transaction-20,000calls inVolume(s) a busyhour, average duration 20 Secs, grade of services 99.98%.
   "Mean Time Failure (MTTF) to -There should be no more than three Severity-1 outage per month".

   Summary: a) classified the requirements into functional and non-functional requirements.
   　　　　b) Identified the requirement defects in the given SRS
   Viva Questions:
   1. Who are stakeholders
   2. List few stakeholders involved in any example project
   3. What do you mean by failure
   4. Define a defect of software

5. What are conflicting requirements

WEEK-5	**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

S No.	Description	Principle Being Violated
1	Important information of a module is directly accessible by other modules.	Encapsulation
2	Too many global variables in the program after implementing design	Coupling & Cohesion, Decomposition and Modularization
3	Code breaks in unexpected places	All
4	Unfulfilled requirements in the code after the design has been implemented	All
5	Cyclic dependencies among classes	Coupling & Cohesion, Decomposition and Modularization
6	Huge class doing too many unrelated operations	Abstraction, Decomposition and Modularization
7	Several unrelated functionalities/tasks are carried out by a single module	Decomposition and Modularization, Abstraction
8	All data of all classes in public	Encapsulation
9	Design resulting in spaghetti code	Sufficiency, Completeness and Primitiveness
10	An algorithm documented as part of design is not understandable by the programmers	Sufficiency, Completeness and Primitiveness

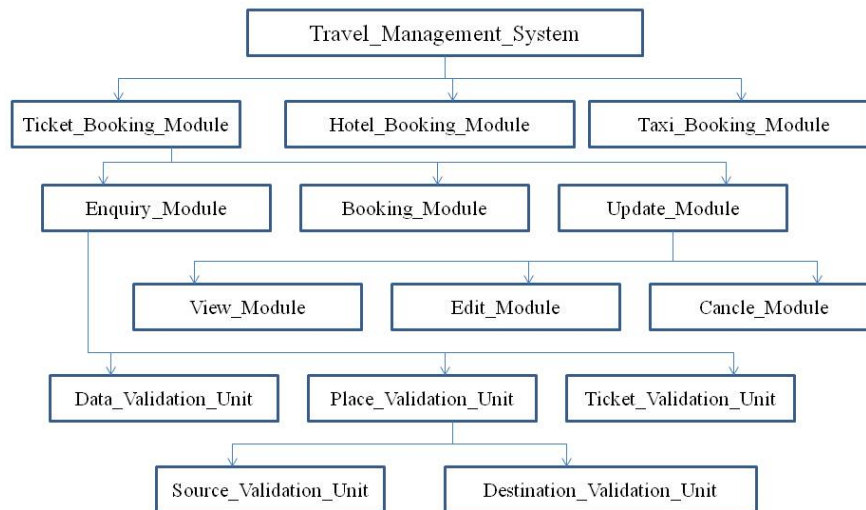**Summary:** identified the relevance of design principles

Viva Questions:

1. List all the design principles
2. Define encapsulation
3. Define coupling and cohesion
4. What do you mean by modularization
5. Define the term modularization

WEEK-6	INTEGRATION TESTING

OBJECTIVE: 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 Place_Validation_Unit**
- A Driver For Ticket_Validation_Unit
- **A Stub for Enquiry_Module** - **YES**
- A Driver for Enquiry_Module
- **A Stub For Ticket_Booking_Module** -**YES**
- 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?

A driver for Ticket_Booking_Module
A driver for Enquiry_Module
A driver for Place_Validation_Unit

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.

- **A Stub for Place_Validation_Unit**
- **A Stub for Enquiry_Module**

- **A Stub for Update_Module**
- **A Stub For Ticket_Booking_Module**

Summary: identified the usage of stubs or drivers in the context of integration testing
Viva Questions:
1. Define testing
2. Differentiate between debugging and testing
3. Differentiate between a stub and driver
4. What do you mean by a module
5. What is validation

WEEK-7	PERFORMANCE TESTING

OBJECTIVE: to identify the different types of integration 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 space craft 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 space craft 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?
3. 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 web site crashed. Which type of performance testing should have been done during web-site development to avoid this unpleasant situation?
4. During unexpected terrorist attack, one of the popular websites crashed as many people logged into the web-site 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 web-site development to avoid this unpleasant situation?

Scenarios	Performance Testing Type
Scenario 1	Performance testing
Scenario 2	Load testing
Scenario 3	Stress testing
Scenario 4	Endurance testing

Summary: Identified the different types of performance testing
Viva Questions:
1. Define performance testing
2. What do you mean by stress testing
3. List the categories of performance testing
4. What do you mean by load testing
5. Differentiate between white box and black box testing

WEEK-8	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.

Statement:
The incorporation of minor changes would have introduced new bugs into other modules, so regression testing should be carried out.

Summary: identified the relevant type of testing needed

Viva Questions:
1. What do you mean by system testing
2. List the classifications of requirements
3. What is sanity testing
4. What do you mean by maintenance with relate to any software
5. List all the phases of software development life cycle

WEEK-9	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, and 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 Carloan 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. Jonh'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.
2 John's productivity is more than Mike.
3 John introduced more defects than Mike.
4 John's effort is more than Mike.

Statement: 3
John introduced more defects than Mike.

Summary: understood how to use basic and derived quality metrics

Viva Questions:
1. What do you mean by a metric?
2. Give an example metric
3. What do you mean by quality
4. Define software quality assurance
5. What is software quality assurance plan

Viva-Questions (Software Engineering)

1 What is Software Engineering?
2 What is the difference between program and software?
3 Write out the reasons for the Failure of Water Fall Model.
4 What are the characteristics of the software?
5 Define the terms :
        a) Agility   b) Agile Team
6 What are the various categories of software?
7 What are the challenges in software?
8 Define software process
9 What are the fundamental activities of a software process?
10 What are the umbrella activities of a software process?
11 What are the merits of incremental model?
12 List the task regions in the Spiral model.
13 What are the drawbacks of spiral model?
14 What is System Engineering?
15 List the process maturity levels in SEIs CMM.
16 What is an effectors process?
17 Define the computer based system.
18 What does Verification represent?
19 What does Validation represent?
20 What is the difference between the "Known Risks" and Predictable Risks"?
21 What are the steps followed in testing?
22 Explain about the incremental model.
23 Explain in detail about the software process.
24 Explain in detail about the life cycle process.
25 Explain Spiral model and win-win spiral model in detail?
26 Name the Evolutionary process Models.
27 What are the Objectives of Requirement Analysis?
28 What is requirement engineering?
29 What are the various types of traceability in software engineering?
30 Define software prototyping.
31 What are the Requirements Engineering Process Functions?
32 What are the benefits of prototyping?
33 What are the prototyping approaches in software process?
34 What are the Difficulties in Elicitation?
35 What are the advantages of evolutionary prototyping?
36 What are the various Rapid prototyping techniques?
37 What is the use of User Interface prototyping?
38 What is System Modeling?
39 What are the characteristics of SRS?
40 What are the objectives of Analysis modeling?
41 What is data modeling?.What is a data object?
42 What is cardinality in data modeling?
43 What does modality in data modeling indicates?
44 What is ERD?
45 What is DFD?
46 What does Level0 DFD represent?
47 What is a state transition diagram?
48 Explain in detail about Functional Modeling.
49 Explain in detail about Structural Modeling.

50 Explain in detail about data modeling.
51 Explain about rapid prototyping techniques.
52 Explain the prototyping approaches in software process.
53 What are the elements of Analysis model?
54 What are the elements of design model?
55 How the Architecture Design can be represented?
56 Define design process. List the principles of a software design.
57 What is the benefit of modular design?
58 What is a cohesive module?
59 What are the different types of Cohesion?
60 What is coupling?
61 What are the various types of coupling?
62 What are the common activities in design process?
63 What are the benefits of horizontal partitioning?
64 What is vertical partitioning?
65 What are the advantages of vertical partitioning?
66 What are the various elements of data design?
67 List the guidelines for data design.
68 Name the commonly used architectural styles.
69 Explain in detail the design concepts.
70 Explain the design principles.
71 Explain the design steps of the transform mapping.
72 What are the testing principles the software engineer must apply while performing the software testing?
73 Define White Box Testing?
74 What are the two levels of testing?
75 What are the various testing activities?
76 Write short note on black box testing.
77 What is equivalence partitioning?
78 What is Regression Testing?
79 What is a boundary value analysis?
80 What are the reasons behind to perform white box testing?
81 What is cyclomatic complexity?
82 How to compute the cyclomatic complexity?
83 Distinguish between verification and validation.
84 What are the various testing strategies for conventional software?
85 Write about drivers and stubs.
86 What are the approaches of integration testing?
87 What are the advantages and disadvantages of big-bang?
88 What are the benefits of smoke testing?
89 What are the conditions exists after performing validation testing?
90 Distinguish between alpha and beta testing.
91 What are the various types of system testing?
92 Explain the types of software testing.
93 Explain in detail about Black box testing.
94 Explain about the software testing strategies.
95 What are the advantages and disadvantages of size measure?
96 Write short note on the various estimation techniques.
97 What is the Objective of Formal Technical Reviews?
98 What is COCOMO model?
99 Give the procedure of the Delphi method.

100 What is the purpose of timeline chart?
101 What is EVA?
102 What are the metrics computed during error tracking activity?
103 Why software change occurs?
104 Write about software change strategies.
105 Define CASE Tools.
106 What is software maintenance?
107 Define maintenance.
108 What are the types of software maintenance?
109 What is architectural evolution?
110 How the CASE tools are classified.
111 Explain about software cost estimation.