Lec#01 SUBJECTIVE

Q : What is software? (marks 2)

Ans. When we write a program for computer we named it as software. But software is not just
a program; many things other than the program are also included in software.

Q : How many items in software? Write name and explain . (marks 3)

Ans. □ ***Program*: **The program or code itself is definitely included in the software.
□ ***Data*: **The data on which the program operates is also considered as part of the
software.
□ ***Documentation*: **Another very important thing that most of us forget is
documentation. All the documents related to the software are also considered as part
of the software.

Q : What is Engineering? (marks 2)

Ans. The process of productive use of scientific knowledge is called engineering.

Q : Difference between computer science and software engineering. (marks 5)

Ans. When we use physics in making machines like engines or cars then it is called mechanical
engineering. And when we apply the knowledge of physics in

developing electronic devices then the process is called electrical engineering. The relation of computer science with software engineering is similar as the relation of physics with the electrical, mechanical or civil engineering or for that matter the relation of any basic science with any engineering field. So,This is the process of utilizing our knowledge of computer science in effective production of software systems.

Q : Difference between Software and Other Systems. (marks 3)

Ans. If the software has a bug and that bug was present in the older CD then that will remain in the new one. This is a fundamental difference between software and other systems.

Q : What is Software Crisis? (marks 3)

Ans. Software Crisis used to process the data of census. More powerful hardware resulted into the development of more powerful and complex software. Those very complex software was very difficult to write. So the tools and techniques that were used for less complex software became inapplicable for the more complex software.

Q :What is software engineering as by IEEE? (marks 2)

Ans. The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. And the IEEE stands for International institute of Electric and Electronic Engineering.

Q : What is software engineering used in software production? (marks 5)

Ans. Software Engineering is the combination of all the tools, techniques, and processes that used in software production.

- Programming Language
- Programming Language Design
- Software Design Techniques
- Tools
- Testing
- Maintenance
- Development

Q : Characteristics of well-Engineered software? (marks 3)

Ans.

- It is reliable
- It has good user-interface
- It has acceptable performance
- It is of good quality
- It is cost-effective

Q : What is Law of balancing act in software? (marks 5)
Ans.
Software Engineering is actually the balancing act. You have to balance many things like cost, user friendliness, Efficiency, Reliability etc. You have to analyze which one is the more important feature for your software is it reliability, efficiency, user friendliness or something

else. There is always a trade-off among all these requirements of software. It may be the case that if you try to make it more user-friendly then the efficiency may suffer. And if you try to make it more cost-effective then reliability may suffer. Therefore there is always a trade-off between these characteristics of software. These requirements may be conflicting

Lec#02 SUBJECTIVE

Q :.What is the Construction and major types of Construction? (marks 5)

Ans. The construction activities are those that directly related to the development of software,
e.g. gathering the requirements of the software, develop design, implement and test the software etc. Some of the major construction activities are listed below.

- Requirement Gathering
- Design Development
- Coding
- Testing

Q : Write the name of the major activities of management? (marks 3)

Ans.

- Project Planning and Management
- Configuration Management
- Software Quality Assurance
- Installation and Training

Q : Write the name of the major stage of software development loop? (marks 2)

Ans.

1. Problem Definition
2. Technical Development
3. Solution Integration
4. Status Quo

Q : How many software Engineering phase? and write the name. (marks 3)

Ans. There are four basic phase of S.E

1.Vision

2.Definition

3.Development

4.Maintenance

Lec#03 SUBJECTIVE

Q : Write the name of the role of software Requirement. (marks 3)

Ans.

1.Project Planning

2.project Tracking

3.Change Control

4.System Testing

5.User Dacumentation

6.Construction Process

Lec#04 SUBJECTIVE

Q : Write the name of the Requirement Statement Characteristics. (marks 3)

Ans.

1. Complete
2. Correct
3. Feasible
4. Necessary
5. Prioritized
6. Unambiguous
7. Verifiable

Q: What is Ambiguity? (marks 2)

ANS: Ambiguity means that two different readers of the same document interpret the
requirement differently.

Q: What is Gold-Plating? (Marks 3)

Ans: These called cool features. Which are not added in requirements but developers add this. Gold-plating refers to features are not present in the original requirement document
and in fact are not important for the end-user but the developer adds

them anyway
thinking that they would add value to the product.

Q: Write the names of level of requirements: (Marks 3)

ANS:

1. Business Requirements
2. User Requirements
3. Functional Requirements
4. Non-Functional Requirements

Q: What is Business Scope? (Marks 2)

ANS: In which document we write business requirements these called business scope individual

Q: What is Requirement definition? (Marks 2)

Ans: In which document we write user requirements these called requirement definition

Q: What is function specification? (Marks 2)

Ans: In which document we write functional/nonfunctional requirements these called function specification

Q: Define Business requirement: (Marks 2)

User will be able to correct spelling errors in a document efficiently and is will be integrated with he existing system.

Q: Define User requirement: (Marks 2)

Finding spelling errors in the document and decide whether to replace each misspelled word with the one chosen from a list of suggested words.

Q: Define Function requirement: (Marks 3)

- Find and highlight misspelled words
- Display a dialogue box with suggested replacements.
- Making global replacements.

Q: Define Non functional requirement: (Marks 2)

It must be integrated into the existing word-processor which runs on windows platform.

Lec#05 SUBJECTIVE

Q : Write the Use Case Model Components? (marks 3)

Ans.

There are two use case model component

1.Cases = case specifies a complete functionality

2.Actors= An actor is an entity that has an
interest in interacting with the system. An actor can be a human or
some other device or
system.

Q: What is scope? (marks 3)

ANs: Project scope defines the concept and range of the proposed solution, and limitations
identify certain capabilities that the product will not include. Clarifying the scope and
limitations helps to establish realistic stakeholder's expectations

Lec#06 SUBJECTIVE

Q :Write the name of Delete Information use case? (marks 2)

Ans. There are two existing use cases

1.Record Transaction

2.Cancel Transaction

Q : Write the name of Customer classes? (marks 2)

Ans. There are two classes.

1. Individual Customer
2. Corporate Customer

Q: What is Exception? (MARKS 2)

ANS: The system will not allow a user to delete information that is being used in the system.
The system will not allow a user to delete another user that has subordinates.

Q: What is Assumption Act? (Marks 3)

ANS: Deleting information covers a permanent deletion of an entire set of data such as a
commission plan, user, group etc. Deleting a portion of an entire set constitutes modifying the set of data.

Deleted information is not retained in the system.
A user can only delete information that has not been used in the system.

Q: Write the names of limitations of use case? (Marks 2)

ANS:

- Usability
- Reliability
- Performance
- Portability
- Access

What is elaborated Use case? Explain it. (marks 3)
Answer:
After the derivation of the use case model, each use is elaborated by adding detail of interaction between the user and the software system. An elaborated use case has the following components: Use Case Name, actors, summary, precondition, post-condition, extend, uses, normal course of events, alternative path, exception, assumption.

Q: UML stands for…..?

ANS: Unified Modeling Language

Lec#07 SUBJECTIVE

Q: What is Source? (Marks 3)

Ans: Sources of requirements are the origins from where the corresponding business process is initiated. By this concept, one has to trace from a requirement back to its origins to see who is involved in its initiation.

Q: What is Sink? (Marks 3)

ANS: Sink is the consumer of certain information. It is that entity which provides a logical end to a business process. Thus, 'sinks of requirements' is a concept that helps in identifying persons, organizations or external systems that gets certain functionality from the system.

Q.Write the name Techniques of Logical System Modals? (marks 3)

Ans.

1. User business processes
2. User activities for conducting the business processes
3. Processes that need to be automated
4. Processes which are not to be automated

Q : What is Business process model? (marks 2)

Ans. The first model that we will look at is called the process model. This model provides a high-level pictorial view of the business process. This model can be used as a starting point in giving the basic orientation to the reader of the document.

Q: There are three models used write their names: (marks 2)

ANs:

- Business Model
- State transition Model
- Data flow Model

Lec#08 SUBEJCTIVE

Q: What is STD? (Marks 2)

STD stands for State transition diagrams. This is another technique to document domain knowledge. This is an easy technique to design a work flow application.

Q : What Types of International Function Point User's Group (IFPUG)? (marks 2)

Ans. There are three types

1. External Inputs
2. External Outputs
3. External Inquiry

Q : What is a data flow model? And explain it. (marks 5)

Ans.

□ Captures the flow of data in a system.

□ It helps in developing an understanding of system's functionality.

□ What are the different sources of data, what different transformations take place on data and what are final outputs generated by these transformations.

□ It describes data origination, transformations and consumption in a system.

□ Information is organized and disseminated at different levels of abstraction. Thus this technique becomes a conduit for top down system analysis and requirements modeling.

Q : Write the types of shapes of notation. And explain it. (marks 5)

Ans. There are four shapes of notations

1. **Process**

□ What are different processes or work to be done in the system.

□ Transforms of data.

2. **External Agent**

External systems which are outside the boundary of this system. These are represented using the squares

3. **Data Store**

□ Where data is being stored for later retrieval.
□ Provides input to the process
□ Outputs of the processes may be going into these data stores.

4. **Data Flow**
□ Where the data is flowing.
□ Represents the movement of the data in a data flow diagram.

Q: What is External Inputs? (Marks 2)

Ans: An external input (EI) is an elementary process that processes data or control information
that comes from outside the application boundary.

Q: What is External Query? (Marks 3)

ANS: An external output (EO) is an elementary process that sends data or control information
outside the application boundary. The primary intent of an external output is to present
information to a user through processing logic other than, or in addition to, the retrieval
of data or control information.

Q : What is the difference between DFD and Flow Chart. (marks 5)

Ans.

DFD

- Processes on a data flow can operate in parallel.
- Looping and branching are typically not shown.
- Each process path may have a very Different timing.

Flow Chart

- Processes on flowcharts are sequential.

- Show the sequence of steps as an algorithm and hence looping and branching are part of flowcharts.

Lec#09 SUBJECTIVE

Q : Write the name of CRUD Operations? And explain it. (marks 3)

Ans.

These are four operations
□ **Create**: creates data and stores it.
□ **Read**: retrieves the stored data for viewing.
□ **Update**: makes changes in an stored data.
□ **Delete**: deletes an already stored data permanently.

Q : What is Common Mistakes in Data Flow Diagrams? (marks 2)

Ans.

- There is no input for the process Freeze Member Account
- In a similar manner, the process Create a New Member Account does not produce any output.

Lec#10 SUBJECTIVE

Q : What is GUI or Graphical user interface? (marks 2)

Ans. Graphical user interface is a computer interface that allows user to interact with a device through graphical elements such as pictures and animations, as opposed to text-based commands.

Q : What is Motivation for GUI? (marks 3)

Ans.

□ System users often judge a system by its interface rather than its functionality

□ A poorly designed interface can cause a user to make catastrophic errors
□ Poor user interface design is the reason why so many software systems are never
Used

Q : Write the three types of Pitfalls of using GUIs in Functional Specifications. (marks 3)

Ans.

□ UIs distract from business process understanding (what) to interfacing details
(how)
□ Unstable requirements cause frequent modifications in UIs
□ An extra work to be done at the requirement level each time a GUI change has to
be incorporated

Q :. What is Prototype? (marks 2)

Ans. Prototyping is yet another technique that can be used to reduce customer dissatisfaction at the requirement stage. A prototype is not the real product. It is rather just a real looking mock-up of what would be eventually delivered and might not do anything useful.

Cs 504 lec no 11

Q1: what parameters are used to measure and analyze design quality? 5 marks

Answer:- (Page 71) A software design can be looked at from different angles and different parameters can be used to measure and analyze its quality. These parameters include efficiency, compactness, reusability, and maintainability. A good design from one angle may not seem to be suitable when looked from a different perspective. For example, a design that yields efficient and compact code may not be very easy to maintain. In order to establish whether a particular design is good or not, we therefore have to look at the project and application requirements

Question No: 2 What should be consideration for maintain design? (Marks: 5)

Answer:- (Page 71) In order to make a design that is maintainable, it should be understandable and the changes should be local in effect. That is, it should be such that a change in some part of the system should not affect other parts of the system. This is achieved by applying the principles of modularity, abstraction, and separation of concern. If applied properly, these principles yield a design that is said to be more cohesive and loosely coupled and thus is easy to maintain.

Question No: 3 It is fact that good design makes maintenance easier. Which design principle help this to be achieved? (Marks: 3)

Answer:- (Page 71) A good design from one angle may not seem to be suitable when looked from a different perspective. For example, a design that yields efficient and compact code may not be very easy to maintain. In order to establish whether a particular design is good or not, we therefore have to look at the project and application requirements.

Q 4 :To manage the complexity of the system we need to apply the principles of separation of concern. Discuss briefly 2 MARKS

Answer: (Page 69) Separation of concern allows us to deal with different individual aspects of a problem by considering these aspects in isolation and independent of each other. A complex system may be divided into smaller pieces of lesser complexity called modules.

Q5: DEFINE Software Design Qualities ?

Ans : A software design can be looked at from different angles and different parameters can be used to measure and analyze its quality. These parameters include efficiency, compactness, reusability, and maintainability.

Q6 : what is data modeling ? 2 marks

Ans : s. Data modeling is an essential activity performed during the design phase. This includes the identification of data entities and their attributes, relationships among these entities, and the appropriate data structures for managing this data.

Lecture no 12

Q1:Define abstraction? 2 marks

Answer:- (Page 79) An abstraction is a technique in which we construct a model of an entity based upon its essential characteristics and ignore the inessential details.

Question No: 2 (Marks: 3) To manage the complexity of the system we need to apply the principle of abstraction. Discuss briefly?

Answer:- (Page 79) An abstraction is a technique in which we construct a model of an entity based upon its essential characteristics and ignore the inessential details. The principle of abstraction also helps us in handling the inherent complexity of a system by allowing us to look at its important external characteristic, at the same time, hiding its inner complexity. Hiding the internal details is called encapsulation.

Q 3 :What are architectural designs Process, explain briefly? 5 Marks

Answer:- (Page 79) System structuring: - System structuring is concerned with decomposing the system into interacting sub-systems. The system is decomposed into several principal sub-systems and communications between these sub-systems are identified. Control modeling:- Control modeling establishes a model of the control relationships between the different parts of the system. Modular decomposition:- During this activity, the identified sub-systems are decomposed into modules. This design process is further elaborated in the following section where architectural views are discussed.

Q4:Define cohesion 2 marks

. Answer: (Page 72) Cohesion is an internal property of a module. Cohesion describes the intra-component linkages while couple shows the inter-component linkages. Cohesion measures the independence of a module

Q 5:Define coupling ? 2 marks

Ans : Coupling is a measure of independence of a module or component. Loose coupling means that different system components have loose or less reliance upon each other.

Q 6:What is encapsulation? 2 marks

Ans : Hiding the internal details is called encapsulation

Question No: 7 (Marks: 5) What is action-oriented approach for Software Design?

Answer: (Page 80) In the case of action-oriented approach, data is decomposed according to functionality requirements. That is, decomposition revolves around function. In the OO approach, decomposition of a problem revolves around data. Action-oriented paradigm focuses only on the functionality of a system and typically ignores the data until it is required. Object- oriented paradigm focuses both on the functionality and the data at the same time. The basic difference between these two is decentralized control mechanism versus centralized control mechanism respectively. Decentralization gives OO the ability to handle essential complexity better than action-oriented approach.

Lecture no 13

Question No: 1 (Marks: 3) HOW DO YOU DETERMINE THAT AN OBJECTIVE BELONGS TO CERTAIN CLASS?

Answer:- (Page 85) The basic unit of object oriented design is an object. An object can be defined as a tangible entity that exhibits some well defined behavior. The structure and behavior of similar objects are defined in their common class. A class specifies an interface and defines an implementation.

Q2: What is behavior driven perceptive of an objective? 3 marks

Answer:- (Page 85) Behavior is how an object acts and reacts in terms of its state changes and message passing. The behavior of an object is completely defined by its actions. A message is some action that one object performs upon another in order to elicit a reaction. The operations that clients may perform upon an object are called methods

Q3: What is the difference between Aggregation and Association? 3 marks

Answer:- (Page 87) As compared to association, aggregation implies a tighter coupling between the two objects which are involved in this relationship. Therefore, one way to differentiate between aggregation and association is that if the two objects are tightly coupled, that is, if they cannot exist independently, it is an aggregation, and if they are usually considered as independent, it is an association.

Q4 : define Relationship Among Objects ? 2 marks

The object model presents a static view of the system and illustrates how different objects collaborate with one another through patterns of interaction. Inheritance, association and aggregation are the three inter-object relationships specified by the object model.

Q5 :Define the object model?

The elements of object oriented design collectively are called the Object Model. The object model encompasses the principles of abstraction, encapsulation, and hierarchy or inheritance.

Question No: 25 (Marks: 5) Code example of High Coupling

Answer: Click here for detail Tightly Coupled Example: public class CartEntry { public float Price; public int Quantity; } public class CartContents { public CartEntry[] items; } public class Order { private CartContents cart; private float salesTax; public Order(CartContents cart, float salesTax) { this.cart = cart; this.salesTax = salesTax; } public float OrderTotal() { float cartTotal = 0; for (int i = 0; i < cart.items.Length; i++) { cartTotal += cart.items[i].Price * cart.items[i].Quantity; } cartTotal += cartTotal*salesTax; return cartTotal; } }

Lecture no 14

Q1: What is Textual Analysis? Explain it 3 marks

Answer:- (Page 90) Textual analysis was developed by Abbot and then extended by Graham and others. In this technique different parts of speech are identified within the text of the specification and these parts are modeled using different components.

Q2 : name the four layers of the OO design pyramid ? 2 marks

The four layers of the OO design pyramid are:

1) The subsystem layer.
2) The class and object layer
3) The message layer
4) The responsibility layer

Q 3: define problem statement? 3 marks

A simple cash register has a display, an electronic wire with a plug, and a numeric keypad, which has keys for subtotal, tax, and total. This cash storage device has a total key, which triggers the release on the drawer..

Q4: Explain the four layer of the OO design pyramid ? 5 marks

1) The subsystem layer. Contains a representation of each of the subsystems that enable the software to achieve its customers defined requirements and to implement the technical infrastructure that supports customer requirements.

2) The class and object layer. Contains the class hierarchies that enable the system to be created using generalization and increasingly more targeted specializations. The layer also contains design representations for each object.

3) The message layer. Contains the details that enable each object to communicate with its collaborators. This layer establishes the external and internal interfaces for the system

. 4) The responsibility layer. Contains the data structures and algorithmic design for all attributes and operations for each object.

Lecture no 16

No: 1 (Marks: 5) How the objects are identified in peter codd's technique?

Answer: (Page 93) Objects are identifying in the following way.
Select actors : Actors are people and organizations that take part in the system under consideration. Examples of actors are: person, organization (agency, company, corporation, foundation)
. **Select Participants** A participant is a role that each actor plays in the system under consideration. Examples of participants are: agent, applicant, buyer, cashier, clerk, customer, dealer, and distributor. Etc.
Select Places : Places are where things come to rest or places that contain other objects. Examples of places are: airport, assembly-line, bank, city, clinic, country, depot, garage and hospital etc
. **Select Transactions :** Transactions are the "events". These transactions usually come from a window (GUI), some object which monitors for significant event and logs that information, or a another system that interacts with the system under consideration and logs some information. Examples of transactions are: agreement, assignment, authorization, contract, delivery, deposit, incident, inquiry, order, payment, problem report, purchase and sales etc.
Select Container Objects Containers are objects that hold other objects. e.g. bin, box, cabinet, folder, locker, safe, shelf, etc. Therefore a place is also a container but every container need not be a place.

Select Tangible things Take a "walk" through the system and select "tangible" things around you used in the problem domain. These may be characterized as all the remaining (not yet selected) "nouns" that make up the problem domain. Examples are: account, book, calendar, cash box, cash drawer, item, plan, procedure, product, schedule, skill, tool, etc

Lecture no 17

Q1: what is Identify Structures?

Ans : A structure is a manner of organization which expresses a semantically strong organization within the problem domain.

Q2: how many types of Identify Structures?

Ans : There are two type of structures
Generalization-Specialization (Gen-Spec) and whole-part.

Q3 : Define Attributes - ?

Ans : The first two activities would identify most of the objects (classes) in the problem domain. Now is the time to think about the role and responsibilities of these objects. The first thing to consider is their attributes,

Q4 : Show Collaborations (associations and aggregations) - Who I

know? 5 marks

The second step in establishing each object's responsibility is to identify and show how this object collaborates with other objects, i.e., who it knows. These collaborations can be identified with the help of the following steps:

1. For an actor, include an object connect to its participants (association).
2. For a participant, include an object connection to its actor (already established) and its transactions (association).
3. For a location, include object connections to objects that it can hold (association), to its part objects (aggregation), and to the transactions that are taking place at that location (association).
4. For transactions, include object connections to its participants (already established), its line items (aggregation), and its immediate subsequent transaction (aggregation).
5. For a transaction line item, include object connections to its transaction (already established), its item (association), a companion

Q5 : GIVE EXAMPLE ARE ATTRIBUTES ?

Ans : Examples of attributes are: number, name, address, date, time, operational state, phone, status, threshold, type, etc

lecture no 19

Q 1:Keeping in mind the Connie's case study, what rule of thumbs was identified, list them down ? 5 marks

Ans :

Who I Know - Rules of Thumb

1 :

an actor knows about its participants person knows about cashier

2 :

a transaction knows about its participants a session knows about its register and cashier

3

A transaction contains its transaction line items sale contains its sales line items

4

A transaction knows its sub transactions session knows about its sales sale knows about its payments

5

A place knows about its transactions store knows about its sessions

6

A place knows about its descriptive objects store knows about its tax categories

7

A container knows about its contents a store knows about its cashiers, items, and registers

Q2:

Keeping connie's case study in mind, as discussed in lecture, list down whole parts which were identified. Answer: (Page 100) 3 marks

Identify Whole-Part Structures

1 :

1 :A store as a whole is made up of cashiers, registers, and items.

2 :PA register contains a cash drawer.

3: A sale is constituted of sale line items.

Q3 :

Define data flow diagram ? 2 marks

Answer: (Page 100) A data flow diagram (DFD) is a graphical representation of the data through an information system, modeling its process aspects. Often they are c preliminary step used to create an overview of the system which can later be elabo

Lecture no 20 :

Q1: Why we use series of diagram? 3 marks

A series of diagrams can be used to describe the *dynamic behavior* of an object-oriented system. This is done in terms of a set of messages exchanged among a set of objects within a context to accomplish a purpose.

Q2 : Define the purpose of Interaction diagrams is to ? 3 marks

The purpose of Interaction diagrams is to:

- Model interactions between objects
- Assist in understanding how a system (a use case) actually works
- Verify that a use case description can be supported by the existing classes
- Identify responsibilities/operations and assign them to classes

Q3 : Define life line of objects ? 2 marks

Ans The boxes denote objects (or classes), the solid lines depict messages being sent from one object to the other in the direction of the arrow, and the dotted lines are called life-lines of objects..

q4 : write the syntax used for naming objects in a sequence diagram ? 3 marks

The syntax used for naming objects in a sequence diagram is as follows:

- syntax: [instanceName][:className]
- Name classes consistently with your class diagram (same classes).
- Include instance names when objects are referred to in messages or when several

Lecture No 21:

(Q) How Many Types of messages are there?

Ans. There Are four type of messages are there.

(1) Synchronous

(2) Asynchronous

(3) Create

(4) Destroy

Q No: 2. what is *Synchronous Messages?*

Synchronous messaging describes communications that takes place between two applications or systems, where the system places a message in a message queue (also called an Event Queue in enterprise messaging systems) and then waits for a message response before it continues processing. *Contrast with asynchronous messaging. They* are denoted by the full arrow.

Synchronous messaging is also known as synchronous communication.

(Q3) What is Asynchronous Messages?

Ans.
Asynchronous messages are "signals," denoted by a half arrow. They do not block the caller.
Asynchronous messages typically perform the following actions:

Create a new threat.
Create a new object.
Communication with threat is already
running.

Q4. *Object Creation and Destruction*

Ans. An object may create another object via a <<create>> message. Similarly an object may

Destroy another object via a <<destroy>> message. An object may also destroy itself. One

Should avoid modeling object destruction unless memory management is critical.

Q: No 5. Collaboration Diagrams depict Dynamic behavior of the system, explain it.

Ans.
Collaboration diagrams can also be used to depict the dynamic behavior of a system. They show how objects interact with respect to organizational units (boundaries!). Since a boundary shapes communication between system and outside world e.g. user interface or other system, collaboration diagrams can be used to show this aspect of the system. The sequence of messages determined by numbering such as 1, 2, 3, 4, This shows which operation calls which other operation.
Collaboration diagrams have basically two types of components: objects and messages.

Objects exchange messages among each-other. Collaboration diagrams can also show

synchronous, asynchronous, create, and destroy message using the same notation as used

In sequence diagrams. Messages are numbered and can have loops

Q: No 6. Evaluating the Quality of an Object-Oriented Design.

Ans. Judging the quality of a design is difficult. We can however look at certain object-

Oriented design attributes to estimate its quality. The idea is to analyze the basic principle

Of encapsulation and delegation to judge whether the control is centralized or distributed,

Hence judging the coupling and cohesion in a design. This will tell us how maintainable a

Design is.

You may also recall our earlier discussion of coupling and cohesion. It can be easy to see

That OO design yield more cohesive and loosely coupled systems.

Lecture No 22:

Q: No 1. What is Software Architecture?

Ans.

Software architecture refers to the fundamental structures of a software system, the discipline of creating such structures, and the documentation of these structures. These structures are needed to reason about the software system. Each structure comprises software elements, relations among them, and properties of both elements and relations,[1] along with rationale for the introduction and configuration of each element. The *architecture* of a software system is a metaphor, analogous to the architecture of a building.

Q: No 2.

Ans. Why is architecture important?

If a project has not achieved a system architecture, including its rationale, the

Project should not proceed to full-scale system development. Specifying the

Architecture as a deliverable enables its use throughout the development and

Maintenance process.

Q: 3. No Architectural design process

Ans. Just like any other design activity, design of software architecture is a creative and Iterative process. This involves performing a number of activities, not necessarily in any Particular order or sequence. These include system structuring, control modeling, and Modular decomposition.

Q: No 4. Architectural Attributes

Ans. Software architecture must address the non-functional as well as the functional Requirements of the software system. This includes performance, security, safety, Availability, and maintainability.

Q: No 5. What Is Performance in Architectural Design?

Ans. – Performance can be enhanced by localizing operations to minimize sub-System communication. That is, try to have self-contained modules as Much as possible so that inter-module communication is minimized.

Q: No 6. What is Security in Architectural Design?

Ans. – Security can be improved by using a layered architecture with critical Assets put in inner layers.

Q: No. 7. what is Safety in Architectural design

Ans. – Safety-critical components should be isolated

Q: No what is Availability in **Architectural design**

Ans. – Availability can be ensured by building redundancy in the system and Having redundant components in the architecture.

Q: No 8. What is Maintainability in Architectural design?

Ans. – Maintainability is directly related with simplicity. Therefore, Maintainability can be increased by using fine-grain, self-contained Components.

Q: No 9. What are architectural designs Process, explain briefly?

Ans. Just like any other design activity, design of software architecture is a creative and Iterative process. This involves performing a number of activities, not necessarily in any

Particular order or sequence. These include system structuring, control modeling, and

Modular decomposition.

System structuring: - System structuring is concerned with decomposing the system into interacting sub-systems. The system is decomposed into several principal sub-systems and communications between these sub-systems are identified.

Control modeling: - Control modeling establishes a model of the control relationships between the different parts of the system.

Modular decomposition: - During this activity, the identified sub-systems are decomposed into modules. This design process is further elaborated in the following section where architectural views are discussed.

Q: No 10. Differentiate between architectural design and system architecture in a single line?

Ans. Architecture faces towards strategy, structure and purpose, towards the abstr
faces towards implementation and practice, towards the concrete.