18CSC206J SOFTWARE ENGINEERING AND PROJECT MANAGEMENT
UNIT-III
SOFTWARE CONSTRUCTION

1 MARKS

S.NO	QUESTIONS	LEVEL	CLO	PG. NO
1.	_____ is an industry strength software product of a large size requires stringent coding standards. A.Coding B.Coding Methods C.Coding Framework **D.Constructing**	1	3	176
2.	Converting the specifications into software code is totally dependent on the _____ Team. A.Coding B.Developing C.Debugging **D.Constructing**	1	3	176
3.	_____increases software code reuse and enhances productivity of developers. **A.Modularity** B.Simplicity C.Clarity D.Relability	1	3	177
4.	Standard naming conventions can be used so that the code has _____ A.Modularity B.Simplicity **C.Clarity** D.Relability	1	3	177
5.	Object-oriented programming, abstraction and information hiding can be used to add_____ A. Degree of Modularity **B.Degree of Simplicity**	1	3	177

C.Degree of Clarity

D.Degree of Relability

6.	_____ is one of the most important aspects of industry strength software products.	1	3	177

A.Modularity

B.Simplicity

C.Clarity

D.Relability

7.	To Ensure safety, the software product must have the error less than _____.	1	3	178

A.0.00001%

B.0.01%

C.0.000001%

D.0.001%

8.	_____will ensure a consistent coding production with standard code that will be easy to debug and test.(L2,CLO3),178	1	3	34

A.Coding

B.Coding Methods

C.Coding Framework

D.Constructing

9.	Which is the most labor intensive phase in software development? (L1,CLO3),178	1	3	36

A.Software Coding

B.Software Developing

C.Software Debugging

D.Software Constructing

10.	_____is a powerful tool to eliminate defects and improve software code.(L2,CLO3),179	1	3	32

A. **Deskcheck**

B. Walkthrough

C. Inspection

D. Code Review

11.	_____ is the formal code review initiated by	1	3	32

developer.(L2,CLO3),179

A.Deskcheck

B. Walkthrough

C. Inspection

D. Code Review

12. _____is the fnal review of the software code.(L2,CLO3),179 1 3 34

A. Deskcheck

B. Walkthrough

C. Inspection

D. Code Review

13. _____enables programmers to store large pieces of code inside procedures and functions.(L2,CLO3),180 1 3 36

A. **Structured programming**

B. Object oriented programming

C.Automatic code generation

D. Pair programming

14. Which among the following firm is working to develop automatic code generation system?(L1,CLO3),180 1 3 36

A. **Sun Microsystems**

B. HP

C. IBM

D. Dell

15. SOA has been evlolved recently for the purpose of _____.(L1,CLO3),181 1 3 36

A.Software construction

B. Inspection

C. Code Reuse

D. Code Review

16. Which technique isused in Test driven development?(L2,CLO3),181 1 3 37

A.SOA

B. eXtreme programming

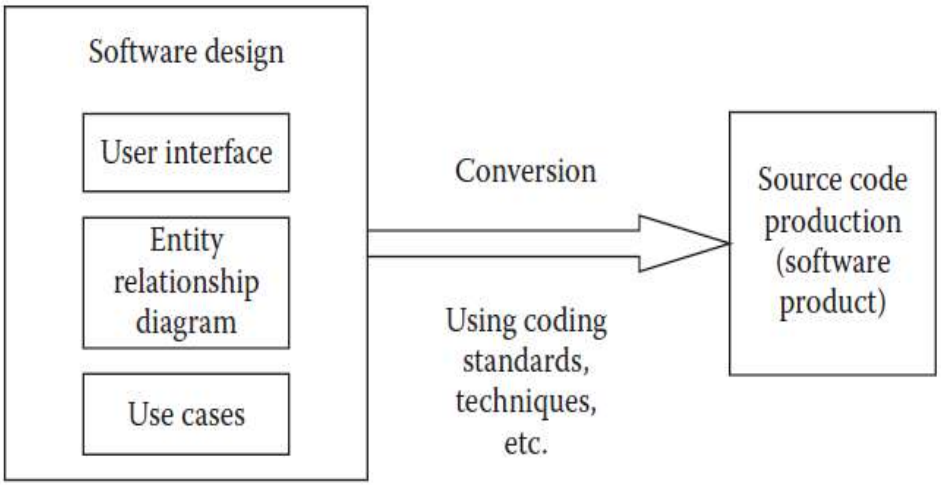
C. Scrum

D.Reuse

17.	_____is the quality driven development technique employed in the eXtreme Programming.(L2,CLO3),181	1	3	38
	A. Structured programming			
	B. Object oriented programming			
	C. Automatic code generation			
	D. **D. Pair programming**			
18.	_____plays an important role in the construction phase.(L2,CLO3),181	1	3	39
	A.Confguration management			
	B.Coding			
	C.Coding Methods			
	D.Coding Framework			
19.	_____ phase is one of the most labor intensive phases in software development cycle.(L1,CLO3),183	1	3	41
	A. **Software construction**			
	B. B.Code Generation			
	C. Automatic Code Generation			
	D. D.Coding			
20.	Which phase generates the complete source code of the application.(L1,CLO3),183	1	3	56
	A.Software construction			
	B.Code Generation			
	C. Automatic Code Generation			
	D.Coding			

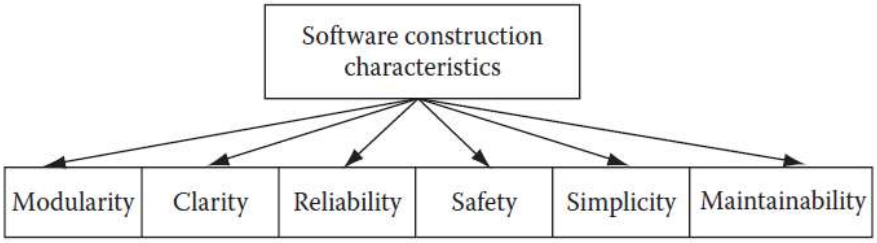
4 MARKS:

1.	Explain Source code production from software design with neat sketch. Developers are given software design specifications in the form of use cases, flow diagrams, UI mock ups, etc., and they are supposed to write a code so that the built software matches these specifications. Converting the specifications into software code is totally dependent on the construction team. How well they do it depends on their experience, skills, and the process they follow to do their job. Apart from these facilities, they also need some standards in their coding so that the work is fast as well as has other benefits like maintainability, readability, and reusability.	L2	CLO3



2. List the coding standards in software construction. L1 CLO3

Some of the coding standards include standards for code modularity, clarity, simplicity, reliability, safety, and maintainability.

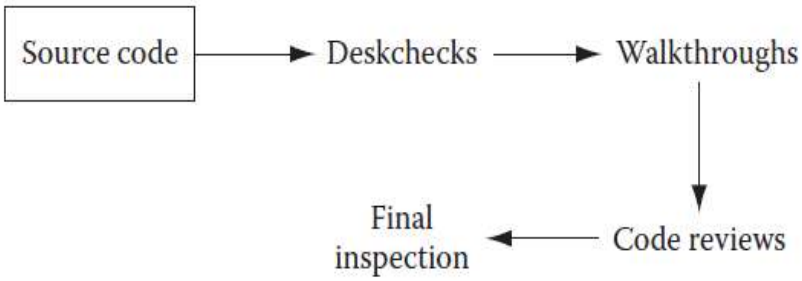


3. Elaborate in detail about the Quality Control. L2 CLO3

It is estimated that almost 70% of software defects arise from faulty software code. To compound this problem, software construction is the most labor intensive phase in software development. Any construction rework means wasting a lot of effort already put in. Moreover, it is also a fact that it is cheaper to fix any defects found during construction at the phase level itself. If those defects are allowed to go in software testing, then fixing those defects will become costlier. That is why review of the software code and fixing defects is very important.

4. Categorize the techniques to ensure the quality of the written code. L2 CLO3

There are some techniques available like deskchecks , walkthroughs, code reviews, inspections, etc. that ensure quality of the written code



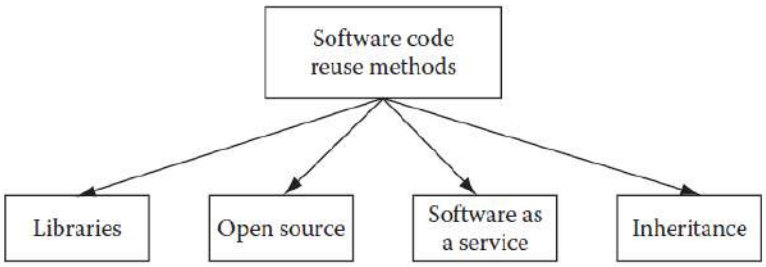
5. Elaborate the evolution of different programming techniques in coding methods. L2 CLO3

Different programming techniques include,

- Structured Programming
- Object-Oriented Programming
- Automatic Code Generation
- Test-Driven Development
- Pair Programming.

Explain in detail about the Code reuse methods. L2 CLO3

Many techniques have evolved to reduce the labor intensive nature of writing source code. Software code reuse is one such technique. Making a block of source code to create a functionality or general utility library and using it at all places in the source code wherever this kind of functionality or utility is required is an example of code reuse. Code reuse in procedural programming techniques is achieved by creating special functions and utility libraries and using them in the source code. In object-oriented programming, code reuse is done at a more advanced level. The classes Software Construction containing functions and data themselves can not only be reused in the same way as functions and libraries, but the classes can also be modified by way of creating child classes and using them in the source code.



7. Explain the importance of configuration management in construction phase. L2 CLO3

Configuration management plays an important role in the construction phase. Due to changes in requirements and design, an already developed source code needs to be changed. So it happens that the development team ends up with many versions of a source code during the project. If the version control management is not handled properly, then many developers may start working on a wrong version of source code, and thus a lot of rework may be needed in the end. There is one more dimension to configuration management for the construction phase. During construction, many software builds are maintained for different versions of the product being developed. These builds can break if a bad piece of code is checked into the build by any developer. When the build is broken, then no other developer can check in his code.

Thus, development is halted until the build is rebuilt with the correct code. Imagine what may happen in the case of distributed teams located at far-flung locations with different time zones and a central build is being maintained. It will be difficult to communicate and manage the build process in such a scenario. In such scenarios, smoke test application can be deployed, which can run whenever a new code is checked-in in the build. If the smoke test fails, that means the build has failed and thus the automated system can e-mail the build information to concerned people. If the build fails, then the developer who had checked-in in the code gets the message and immediately tries to fix the build. Once the build is fixed, then other developers can check-in their code.

8. Discuss about the unit testing in detail. L2 CLO3

Whenever a developer writes a piece of code, he feels confident that he has written a clean code and that it does not need testing. But most of the time he is wrong. It is because no source code is perfect, especially the first time. Only after some rounds of review it becomes perfect. At the same time, it is very difficult to review one's own code. That is why a quality control measure is taken in form of unit testing to ensure that developers test their codes themselves and only then can submit their code if the code passes the unit tests .

For unit testing, generally developers are comfortable as long as there are no changes required (due to change in design or requirements) in their code. But once some change takes place in the code somewhere, other things change. What would be the impact of that change on other parts of the software product under development?

Similarly what impact will it have on their own code if changes take place in other modules being written by other people? Generally, it is one of the most challenging situations in software construction to find the impact of change on other parts of the product under development. Such situations call for unit testing of the written code, and no piece of code should go to build without doing this. A formal and rigid adherence to unit tests should be a must for all source codes being written and no liberty should be allowed.

9. Discuss about the Integration testing in detail. L2 CLO3

Most software development is done after partitioning the software application under development first and then allocating it to distributed teams. Generally, modules of code are developed first. Later, they need to be integrated with each other to make a complete software application. Modules are integrated with each other through open interfaces. Whether or not the integration is working fine, it must be tested to ensure integration has been achieved. This kind of testing is known as integration testing.

Integration testing has been becoming more and more important, as most software being developed is modular in nature. With the advent of SOA, which is all about loosely coupled software components, integration testing has become even more important.

10. Explain in detail about software construction Artifacts. L2 CLO3

The software construction phase is one of the most labor intensive phases in software development cycle. This phase generates the complete source code of the application. Apart from source code, documentation is also made so that when any maintenance is required on the built application, the source code could be well understood, and changing any source code will be easy. Review reports are also generated after reviews are conducted.

11. Summarize about the Pair Programming. L2 CLO3

Pair programming is a quality driven development technique employed in the eXtreme Programming development model. Here, each development task is assigned to two developers. While one developer writes the code, the other developer sits behind him and guides him through the requirements (functional, nonfunctional). When it is the turn of the other developer to write the code, the first developer sits behind him and guides him on the requirements. So developers take turns for the coding and coaching work. This makes sure that each developer understands the big picture and helps them to write better code with lesser defects.

12. Explain about the Test-Driven Development. L2 CLO3

This concept is used with iteration-based projects especially with eXtreme Programming technique. Before developers start writing source code, they create test cases and run the tests to see if they run properly and their logic is working. Once it is proved that their logic is perfect, only then they write the source code. So here, tests drive software development, and hence it is appropriately named test-driven development.

13. Explain in detail about Automatic Code Generation. L2 CLO3

Constructing and generating software code is very labor intensive work. So there has always been fascination about automatic generation of software code. Unfortunately, this is still a dream. Some CASE and modeling tools are available that generate software code. But they are not sophisticated. They are also not complete. Then there are business analyst platforms developed by many ERP software vendors that generate code automatically when analysts configure the product. These analyst

platforms are first built using any of the software product development methodologies. The generated code is specific to the platform and runs on the device (hardware and software environment) for which the code is generated.

Generally, any code consists of many construction unit types. Some of these code types include control statements such as loop statements, if statements, etc., and database access, etc. Generating all of the software code required to build a software application is still difficult. But some companies like Sun Microsystems are working to develop such a system.

14. Elaborate Peer reviews. L2 CLO3

Peer Reviews are employed when a complete review of the source code is not important. Here, the developer sends his piece of code to the designated team members. These team members review the code and send feedback and comments to the developer as suggestions for improvement in the code. The developer reads those feedbacks and may decide to incorporate or to discard those suggestions. So this form of review is totally voluntary. Still, it is a powerful tool to eliminate defects or improve software code.

15. Explain in detail about Reliability. L2 CLO3

Reliability is one of the most important aspects of industry strength software products. If the software product is not reliable and contains critical defects, then it will not be of much use for end users. Reliability of source code can be increased by sticking to the standard processes for software construction. During reviews, if any defects are found, they can be fixed easily if the source code is neat, simple, and clear.

Reliable source code can be achieved by first designing the software product with future enhancement in consideration as well as by having a solid structure on which the software product is to be built. When writing pieces of source code based on this structure, there will be little chance of defects entering into the source code. Generally during enhancements, the existing structure is not able to take load of additional source code and thus the structure becomes shaky. If the development team feels that this is the case, then it is far better to restructure the software design and then write a code based on the new structure than to add a spaghetti code on top of a crumbling structure.

12 Marks:

1. Categorize the various coding standards and explain its characteristics with examples. L2 CLO3
2. Classify the different kinds of reviews done at different stages in software code writing. L2 CLO3
3. List the techniques to ensure the quality of written code and discuss them in detail. L2 CLO3
4. Elaborate in detail about Quality control. L2 CLO3
5. Categorize the various coding methods and explain them in detail. L2 CLO3
6. Classify the different kinds of programming techniques and elaborate in detail L2 CLO3
7. Compare and contrast unit testing and integration testing with appropriate scenarios. L2 CLO3
8. Discuss in detail about L2 CLO3
   a. Pair Programming
   b. Test driven development
   c. Object oriented programming
9. Explain the following in detail L2 CLO3
   a. Structured Programming

b. Automatic code generation
c. Software code reuse

10. Explain in detail about L2 CLO3
a. Configuration management
b. software construction Artifacts

18CSC206J SOFTWARE ENGINEERING AND PROJECT MANAGEMENT
UNIT-II
SOFTWARE DESIGN
1 MARKS

S.NO	QUESTION	LEVEL	CLO	PG. NO
1.	A program should not have any bugs that inhibit its function is called as ---------- a. Commodity b. Delight c. **Firmness** d. Analysis	1	2	216
2.	A program should be suitable for the purposes for which it was intended is called as ---------- a. **Commodity** b. Delight c. Firmness d. Analysis	1	2	216
3.	The experience of using the program should be a pleasurable one is called as ----------. a. Commodity b. **Delight** c. Firmness d. Analysis	1	2	216
4.	The objects and relationships defined in the ---------- diagram provide the basis for the data design action. a. **CRC diagram** b. Activity diagram c. Class diagram d. Usecase diagram	1	2	217
5.	The---------- design describes how the software communicates with systems that interoperate with it, and with humans who use it. a. Interface	1	2	217

b. Collaboration

c. Activity

d. package

6.	Which design is used to transforms structural elements of the software architecture In to a procedural description of software components?	2	2	217

a. Interface

b. **Component**

c. Activity

a. package

7.	---------- is assessed by evaluating the feature set and capabilities of the Program.	1	2	220

a. **Functionality**

b. Usability

c. Reliability

d. Performance

8.	What is measured by considering processing speed, response time, resource consumption, throughput, and efficiency?	1	2	220

a. Reliability

b. **Performance**

c. Functionality

d. Usability

9.	An ---------- Abstraction refers to a sequence of instructions that have a specific and limited function.	1	2	223

a. design

b. **procedural**

c. data

d. component

10.	A --------- Abstraction is a named collection of data that describes a data object.	1	2	223

a. **design**

b. procedural

c. data

d. component

11.	---------- represent architecture as an organized collection of program components. a. **Structural models** b. Framework models c. Dynamic models d. Process models	1	2	224
12.	---------- increase the level of design abstraction by attempting to identify repeatable architectural design frameworks that are encountered in similar types of applications. a. Structural models b. **Framework models** c. Dynamic models d. Process models	1	2	224
13.	Which models address the behavioral aspects of the program architecture, indicating how the structure or system configuration may change as a function of external events? a. Structural models b. Framework models c. **Dynamic models** d. Process models	1	2	224
14.	Which models focus on the design of the business or technical process that the system must accommodate? a. Structural models b. Framework models c. Dynamic models d. **Process models**	1	2	224
15.	---------- can be used to represent the functional hierarchy of a system. a. Structural models b. **Functional models** c. Dynamic models d. Process models	1	2	224

	Which is an indication of the relative functional strength of a module?	1	2	227
	a. **Cohesion**			
	b. Coupling			
	c. Elaboration			
	d. refactoring			
17.	---------- is an indication of the relative interdependence among modules.	1	2	227
	a. Cohesion			
	b. **Coupling**			
	c. Elaboration			
	d. refactoring			
18.	Which is a reorganization technique that simplifies the design (or code) of a component without changing its function or behavior?	1	2	229
	a. a. Cohesion			
	b. Coupling			
	c. Elaboration			
	d. **refactoring**			
19.	Which class is used to represent data stores that will persist beyond the execution of the software?	1	2	230
	a. System classes			
	b. **Persistent classes**			
	c. Business domain classes			
	d. User interface classes			
20.	The ---------- dimension indicates the evolution of the design model as design tasks are executed as part of the software process.	1	2	233
	a. Component			
	b. Abstraction			
	c. **Process**			
	d. deployment			
21.	The ---------- dimension represents the level of detail as each element of the analysis model is transformed into a design equivalent and then refined iteratively.	1	2	233
	a. Component			

b. **Abstraction**

c. Process

d. deployment

22.	---------- Elements indicate how software functionality and subsystems will be allocated within the physical computing environment that will support the software.	1	2	233

a. Component level design

b. **Deployment-level design**

c. Architectural Design Elements

d. Interface Design Elements

23.	Which design is used to represents the structure of data and program components that are required to build a computer-based system?	1	2	242

a. Pattern oriented design

b. Web application design

c. **Architectural design**

d. Component level design

24.	A data store resides at the center of ---------- architecture and is accessed frequently by other components that update, add, delete, or modify data within the store.	1	2	250

a. Object oriented

b. **Data center**

c. Data flow

d. Call & return

25.	---------- Architecture is applied when input data are to be transformed through a series of computational or manipulative components into output data.	1	2	250

a. Object oriented

b. Data center

c. **Data flow**

d. Call & return

26.	The systems that use the target system as part of some higher-level processing scheme is called as ----------	1	2	256

a. **Superordinate systems**

b. Subordinate systems

c. Peer-level systems

d. Actors

27.	The system that are used by the target system and provide data or processing that are necessary to complete target system functionality is known as ----------	1	2	256

a. Superordinate systems

b. **Subordinate systems**

c. Peer-level systems

d. Actors

28.	The information is either produced or consumed by the peers and the target system is called as ----------	1	2	256

a. Superordinate systems

b. Subordinate systems

c. **Peer-level systems**

d. Actors

29.	Which dependencies is used to represent dependence relationships among consumers who use the same resource or producers who produce for the same consumers?	1	2	263

a. **Sharing**

b. Flow

c. Constrained

d. data

30.	Which dependency represents relationships between producers and consumers of resources?	1	2	263

a. Sharing

b. **Flow**

c. Constrained

d. data

31.	---------- Dependencies represent constraints on the relative flow of control among a set of activities.	1	2	264

a. Sharing

b. Flow

c. **Constrained**

d. data

32. A --------- Component that coordinates the invocation of all other problem domain components. 1 2 279

a. Problem domain component

b. Infrastructure component

c. **Control component**

d. Design component

33. What type of a component implements a complete or partial function that is required by the customer. 1 2 279

a. **Problem domain component**

b. Infrastructure component

c. Control component

d. Design component.

34. Which component is responsible for functions that support the processing required in the problem domain? 1 2 279

a. Problem domain component

b. **Infrastructure component**

c. Control component

d. Design component

35. Which level of cohesion occurs when a component performs a targeted computation and then returns a result? 1 2 287

a. **Functional**

b. Layer

c. Communicational

d. congestion

36. What type of cohesion occurs when a higher layer accesses the services of a lower layer, but lower layers do not access higher layers? 1 2 287

a. Functional

b. **Layer**

c. Communicational

d. congestion

	All ---------- operations that access the same data are defined within one class. a. Functional b. Layer c. **Communicational** d. congestion	1	2	287
38.	Which coupling Occurs when operation A() invokes operation B() and passes a control flag to B? a. **Control coupling** b. Layer c. Communicational d. congestion	1	2	289
39.	Which coupling occurs when operations pass long strings of data arguments? a. Control coupling b. **Data** c. Communicational d. congestion	1	2	289
40.	Which coupling occurs when one component “surreptitiously modifies data that is internal to another component”? a. Control coupling b. Data c. Communicational d. **content**	1	2	289
41.	The intent of ---------- engineering is to identify, construct, catalog, and disseminate a set of software components that have applicability to existing and future software in a particular application domain. a. data b. **domain** c. pattern d. structure	1	2	303
42.	Which focuses on the profile of the users who will interact with the System?	1	2	320

a. **Interface analysis**

b. interface design

c. Interface construction

d. Interface validation

43.	---------- is to define a set of interface objects and actions that enable a user to perform all defined tasks.	1	2	320

a. Interface analysis

b. **interface design**

c. Interface construction

d. Interface validation

44.	Which is normally begins with the creation of a prototype that enables usage scenarios to be evaluated?	1	2	320

a. Interface analysis

b. interface design

c. **Interface construction**

d. Interface validation

45.	---------- focuses on the ability of the interface, the degree to which the interface is easy to use and easy to learn, and the users' acceptance of the interface as a useful tool in their work.	1	2	320

a. Interface analysis

b. interface design

c. Interface construction

d. **Interface validation**

46.	Which patterns focus on the "creation, composition, and representation" of Objects?	1	2	350

a. **Creational pattern**

b. Structural patterns

c. Behavioral patterns

d. Object pattern

47.	---------- focus on problems and solutions associated with how classes and objects are organized and integrated to build a larger structure.	1	2	350

a. Creational pattern

b. **Structural patterns**

c. Behavioral patterns

d. Object pattern

48.	Which pattern address problems associated with the assignment of responsibility between objects and the manner in which communication is effected between objects?	1	2	351

a. Creational pattern

b. Structural patterns

c. **Behavioral patterns**

d. Object pattern

49.	Which pattern relate to the overall structure of the information space, and the ways in which users will interact with the information.	1	2	362

a. **Information architecture patterns**

b. Interaction patterns

c. Navigation patterns

d. Presentation patterns

50.	---------- contribute to the design of the user interface.	1	2	362

a. Information architecture patterns

b. **Interaction patterns**

c. Navigation patterns

d. Presentation patterns

4 MARKS:

1.	Write short notes on Software Design.	L2	CLO2

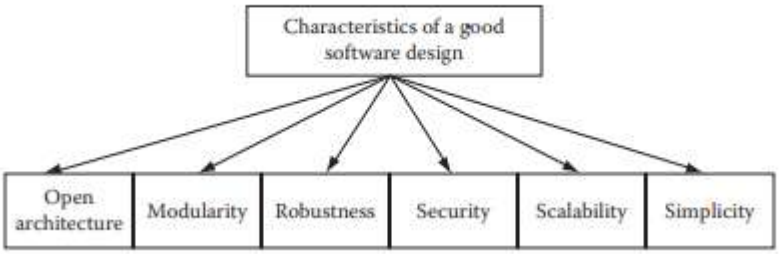
Software design follows the software requirement phase. Based on the requirements, software is designed in such a way that the features required in the requirements document can be implemented in the software design. Apart from how the features as per functional requirements can be implemented, design also considers factors such as reliability, robustness, security, ease of use, internationalization, localization, and compatibility. All of these are collectively termed as non-functional design requirements.

2.	What are the important roles of Software Design	L1	CLO2

Software design plays an important role in software development. If the design is good, software will have fewer defects and may be considered reliable. Due to requirement creep as mentioned in a previous section, the design may get unstable, which may lead to a poor quality product. Enterprise software products though may

have a lot of features; nevertheless, they need to have open interfaces so that they can be integrated with other software products

3. Draw the Characteristics of a good software design L1 CLO2



4. What are the types of Software Design? L1 CLO2

Software design on any project may consist of many work products, which together can be termed the software design for the software product that will be built during the software project. Some examples include prototypes, structural models, object-oriented design, systems analysis, and entity relationship models.

5. Write short notes on Design Standards. L2 CLO2

If design standards are implemented on a project, then it will help in streamlining activities that are involved during the software design phase. Some industry standards for software design include operator interface standards, test scenarios, safety standards, design constraints, and design tolerances.

6. Difference between Top-Down approach and Bottom-Up Approach. L2 CLO2

In the top-down approach, the top structure of the product is conceived and designed first. Once the structure is perfected, components that will make the product are designed. Once the major components are designed, the features that make the component are designed.

In the bottom-up approach, first, the minute functions of the software product are structured and designed. Then, the middle-level components are designed, and, finally, the top-level structure is designed. Once some components are designed, they can be shown to the customer, and a buy in can be made for the project.

7. Write short notes on Architectural Design. L2 CLO2

Architectural design represents the structure of data and program components that are required to build a computer-based system. It considers the architectural style that the system will take, the structure and properties of the components that constitute the system, and the interrelationships that occur among all architectural components of a system

8. Why is Architecture Design Important? L2 CLO2

• Representations of software architecture are an enabler for communication between all parties (stakeholders) interested in the development of a computer-based system.
• The architecture highlights early design decisions that will have a profound impact on all software engineering work that follows and, as important, on the ultimate success of the system as an operational entity.
• Architecture "constitutes a relatively small, intellectually graspable model of how the system is structured and how its components work together"

9. Compare Data-Centered and Data-Flow architectures. L2 CLO2

A data store (e.g., a file or database) resides at the center of this architecture and is accessed frequently by other components that update, add, delete, or otherwise modify data within the store. Client software accesses a central repository. In some cases the data repository is passive.

Data-Flow architecture is applied when input data are to be transformed through a series of computational or manipulative components into output data. A pipe-and-filter pattern has a set of components, called filters, connected by pipes that transmit data from one component to the next.

10. Distinguish between Object-Oriented and Layered Architectures L2 CLO2

The components of a system encapsulate data and the operations that must be applied to manipulate the data. Communication and coordination between components are accomplished via message passing.

A number of different layers are defined, each accomplishing operations that progressively become closer to the machine instruction set. At the outer layer, components service user interface operations. At the inner layer, components perform operating system interfacing. Intermediate layers provide utility services and application software functions.

11. Define the term Archetypes. L1 CLO2

An archetype is a class or pattern that represents a core abstraction that is critical to the design of an architecture for the target system. In general, a relatively small set of archetypes is required to design even relatively complex systems. The target system architecture is composed of these archetypes, which represent stable elements of the architecture but may be instantiated many different ways based on the behavior of the system

12. How to Refine the Architecture into Components L2 CLO2

The software architecture is refined into components, the structure of the system begins to emerge. You begin with the classes that were described as part of the requirements model. These analysis classes represent entities within the application (business) domain that must be addressed within the software architecture. The architecture must accommodate many infrastructure components that enable application components but have no business connection to the application domain.

13. What is meant by Module Division (Refactoring)? L2 CLO2

Whenever a software product is designed, it is done with good intentions. Care is taken to ensure that the design is extensible, so that when customer needs increase over time, the product can be extended to take care of those increased needs. one technique is employed, which is known as refactoring. Using refactoring, the internal design of a piece of software code is improved by decreasing coupling among classes of objects and increasing cohesion among classes. Refactoring is very similar to the concept of normalization in relational databases

14. Write short notes on Module Coupling. L2 CLO2

One area similar to refactoring is coupling between modules. As products mature and more and more lines of code are added to the existing product, coupling between modules tends to increase. This has a profound impact when any changes in code are required. To reduce the chances of product defects, it is necessary to reduce the number of calls among different modules and classes. SOA architecture provides great help here. SOA architecture essentially promotes loose coupling.

15. Define Component-level design L1 CLO2

Component-level design occurs after the first iteration of architectural design has been completed. At this stage, the overall data and program structure of the software has been established. The intent is to translate the design model into operational software.

16. Difference between Cohesion and Coupling L2 CLO2

Cohesion as the "single-mindedness" of a component. Within the context of component-level design for object-oriented systems, cohesion implies that a component or class encapsulates only attributes and operations that are closely related to one another and to the class or component itself.

Coupling is a qualitative measure of the degree to which classes are connected to one another. As classes (and components) become more interdependent, coupling increases. An important objective in component-level design is to keep coupling as low as is possible.

17. What are the different types of Coupling? L1 CLO2

- Content coupling
- Common coupling
- Control coupling
- Stamp coupling
- Data coupling
- Routine call coupling
- Type use coupling
- Inclusion or import coupling
- External coupling

18. What are the steps for conducting Component-Level Design? L2 CLO2

- Identify all design classes that correspond to the problem domain.
- Identify all design classes that correspond to the infrastructure domain
- Elaborate all design classes that are not acquired as reusable components
- Describe persistent data sources (databases and files) and identify the classes required to manage them
- Develop and elaborate behavioral representations for a class or component

19. Define the term User interface design L1 CLO2

User interface design creates an effective communication medium between a human and a computer. Following a set of interface design principles, design identifies interface objects and actions and then creates a screen layout that forms the basis for a user interface prototype.

20. What are the steps involved in User-Interface Design? L2 CLO2

1. Using information developed during interface analysis, define interface objects and actions (operations).
2. Define events (user actions) that will cause the state of the user interface to change. Model this behavior.
3. Depict each interface state as it will actually look to the end user.
4. Indicate how the user interprets the state of the system from information provided through the interface.

21. Write short notes on Pattern-based design. L2 CLO2

Pattern-based design creates a new application by finding a set of proven solutions to a clearly delineated set of problems. Each problem and its solution is described by a design pattern that has been cataloged and vetted by other software engineers who have encountered the problem and implemented the solution while designing other applications. Each design pattern provides you with a proven approach to one part of the problem to be solved.

22. What is meant by Design Pattern? L1 CLO2

A design pattern can be characterized as "a three-part rule which expresses a relation between a certain context, a problem, and a solution" .For software design, context allows the reader to understand the environment in which the problem resides and what solution might be appropriate within that environment. A set of requirements, including limitations and constraints, acts as a system of forces that influences how the problem can be interpreted within its context and how the solution can be effectively applied.

23. Define the term WebApps Design. L1 CLO2

Design for WebApps encompasses technical and nontechnical activities that include: establishing the look and feel of the WebApp, creating the aesthetic layout of the user interface, defining the overall architectural structure, developing the content and functionality that reside within the architecture, and planning the navigation that occurs within the WebApp

24. How to design a Pyramid for WebApps? L2 CLO2

The creation of an effective design will typically require a diverse set of skills. Sometimes, for small projects, a single developer may need to be multi-skilled. For larger projects, it may be advisable and/or feasible to draw on the expertise of specialists: Web engineers, graphic designers, content developers, programmers, database specialists, information architects, network engineers, security experts, and testers. Drawing on these diverse skills allows the creation of a model that can be assessed for quality and improved before content and code are generated, tests are conducted, and end-users become involved in large numbers. If analysis is where WebApp quality is established, then design is where the quality is truly embedded.

25. Write about the objectives of WebApps Design. L1 CLO2

The objectives of a WebApp interface are to:
(1) establish a consistent window into the content and functionality provided by the interface,
(2) guide the user through a series of interactions with the WebApp, and
(3) Organize the navigation options and content available to the user.

26. Write short notes on Design Reuse. L2 CLO2

A more potent design reuse is becoming available after the advent of the open source paradigm and SOA. In the case of open source, the design reuse is in fact a case of copying existing design and then using it exactly as it is or modifying it to suit your needs. The full interface details are provided by the owner. Using this information, you design your own application.

27. What is meant by Concurrent Engineering in Software Design? L2 CLO2

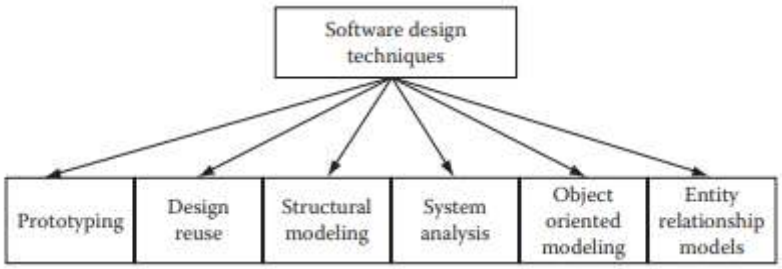
Concurrent engineering deals with taking advance information from an earlier stage for a later stage in project, so that both the stages can be performed

simultaneously. Though project activities are planned ahead in time, most often there are dependencies between a previous task and the next task in line. So, the latter task cannot start until the previous task finishes. That is why you cannot start developing an application until its design is complete. Moreover, the development will depend on the design.

28. Write Short notes on Design Life-Cycle Management. IO 2 CLO2

Software requirements go through design process steps to become a full-fledged software design. At the high level, system analysis is performed. System analysis includes a study of requirements and finding feasibility of converting them into software design. Once the feasibility is done, then the actual software design is made. The software design is in the form of activity diagrams, use cases, prototypes, etc. Once the design process is complete, these design documents are verified and validated through design reviews. Once the design is reviewed and approved, then the design phase is over.

29. Draw the diagram for Software design techniques L1 CLO2



30. Write short notes on Design Activities. L2 CLO2

Software design activities produce many intermediate documents and work products. These include product architecture description, allocated requirements, product component descriptions, product-related life-cycle process descriptions, key product characteristic descriptions, required physical characteristics and constraints, interface requirements, verification criteria used to ensure that requirements have been achieved, operating environments, modes and states for operations, support, training, manufacturing, disposal, and verifications throughout the life of the product.

12 Marks:

1. Explain how to Translating the requirements model into the design model. L2 CLO2
2. Describe the Evolution of Software Design in detail. L2 CLO2
3. Illustrated in detail how the design model can be viewed in two different dimensions. L3 CLO2
4. Explain the Characteristics of a good software design with a neat diagram. L2 CLO2
5. Describe the two methods for designing software products or components. L2 CLO2
6. Describe some of the design characteristics of the software project. L2 CLO2
7. Discuss different types of software design techniques. L2 CLO2
8. Explain the refactoring method used to refine the software product. L2 CLO2
9. What is architecture design why it is important for software design of a project L2 CLO2
10. Describe in detail about the Data-centered Architecture of a software design. L2 CLO2
11. Discuss about data flow, Main program/subprogram, and Layered architecture of a software project. L2 CLO2
12. With neat diagram explain architectural context diagram (ACD) with an example. L2 CLO2

13.	Explain the architecture trade-off analysis method (ATAM) that establishes an iterative evaluation process for software architectures.	L2	CLO2
14.	With a step by step process to transform flow characteristics to be mapped into a specific architectural style.	L3	CLO2
15.	Examine three important views of what a component is and how it is used as design modeling.	L2	CLO2
16.	Illustrate the steps to represent a typical task set for component-level design, when it is applied for an object-oriented system.	L3	CLO2
17.	Explain how to design traditional components for a software project design.	L2	CLO2
18.	What are the golden rules to form user interface design? Explain.	L2	CLO2
19.	Describe user interface design process with a neat diagram.	L2	CLO2
20.	Elaborate the different kinds of Interface analysis.	L2	CLO2
21.	Discuss User interface design steps of a software project.	L2	CLO2
22.	Explain the Interface Design Workflow for WebApps.	L3	CLO2
23.	Explain design pattern along with its characteristics and various kinds of patterns	L2	CLO2
24.	With a neat diagram explain web apps interface design.	L2	CLO2
25.	Describe the web apps architecture of a software project.	L2	CLO2