UNIT I

1.Define software component.

A software component is a system element offering a predefined service able to communicate with other components.

2.Specify the characteristics of object.

Object is a unit of instantiation, it has a unique identity. It may have state and this can be externally observable.

3.What is prototype object?

The object may be implicitly available in the form of an object that already exists. Such a preexisting object is called a prototype object.

4.What is factory object and methods?

Factory object: It can be an object of its own.
Factory methods: Methods on objects that return freshly created other objects are another variation.

5.What is Modules?

Modules do not have a concept of instantiation whereas classes do. Modules can be used and always and have been used to package multiple entities.

6.Specify the non-technical aspects that are need in interfaces.

- A component have multiple interfaces, each representing a service that the component offers.
- Redundant introductions of similar interfaces need to be minimized.
- Requires a small number of widely accepted unique naming schemes.

7.Define call back.

Callbacks are a common feature in procedural libraries that has to handle asynchronous events.

8.What is component architecture?

Component architecture is the pivotal basis of any large-scale software technology and is of utmost importance for component based systems.

9.Specify some cornerstones of a component architecture.

- Interaction between components and their environment is regulated.
- The roles of components are defined.
- Tool interfaces are standardized.

10.Specify the roles of an architecture.

- Architecture needs to create simultaneously the basis for independence and cooperation.
- An architecture defines overall invariants.
- It needs to be based on the principal considerations of overall functionality.
- It prescribes proper frameworks for all involved mechanisms.

11.What is the use of conceptual level?

It is obviously useful to introduce layers, to single out components and separate concerns.

12.Define component frame work.

A component framework is a dedicated and focused architecture usually around a few key mechanisms and a fixed set of policies for mechanisms at the component level.

13.What is resource?

A resource is a frozen collection of typed items.

14.Define middleware.

Middleware is a name for the set of software that sits between various operating systems and a higher distributed programming platform.

15.Categorize the middleware.

- Message oriented middleware (MOM)
- Object oriented middleware (OOM)

16.What is generative programming?

Generative programming aims at transformations approach to the construction of software.

17.Specify the areas used in generative approaches.

- Used to produce individual components.
- Used to enhance composed systems.

18.Specify the criteria that is used to fulfill the software definition.

- Multiple use
- Non-context specific
- Composable with other component.
- Encapsulated.

19.Specify the fundamental properties of component technology.

- If a component fails to function it must not violate system-wide rules.
- Software development processes that do not depending on testing.
- Performance of a component system is affected in non-trivial ways by the actual composition.

20.What is enhance composed systems and give example.

They need to be positioned in such a way that they do not interfere with the unit of deployment characteristic of components. Eg:JIT technique.

BIG QUESTIONS

1. Explain the fundamental properties of component technology.
   - Technological feasibility
   - Technical problems
   - Integration and testing
   - Potential o0f software components
2. Explain briefly about component architecture.
   - Role of an architecture
   - A tiered component architecture
   - Components and middleware
3. Explain briefly about callbacks with example.
   - Definition
   - Diagram
   - Explanation
   - Eg: directory service
4. Write short notes on modules, interfaces.
   - Modules
   - interfaces
5. Write short notes on objects and software components.
   - Software components
   - Objects

UNIT II

1. Define java Bean.

   Java Bean is a software component that has been designed to be reusable in a variety of different environment.

2. What are the aspects of a bean model?

   Events
   Properties
   Introspection
   Customization
   Persistence.

3. Give any 4 advantages of Java Bean.

   - A bean contain all the benefits of Java "Write once, Run anywhere" paradigm.
   - The configuration settings of a bean can be saved in persistence storage and restored at later time.
   - A bean may register to receive events from other objects and can generate events that are send to other objects.
   - Auxillary software provided to help a person configure a bean.

4. What is the use of application builder tool?

When working with java beans, most developers use an application builder tool, a utility that enables you to configure a set of Beans, connect them together and produce a working application.

5. What are major capabilities of application builder tool?

   - A palette is provided that lists all of the available beans. As additional beans are developed or purchased, they can be added to the palette.
   - A worksheet is displayed that allows the designer to layout beans in a graphical user interface. A designer may drag and drop a bean from the palette to this worksheet.

6. What is Events?

Bean instances are potential sources or listeners of specific types of events. An assembly tool can then connect listeners to sources.

7. What is multicast and unicast semantics?

   Unicast semantics $\rightarrow$ At a time, one listener is active.
   Multicast semantics $\rightarrow$ All listeners are active at any time.

8. Define property.

Property is a discrete named attribute of a bean that can affect its appearance or behavior.

9. What are the different properties that a bean can define?

- Simple
- Indered
- Bound
- Constrained.

10. Differentiate between simple and indered properties.

No	Simple	Indered
1.	It is a single value	It has multiple values
2.	It does not affect other properties.	It affects other properties.

11. Differentiate between bound and constrain properties.

No	Bound	Constrain
1.	It is a property change event.	It is a veto able change event.
2	It gives notification to all listeners and there is no property that not affect the value.	It can veto the change.

12. What is introspection and reflection?

Introspection is used to find out events methods, properties, standard interfaces and classes.

Reflection is a type of event listener. It is used to analyse software components or software.

13. Write syntax to create jar and manifest file.

For jar file.

```
Jar  cf   XYZ . jar  *  class   * . gif
```

For manifest file.

```
Jar  cfm  XYZ . jar   YXZ .mf  * .class  * . gif.
```

14. Write the syntax for following?

i) tabulating a contents of jar file.

jar tf XYZ . jar.

ii) extracting the contents from jar file.

jar xf XYZ . jar.

15. What is the use of manifest file?

a developer much provide a manifest file to indicate which of the components in a JAR file are java beans

16. Advantage of reflection.

- It allows a package (java . lang .reflect) store in class.
- It is used to get information at runtime.

17. Which method is used for serializes and deserialize an object.

The method Externalizable is used for serializes and deserialize an object.

18. Specify the method that are used in Externalizable interface.

Void read External (object input in stream)
Throws 10 Exception , class Not found Exception.
Void write External (object Output out stream)throws to Exception.

19. What is EJB?

EJB is a comprehensive technology that provides the infrastructure for building enterprise level server side distributed java components.

20. Difference between java bean and EJB.

Java bean approach to composition is connection – oriented programming where as, EJB to composition is conventional model of object oriented composition.

21. Specify the varieties of bean.

- session
    1. stateful
    2. stateless
- Entity
- Message driven.

22. What are the steps to create a simple bean.

- Create subdirectory
- Compile the source code
- Create manifest file
- Create jar file
- run

BIG QUESTIONS

1. Explain the thread lifecycle.
    - New born state
    - Running state
    - Runnable state
    - Dead state
2. Explain the different properties that a bean can define.
    - Simple
    - Indexed
    - Bound
    - constrained
3. Explain object serialization in detail.
    - Serializable
    - Externalizable
    - Object input/ stream
    - Object output/stream
    - example
4. What is EJB? Explain about EJB architecture.
    - Definition
    - Diagram
    - Varieties of bean
    - Entity relationship
5. Write a program to create c color bean by using event handling Program
6. Write briefly about distributed object models and RMI
    a. Definition
    b. Explanation
    c. Example

UNIT III

1.Why we need CORBA?

It is used to solve one fundamental problem - "how can distributed object oriented systems implemented in different languages and running on different platforms?"

2.What are the essential parts of CORBA?

- Invocation interfaces
- Object Request Broker
- Object Adapters

3.What are the requirements need to work invocation interfaces and object adapter?

- All object interfaces need to be described in a common language.
- All languages used must have bindings to the common language.

4.What is the use ORB?

An ORB is capable of loading and starting an object servant which receiving invocation requests for an object of that servant.

5.What is the responsible of object adapter?

An object adapter is responsible for telling an ORB which new object is served by Which servant.

6.Differentiate stubs and skeletons.

Stubs	Skeletons
1.Called client side proxy objects. 2. It forwards all invocations through the ORB to the real target object.	1.Called server side stubs. 2. It directly invokes the target method.

7.Specify the features of SOM.

- Metaprogramming
- Binary compatibility

8.Specfy any four services supporting enterprise distributed computing.

- Naming and Trader service.
- Event and notification service
- Object transaction service
- Security service

9.Specify any four services supporting architecture using fine-grained objects.

- Concurrency control service.
- Licensing service
- Lifecycle service
- Relationship service
- Persistent state service.

10. What is naming service?

It allows arbitrary names to be associated with an object. Names are unique within a naming context and naming contexts form a hierarchy.

11.What is push model.

In push model the event supplier calls a push method on the event channel, which reacts by calling the push method of all registered consumers.

12.What is pull model?

In pull model the consumers call the pull method of the event channel, effectively polling the channel for events.

13.What is life cycle service.

This service supports creation, copying, moving and deletion of objects and related group of objects.

14. What are the features of CCM components?

- Ports that are classified into facets, receptacles, event sources and event sinks.
- Primary keys, which are values that instances of entity components
- Attributes and configuration
- Home interfaces.

15. What is facet?

A special facet of a CCM component is the equivalent interface, which enables navigation between the different facets of a CCM component.

16.What is Receptacles?

Receptacles provide connect and disconnect operations and internally correspond to object references to other objects of appropriate type.

17. What is home interface?

The home interface is provided by a component, not its instances, and supports the creation of new instances.

18.What is the use of primary keys?

Primary keys which are values that instances of entity components provide to allow client identification of the instances.

19. What is attributes and configuration?

Configuration interfaces support initial configuration of new component instances. They are described as IDL attributes with set and get operations.

20. What is MDA?

The OMG architecture board introduced a new approach called model driven architecture. It is the base architecture for all forthcoming OMG specifications.

BIG QUESTIONS

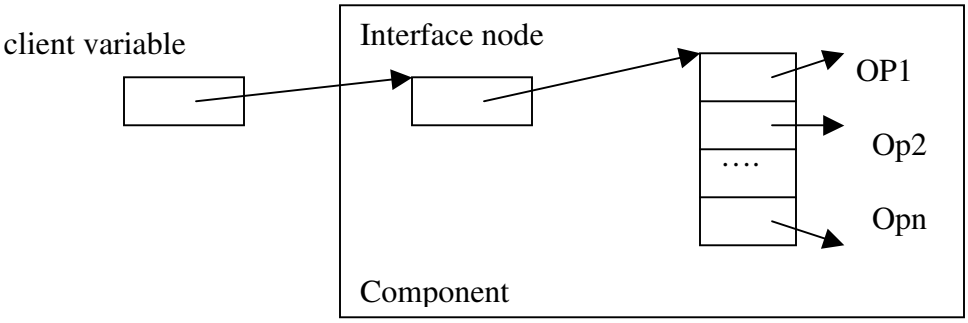
1. Explain briefly about the CORBA services.
    - Services supporting enterprise distributed computing.
    - Services supporting architecture using fine-grained objects.
2. Explain the CORBA component model.
    - Portable object adapter
    - CCM componenets
    - CCM containers
3. Write short notes on SOM and MDA
    - SOM:
        - meta programming
        - binary compactibility
    - MDA
4. Explain briefly about IDL interface.
    a. IDL interface
    b. Modules
    c. Data types
5. Explain briefly about CCM components
        - Features
        - Diagram

UNIT IV

1.Define COM.

COM is a binary standard- it specifies nothing about how a particular programming language may be bound to it. COM does not even specify what a component or an object is.

2.Draw the binary representation of a COM interface.



3. Name any two interfaces that is used in Com.

- Query interface
- Iunknown interface

4.What is the use of query interface method?

Query interface takes the name of an interface, checks if the current COM object supports it, and, if so, returns the corresponding interface reference.

5.What is the use of Iunknown interface?

The identity of the Iunknown interface can serve to identify the entire COM object with out requiring any specific functionality.

6.What are the methods supported by Iunknown interface?

- QueryInterface
- AddRef
- AddRelease

7.Specify the use of HRESULT type.

The type HRESULT is used by most COM interface methods to indicate success or failure of a call. It may also indicate network failure.

8.What is AddRef and Release.

On creation of an object or node, the reference count is initialized to 1 before handing out a first reference. Each time a copy of a reference is created the count must be incremented (AddRef). Each time a reference is given up, the count must be decremented (Release).

9.Specify the two forms of object reuse.

- Containment
- Aggregation

10.What is containment?

Containment is a form of object reuse. It is just the simple object composition technique. One object holds an exclusive reference to another. Here the call is forwarded and handled by another object.

11.What is aggregation?

Aggregation is a form of object reuse. Here instead of forwarding requests, an inner object's interface reference could be handed out directly to an outer object's client.

12.When we use aggregation?

Aggregation is only useful only where the outer object does not wish to intercept calls to perform some filtering or additional processing.

13.Compare COM and DCOM.

DCOM transparently expands the concepts and services of COM. DCOM builds on the client-side proxy objects and the server side stub objects already present in COM.

14.What is outgoing interface?

An outgoing interface is an interface that a COM object would use if it were "connected" to an object that provides this interface.

15.Name the interface that are declared with in outgoing interface.

- IConnectionPointContainer
- IconnectionPoint

16.Specify the use of IconnectionPointContainer.

Using IconnectionPointContainer , the various connection point objects of a connectable object can be found and enumerated.

17.Specify the use of IconnectionPoint.

IconnectionPoint can be used to establish , enumerate, and tear down connections.

18.How the connectable objects implement change propagation?

Connectable objects provide a uniform way to implement change propagation. As outgoing and incoming interfaces are matched, the propagation can take the form of regular method invocations instead of requiring the creation of event objects.

19.Specify the policies that are used to determine the shared assemblies in GAC.

- Publisher policy
- Application policy
- Machine policy

20.What is Appdomains?

The CLR execution engine partitions a process into one or more AppDomains. An Appdomain isolates sets of objects from all objects in other appdomain, but is more lightweight and cheaper than operating systems.

21.What is contexts?

A context is a partition of an appdomain the member objects objects of which share the properties of their contexts.

22.What is reflection?

The CLI reflection support grants full access to the type structur of loaded assemblies, including all attributes and custom attributes defined on these types.

23.What is remoting?

The CLI remoting support combines context and reflection infrastructure with flexible support for proxies, channels, and messages provide building blocks for a wide variety of communication styles and patterns.

BIG QUESTIONS

1.Explain the COM services in detail.

- Dispatch interfaces
- Connectable objects

2.Expalin in detail the compound document and OLE.

- Ole containers and servers
- Controls- ActiveX controls

3.Explain briefly about the .Net components.

- Assemblies
- Single application
- Shared application
- Policies

4.Write short notes on AppDomains, contexts, reflection, remoting

- AppDomains
  - Loading
  - Unloading
- Contexts
  - Context bound
  - Context Agile
- Reflection
- Remoting

5.Explain about dual interface and outgoing interfaces.

- Dual interface
  - Idispatch method
  - DispID
- Outgoing interface
  - IconnectionPointcontainer
  - IConnectionPoint

UNIT V

1.What is components and connectors?

Architecture Description Language (ADL) typically distinguish components and Connectors. Components are meant to provide functionality while connectors focus on connectivity.

2.What is connection oriented approach?

A pure connection oriented approach, all components are restricted to only interact with other components if connected appropriately.

3.What is the main difference between early and later ADL?

Early ADL: were restricted to static connectivity.

Later ADL: Added support for dynamic connectivity and dynamic reconfiguration.

4.What is the use of COM apartments and MTS?

COM apartments are used to separate objects by threading model. MTS contexts are used to separate objects by transactional domain.

5.What is the use of EJB container?

An EJB container allows a class to be attributed to request explicit transaction control.

6.What is CLR contexts?

The CLR context infrastructure is probably the first mainstream attempt to provide a genuinely extensible infrastructure for contextual composition.

7.What are the flavours in CLR objects?

- Value types
- Pass-by-value type
- Pass-by-reference type
- Context-bound type

8.What is context bound type?

Context bound types always reside inside a context that is equipped with appropriate properties. All other objects reside outside of any context and snd so are context agile.

9.What is black box component framework?

The black box component frame work is part of the black box component builder, a component oriented rapid development tool and component oriented programming environment.

10.Why we need blackbox?

Black box was designed to avoid a single language island syndrome by supporting the standard object model of the underlying platform, most prominently COM on windows platforms.

11.Compare OOP and COP.

OOP: addresses the fundamental aspects of programming object-oriented solutions.

COP: addresses the aspects of programming components.

12.Specify the things that are required in component programming.

- Polymorphism
- Modular encapsulation
- Late binding and loading
- Safety

13.What is caller encapsulation?

Is the area that benefits from language support is that of interface definitions. When exposing an interface on a component boundary, two different intentions may be involved.

14.Specify the two different intentions that are used in caller encapsulation.

- Component external code may need to invoke operations of the exposed interface.
- Component internal code may need to invoke operations that implement the exposed interface.

15.What are the problems of asynchrony?

- The natural form of event distribution is multicasting
- Event object recipients are themselves free to post new events
- The set of recipients could change while a multicast is in progress.

16.What is multithreading?

It is the concept of supporting multiple sequential activities concurrently over the same state space. The resulting increase in complexity over sequential programming is substantial.

17.What is the use of RAD?

Component development should use rapid application development methods to capture requirements quickly within a working component system.

18.what are the two strategies used in component testing tool?

- To avoid errors statically wherever possible
- To make sure that components are deployed in such a way that faults leave logged traces.

19.What is a rule of thumb?

A rule of thumb is that most errors that can be caught using automated runtime debugging aids could be statically avoided, had a better language been chosen for the implementation.

20.What is software assembly?

Software assembly is different from hardware assembly in that it is not necessary to assemble individual instances repeatedly- the entire assembled product can instead be cloned.

BIG QUESTIONS

1. Explain briefly about Component oriented programming
    a. Problems of asynchrony
    b. Multithreading
    c. Learning from circuit design
    d. Nutshell classes
    e. Language support
    f. Caller encapsulation
2. Explain about component framework.
    a. Component framework versus connectors
    b. Component framework versus metaprogramming
    c. Component framework versus aspect-oriented programming
3. Explain about black box and OLE.
    a. Direct-to-COM
    b. Black box componenets
    c. OLE
4. Explain the Tools that are used in Components
    a. Component design and implementation tools
    b. Component testing tools
    c. Component assembly tools
5. Explain briefly the cross-development environment'
    a. Portos IDE
    b. IDE with blackbox
    c. Cross-development tools