MC7502 – SERVICE ORIENTED ARCHITECTURE

UNIT – I : SOABASICS

Part – A

1. **What is Service Oriented Architecture?**

   Service oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. An SOA can refer to application architecture or the approach used to standardize technical architecture across the enterprise.

2. **Define Contemporary SOA.**

   Contemporary SOA represents an architecture that promotes service orientation through the use of web services.

3. **List out some characteristics of Contemporary SOA.**

   Some of the characteristics of contemporary SOA are:-

   i. Contemporary SOA is at the core of the service oriented platform.
   ii. Contemporary SOA increases quality of service.
   iii. Contemporary SOA is fundamentally autonomous.
   iv. Contemporary SOA is based on open standards.
   v. Contemporary SOA supports vendor diversity.
   vi. Contemporary SOA fosters intrinsic interoperability.
   vii. Contemporary SOA promotes discovery.
   viii. Contemporary SOA promotes federation.
   ix. Contemporary SOA promotes architectural composability.
   x. Contemporary SOA fosters inherent reusability.

4. **What are the benefits of SOA?**

   The benefits of SOA are:

   i. Improved integration and intrinsic interoperability
   ii. Inherent reuse
   iii. Streamlined architectures and solutions
   iv. Leveraging the legacy investment
   v. Establishing standardized XML data representation
   vi. Focused investment on communications infrastructure
   vii. “Best-of-breed” alternatives
   viii. Organizational agility

5. **What are the common pitfalls of adopting SOA?**

   The common pitfalls of adopting SOA are:

   i. Building service oriented architectures like traditional distributed architectures
   ii. Not standardizing SOA
   iii. Not creating a transition plan
   iv. Not starting with an XML foundation architecture
   v. Not understanding SOA performance requirements
   vi. Not understanding web services security
   vii. Not keeping in touch with product platforms and standards development

6. **What are the requirements is needed to fill QoS gaps between contemporary and Primitive SOA?**

   Contemporary SOA is striving to fill the QoS gaps of the primitive SOA model with the following requirements:

i. Security (contents and access)
ii. Reliability (message guaranteed delivery)
iii. Appropriate performance
iv. Protecting business integrity
v. Executing exception logic in case of failure

7. What is Autonomous Principle?

Autonomous Principle represents the ability of a service to carry out its logic independently of outside influences.

List out the different levels of Autonomy.

Different levels of Autonomy are:

i. Runtime autonomy
ii. Design time autonomy

8. What is Runtime Autonomy?

Runtime Autonomy represents the amount of control a service has over its execution environment at runtime.

9. What is Design time autonomy?

Design time autonomy represents the amount of governance control a service owner has over the service design.

10. Expand UDDI.

UDDI stands for Universal Description Discovery and Integration.

11. What are the design characteristics required to facilitate interoperability in contemporary SOA?

The design characteristics required to facilitate interoperability are:

i. Standardization
ii. Scalability
iii. Behavioral predictability
iv. Reliability

12. How is loose coupling concept achieved in SOA?

The loose coupling concept is achieved by implementing standardized service abstraction layers when service orientation principles are applied to both business modeling and technical design.

13. What is referred as Organizational Agility?

Organizational Agility refers to efficiency with which an organization can respond to change.

14. What is Architecture?

Architecture refers a systematic arrangement of computerized automation technological solutions.

15. What is application architecture?

Application architecture is a template for all others which specifically explained the technology, boundaries, rules, limitations, and design characteristics that apply to all solutions based on this template.

16. **What is enterprise architecture?**

Enterprise architecture is a creation of master specification when numerous, disparate and integrate application architectures exist within an organization.

17.What is Single-tier client-server architecture?

Single-tier client-server architecture is an environment in which bulky mainframe back-ends server served the thin clients.

18.List out the primary characteristics of the two tier client server architecture?

The primary characteristics of the two tier client server architectures is given below which is compared to SOA

i. Application logic
ii. Application processing
iii. Technology
iv. Security
v. Administration

19.What is multi-tier client-server architectures?

Multi-tier architecture (often referred to as n-tier architecture) is a client-server architecture in which the presentation, the application processing, and the data management are logically separate processes.

20. List out the types of communications of mainframe systems?

The different types of communications of mainframe systems are:

i. Synchronous communication
ii. Asynchronous communication

21. Define synchronous communication.

Synchronous communication allows the client and server to wait for each other to transfer the message. That is, the client will not continue until the server has received the message.

22. Define asynchronous communication.

Asynchronous communication allows the server to continuously receive messages from the client without waiting for the server to respond.

23. List out the types of service autonomy?

The different types of service autonomy are:

i. Service-level autonomy
ii. Pure autonomy

24. **What are the key benefits of service reuse?**

The key benefits of service reuse are:

i. Accommodate future requirements with less development effort
ii. Reduce the need for creating wrapper services
iii. Reduction of cost by not just avoiding duplication of code
iv. Reducing risks by reusing well-tested code and runtime environments

25. State Separation of concerns.

"Separation of concerns" is an established software engineering theory based on the idea of breaking down a large problem into a series of individual concerns.

26. What are the parts of automation logic?

The four identified parts of automation logic related to different sized units of logic as follows:

i. messages = units of communication
ii. operations = units of work
iii. services = units of processing logic (collections of units of work)
iv. processes = units of automation logic (coordinated aggregation units of work)

27. What are the issues that are raised in the client-server and the distributed Internet architecture?

The issues that are raised in the client-server and the distributed Internet architecture comparisons are discussed in a comparison between multi-tier client-server and SOA.

i. Application logic
ii. Application processing
iii. Technology
iv. Security
v. Administration

28. What is the use of RPC?

Client-server remote procedure call (RPC) connection is used for remote communication between components residing on client workstations and servers.

Part – B

1. Explain about the characteristics of SOA.
2. Compare SOA with client-server and distributed internet architectures.
3. Explain in detail about various components of SOA and how they inter – relate to each other.
4. Discuss in detail about the Common principles of service- orientation.
5. Write short notes on Anatomy of SOA.
6. Explain about Service Layers with neat diagram.

UNIT – II: XML AND WEB SERVICES

Part – A

1. **What is XML?**

XML is a set of rules for structuring, storing and transferring information. This language is used to describe the data which will be passed from one computer application to another. XML tells a computer what the actual data is, not what it should look like.

2. **What is the main disadvantage of HTML?**

The main disadvantage was that it was not designed to share information between computers, and so XML was developed to overcome this limitation.

3. **What are the uses of XML?**

- Connecting databases to the Web
- Exchanging data automatically between different computer applications
- Moving the processing from a Web server to the local PC
- Using the same information in many different ways
- Changing the presentation of information automatically for different viewing devices.

4. **What is the emergence of XML?**

- **XLINK** - a standard designed to hyperlink between XML documents

- **XML Query -** a language used to query XML documents
- **XSL -** a style sheet language for XML
- **Resource Description Framework (RDF) -** a standard for metadata. This will be similar to library cards and should make searching the Web much faster

5. **What are the major XML news formats?**

The major XMLNews formats are XMLNews-Story and XMLNews-Meta,

6. **What are markup and text in an XML document?**

XML documents mix markup and text together into a single file: the markup describes the structure of the document, while the text is the documents content

7. **Write the rules of XML element**

Elements may not overlap: an end tag must always have the same name as the most recent unmatched start tag. The following example is not well-formed XML, because "</person>" appears when the most recent unmatched start tag was "<function>":

b. <!-- WRONG! -->

c. **<function><person>**President**</function>** Habibe<**/person>**

8. **What are the revolutions of XML?**

- Data Revolution
- Architectural Revolution
- Software Revolution

9. **Write on Tags and elements?**

XML tags begin with the less-than character ("<") and end with the greater-than character (">"). You use tags to mark the start and end of **elements**, which are the logical units of information in an XML document, an element consists of a **start tag**, possibly followed by text and other complete elements, followed by an **end tag**.

10. **What are attribute name and attribute value?**

Every attribute assignment consists of two parts: the **attribute name** (for example, *href*), and the **attribute value** (for example, http://www.yahoo.com/). There are a few rules to remember about XML attributes:

1. Attribute names in XML (unlike HTML) are case sensitive: *HREF* and *href* refer to two different XML attributes.

2. You may not provide two values for the same attribute in the same start tag. The following example is not well-formed because the *b* attribute is specified twice:

11. **What are the various features of XML?**

- Security
- Portability
- Scalability
- Reliability

12. **List out the advantages of XML.**

- XML files are human - readable
- Widespread industry support
- Relational Databases
- XML support technologies
- More meaningful searches
- Development of flexible web applications
- Data integration from disparate sources

- Local computation and manipulation of data
- Multiple views of the data
- Granular updates

13. **List out the XML structure.**

- Physical structure
- Logical structure

14. **What is physical structure?**

The physical structure consists of the contents used in an XML document. It holds the actual data to be represented in an XML document. This actual data storage can be called as Entities. These entities are identified by a unique name and may be part of the XML document or external to the document.

An entity is declared in the XML declaration part and referenced in the document element. Once declared in the DTD, an entity can be used anywhere.

15. **List out the Physical structure.**

- Parsed Entity
- Unparsed Entity
- Entity Reference
- Predefines Entities
- Internal and External Entities
- XML Syntax
- Attributes

16. **What is XML declaration?**

It identifies the version of the XML specification to which the document conforms.

Example:

<?xml version=”1.0”?>

An XML declaration can also include an

- Encoding Declaration
- Stand-alone Document Declaration

17. **Define Elements**

Element are the primary means for describing data in XML. The rules for composing elements are

- Flexible
- Allowing different combinations of text content, attributes and other elements.

18. **Define Name space**

An **XML namespace** is identified by a URI reference; element and attribute names may be placed in an XML namespace using the mechanisms described in this specification.

19. **Define expanded name space**

E**xpanded name** is a pair consisting of a namespace name and a local name. Definition: For a name *N* in a namespace identified by a URI *I*, the **namespace name** is *I*. For a name *N* that is not in a namespace, the **namespace name** has no value. Definition: In either case the **local name** is *N*. It is this combination of the universally managed IRI namespace with the vocabulary\'s local names that is effective in avoiding name clashes.

20. **Define qualified name space**

A **qualified name** is a name subject to namespace interpretation.] In documents conforming to this specification, element and attribute names appear as qualified names. Syntactically, they are either prefixed names or unprefixed names

21. **Define Name space prefix**

If the attribute name matches Prefixed Att Name, then the NC Name gives the namespace prefix, used to associate element and attribute names with the namespace name in the attribute value in the scope of the element to which the declaration is attached. In such declarations, the name space name may not be empty.

22. **Write on Declaring name space**

A namespace (or more precisely, a namespace binding) is **declared** using a family of reserved attributes. Such an attribute\'s name must either be **xmlns** or begin **xmlns:**. These attributes, like any other XML attributes, may be provided directly or by default.

23. **Define XML schema**

An XML Schema consists of components such as type definitions and element declarations. These can be used to assess the validity of well-formed element and attribute information items (as defined in [XML-Infoset]), and furthermore may specify augmentations to those items and their descendants.

24. **Define Schema component**

Schema component is the generic term for the building blocks that comprise the abstract data model of the schema. [Definition:] An **XML Schema** is a set of ·schema components·.

25. **What is mean by Uniform Resource Identifier (URI)?**

A **Uniform Resource Identifier** (URI) is a string of characters which identifies an Internet Resource. The most common URI is the **Uniform Resource Locator** (URL) which identifies an Internet domain address. Another, not so common type of URI is the **Universal Resource Name** (URN). In our examples we will only use URLs.

26. **Define Web services**

A Web service (also Web Service, Web service) is defined by the W3C as a software system designed to support interoperable machine-to-machine .Web Services is the umbrella term of group of loosely related Web-based resources and components that may be used by other Web.

27. **List out the characteristic of Web services.**

- XML based everywhere
- Message-based
- Programming language independent
- Could be dynamically located
- Could be dynamically assembled or aggregated
- Accessed over the internet
- Loosely coupled
- Based on industry standards
- Are platform neutral
- Are accessible in a standard way
- Are accessible in an interoperable way
- Use simple and ubiquitous plumbing
- Are relatively cheap
- Simplify enterprise integration

28. **What are the uses of Web services?**

- Interoperable – Connect across heterogeneous networks using ubiquitous web-based standards
- Economical – Recycle components, no installation and tight integration of software
- Automatic – No human intervention required even for highly complex transactions

- Accessible – Legacy assets & internal apps are exposed and accessible on the web.
- Available – Services on any device, anywhere, anytime
- Scalable – No limits on scope of applications and amount of heterogeneous applications

29. **What are the three roles of Web service?**

The three role of web service are

- Client
- Service
- Broker.

30. **Define client**

A client is any computer that accesses functions from one or more other computing nodes on the network. Typical clients include desktop computers, Web browsers, Java applets, and mobile devices. A client process makes a request for a computing service and receives results for that request.

31. **Define Service**

A service is a computing process that receives and responds to requests and returns a set of results.

32. **Define Broker**

A broker is essentially a service metadata portal for registering and discovering services. Any network client can search the portal for an appropriate service.

33. **What are the standard protocols used in web service?**

The standard protocols used in web service

- WSDL
- UDDL

Part – B

1. Explain in detail the architectural revolution and software revolution of XML.
2. Write short notes on web services and the role of XML in an enterprise.
3. Explain the web service architecture with a neat diagram.
4. Write about working with elements in XML.
5. Explain about XSL transformation with neat diagram.

UNIT – II: WSDL, SOAP and UDDI

Part – A

1. **Define WSDL**

WSDL stands for Web Services Description Language. WSDL is a document written in XML. The document describes a Web service. It specifies the location of the service and the operations (or methods) the service exposes.

2. **Define UDDI**

Universal Description, Discovery and Integration (UDDI) is a platform-independent, XML- based registry for businesses worldwide to list themselves on the Internet. UDDI is an open industry initiative, sponsored by OASIS, enabling businesses to publish service listings and discover each other and define how the services or software applications interact over the Internet.

3. **What are the three components used in UDDI?**

- White Pages — address, contact, and known identifiers;
- Yellow Pages — industrial categorizations based on standard taxonomies
- Green Pages — technical information about services exposed by the business

4. **List out the UDDI register**

- Public Register
- Private Register

5. **What are the major elements used in WSDL?**

The major elements used in WSDL are

- PortType
- Message
- Types
- Binding

6. **Define the structure of WSDL**

```
<definitions>
<types>
definition of types........
</types>
<message>
definition of a message....
</message>
<portType>
definition of a port.......
</portType>
<binding>
definition of a binding....
</binding>
</definitions>
```

7. **Define WSDL Messages**

The <message> element defines the data elements of an operation. Each message can consist of one or more parts. The parts can be compared to the parameters of a function call in a traditional programming language.

8. **Define WSDL Type**

The <types> element defines the data type that are used by the web service. For maximum platform neutrality, WSDL uses XML Schema syntax to define data types.

9. **Define WSDL Binding**

The <binding> element defines the message format and protocol details for each port.

10. **What is SOAP?**

SOAP, to put it simply, allows Java objects and COM objects to talk to each other in a distributed, decentralized, Web-based environment. More generally, SOAP allows objects (or code) of any kind -- on any platform, in any language -- to cross-communicate. At present, SOAP has been implemented in over 60 languages on over 20 platforms.

11. **Write on SOAP-RPC**

SOAP messages are fundamentally one-way transmissions from a sender to a receiver, but SOAP messages are often combined to implement request/response mechanisms. To do

RPC using SOAP, a few conventions must be followed. First of all, request and response messages must be encoded as structures.

12. **Write on SOAP message format**

SOAP does all this in the context of a standardized message format. The primary part of this message has a MIME type of \"text/xml\" and contains the SOAP envelope. This envelope is an XML document. The envelope contains a header (optional) and a body (mandatory). The body part of the envelope is always intended for the final recipient of the message, while the header entries may target the nodes that perform intermediate processing. Attachments, binary or otherwise, may be appended to the body.

13. **How are input parameters is handled?**

Input parameters are handled in the following ways:

- If a SOAP method requires an input parameter, and this parameter is not included in the SOAP request, no value is passed to the called stored procedure. The default action defined in the stored procedure occurs.
- If a SOAP method requires an input parameter, and this parameter is included in the request but no value is assigned to it, the parameter is passed to the stored procedure with an empty string as its value. Note that it is not NULL.
- If a SOAP operation requires an input parameter and if you want to send a NULL value for this parameter, you must set an xsi:nil attribute to \"true\" in the SOAP request

14. **What is SqlDbType?**

Specifies the parameter type. For a list of SQL Server 2005 system data types you can specify i.e. Data Type Mappings in Native XML Web Services.

15. **What is precision?**

Attribute of <SqlParameter> element provides the precision of the parameter value. Default value is 18.

16. **What is the need for intermediaries?**

SOAP intermediaries are applications that can process parts of a SOAP message as it travels from its origination point to its final destination point. The route taken by a SOAP message, including all intermediaries it passes through, is called the SOAP *message path.*

17. **Write the rules for intermediaries?**

By default, all headers targeted at a particular intermediary are removed from the message when it\'s forwarded on to the next node. This is because the specification tells us that the contract implied by a given header is between the sender of that header and the first node satisfying the role at which it\'s targeted. Headers that are not targeted at a particular intermediary should, in general, be forwarded through untouched.

18. **What is fault string?**

This element is used to pass to the caller a descriptive human-readable error. It must be present in a SOAP Fault element and should provide at least some information explaining the nature of the fault. This could contain the high-level error message used to determine generally what went wrong.

19. **What are fault codes?**

The namespace identifier for these *fault code* values is http://schemas.xmlsoap.org/soap/envelope/. The following *SOAP* fault codes can be used:

· Version mismatch

- Must understand
- Client
- Server

20. **What is fault actor?**

This element is used to provide information about who caused the fault and usually contains the Uniform Resource Identifier (*URI*) of the perpetrator. This is similar to the SOAP actor attribute in the SOAP body. Applications that do not act as the ultimate destination of the SOAP message must include this element. The final destination of a message may use this element to indicate that it alone threw the exception.

21. **What are the SOAP faults Builder exception class?**

- Using System Collections
- Using System Component Model
- Using System Data
- Using System Diagnostics
- Using System. Web

22. **Define SOAP-RPC**

SOAP messages are fundamentally one-way transmissions from a sender to a receiver, but SOAP messages are often combined to implement request/response mechanisms. To do RPC using SOAP, a few conventions must be followed. First of all, request and response messages must be encoded as structures. For each input parameter of an operation, there must be an element (or member of the input structure) with the same name as the parameter. And for every output parameter, there must be an element (or member of the output structure) with a matching name.

23. **List out the parts of SOAP?**

- SOAP ENVELOPE
- SOAP ENCODING RULES
- SOAP RPC

24. **What are the features of SOAP?**

- PROTOCOL independence
- LANGUAGE independence
- PLATFORM AND OS independence.

25. **List out the type of encoding?**

- SIMPLE ENCODINGS & COMPOUND ENCODINGS

26. **What is SOAP header element?**

The optional SOAP Header element contains application specific information (like authentication, payment, etc) about the SOAP message. If the Header element is present, it must be the first child element of the Envelope element.

27. **Define actor Attribute.**

A SOAP message may travel from a sender to a receiver by passing different endpoints along the message path. Not all parts of the SOAP message may be intended for the ultimate endpoint of the SOAP message but, instead, may be intended for one or more of the endpoints on the message path.

28. **Define SOAP body Element?**

SOAP Body element contains the actual SOAP message intended for the ultimate endpoint of the message. Immediate child elements of the SOAP Body element may be namespace-qualified. SOAP defines one element inside the Body element in the default namespace (\"http://www.w3.org/2001/12/soap-envelope\"). This is the SOAP Fault element, which is used to indicate error messages.

List out the Sub element of SOAP fault

- Faultcode
- Faultstring
- Faultactor
- Detail

29. **Define SOAP Attachment**

SOAP with Attachments (SwA) or MIME for Web Services refers to the method of using Web Services to send and receive files using a combination of SOAP and MIME, primarily over HTTP.

30. **Define the important syntax rules.**

- A SOAP message MUST be encoded using XML
- A SOAP message MUST use the SOAP Envelope namespace
- A SOAP message MUST use the SOAP Encoding namespace
- A SOAP message must NOT contain a DTD reference
- A SOAP message must NOT contain XML Processing Instructions

31. **Define SOAP Envelop Element.**

SOAP Envelope element is the root element of a SOAP message. It defines the XML document as a SOAP message.

32. **List out the values of faultcode.**

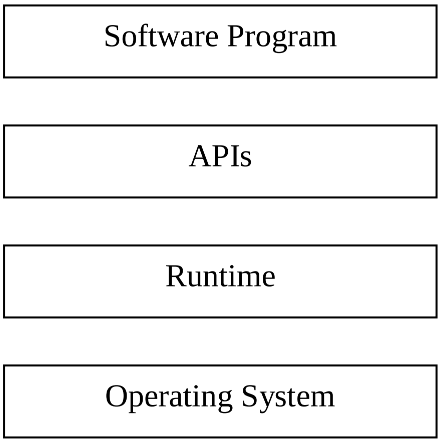
- Error
- VersionMismatch
- MustUnderstand
- Client
- Server

Part – B

1. Explain about WSDL and its Manipulation with example.
2. Explain the structure of WSDL.
3. Describe the SOAP protocol and message structure briefly.
4. Explain about SOAP actors and architectural design patterns.
5. Explain about XML-RPC with SOAP.
6. Explain about the organization of UDDI.

UNIT – IV: SOA in J2EE and .NET
Part – A

1. **Draw the fundamental software technology architecture layers.**



Software Program

APIs

Runtime

Operating System

2. **Give the architecture components of J2EE to SOA.**

    i. Java Server Pages (JSPs)
    ii. Struts
    iii. Java Servlets
    iv. Enterprise JavaBeans (EJBs)

3. **What is JAX-WS?**

    JAX-WS is a technology for building web services using XML. In JAX-WS, a web service operation invocation is represented by an XML-based protocol such as SOAP.

4. **Expand SEI.**

    SEI stands for

    - Service Endpoint Interface or
    - Service Endpoint Implementation

5. **What is SEI?**

    SEI is a java interface or class that declares the methods that a client can invoke on the service.

6. **Expand JAXB and JAXR.**

    JAXB stands for Java Architecture for XML Binding (JAXB)
    JAXR stands for Java API for XML Registries (JAXR)

7. **What is JAXB?**

    Java Architecture for XML binding API (JAXB) provides a means of generating Java classes from XSD schemas and further abstracting XML-level development.

8. **Give the general steps to use the JAXB API.**

    The general steps to use the JAXB API are:

    i. Bind the schema
    ii. Unmarshal
    iii. Marshal

9. **What are the steps needed to bind the schema?**

    Step 1: Generate classes
    Step 2: Compile classes

10. What are the steps needed to unmarshal the schema?

Step 1: Generate content tree
Step 2: Validate (optional)
Step 3: Process the content

11. Write down the advantages of JAXB.

It simplifies access to an XML document form a Java program.
It uses memory efficiently.
It is flexible.
It allows transportation from one XML document to another.

12. What is JAXR?

The Java API for XML Registries (JAXR) provides a uniform and standard Java API for accessing various kinds of XML registries.

13. What are the components of JAXR?

i. JAXR client
ii. JAXR provider
iii.

14. **Write down the packages that are implemented by JAXR.**

i. javax.xml.registry
ii. javax.xml.registry.infomodel

15. **What are the tasks involved in managing registry data?**

i. Getting authorization from the registry
ii. Creating an organization
iii. Adding classifications
iv. Adding services and service binding to an organization
v. Publishing a specification concept
vi. Removing data from the registry

16. **Expand JAX-RPC and WSIT.**

JAX-RPC stands for Java API for XML based RPC.
WSIT stands for Web Services Interoperability Technologies.

17. **What is the use of JAX-RPC?**

JAX-RPC is used for building and deploying SOAP+WSDL web services clients and endpoints. It enables clients to invoke web services developed across heterogeneous platforms.

18. **What are the benefits of JAX-RPC?**

i. Portable and interoperable web services
ii. Ease of development of web service endpoints and clients
iii. Increased developer productivity
iv. Support for open standards: XML, SOAP, WSDL
v. Standard API developed under Java Community Process (JCP)
vi. Support for tools
vii. RPC programming model with support for attachments
viii. Support for SOAP message processing model and extensions
ix. Secure web services
x. Extensible type mapping

19. **Expand CLS and CLR.**

CLS – Common Language Specification
CLR – Common Language Runtime

20. What is Common Language Runtime?

The Common Language Runtime (CLR) is an execution environment. It works as a layer between operating systems and te applications written in .net languages that conforms to the Common Language Specification (CLS).

21. **Give the features of CLR.**

- Manages memory
  - Allocation of memory
  - De-allocation of memory (garbage collection)
- Thread execution support
- Code execution
- Code safety verification
  Compilation

22. **What are three types of controls in asp.net?**

i. HTML controls
ii. HTML Server controls
iii. Web Server controls

Part – B

1. Explain in detail about SOA Platform Layers.
2. Discuss in detail about SOA Support with J2EE and its API's.
3. Discuss in detail about SOA Support with .NET.
4. Write short notes on Java architecture for XML binding.
5. Explain about JAXR and JAX-RPC with example.

UNIT – V: CLOUD COMPUTING

Part – A

1. Define Cloud computing with example.

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. For example, Google hosts a cloud that consists of both smallish PCs and larger servers. Google's cloud is a private one (that is, Google owns it) that is publicly accessible (by Google's users).

2. What are the properties of Cloud Computing?

There are six key properties of cloud computing:

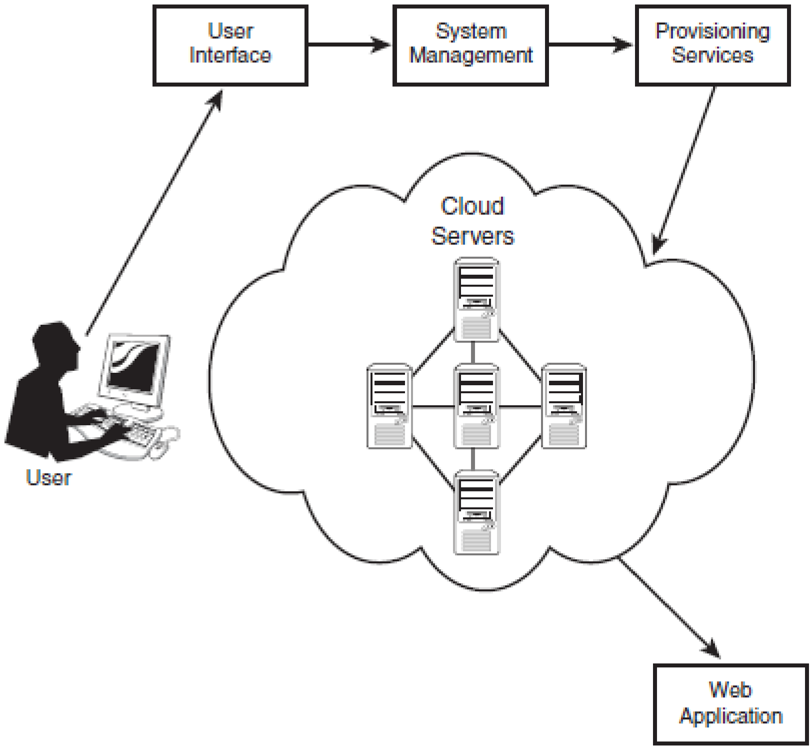
Cloud computing is

- user-centric
- task-centric
- powerful
- accessible
- intelligent
- programmable

3. What is the working principle of Cloud Computing?

The cloud is a collection of computers and servers that are publicly accessible via the Internet. This hardware is typically owned and operated by a third party on a consolidated basis in one or more data center locations. The machines can run any combination of operating systems.

4. Draw the architecture of Cloud.



5. Define Cloud services with example.

Any web-based application or service offered via cloud computing is called a cloud service. Cloud services can include anything from calendar and contact applications to word processing and presentations.

6. What are the types of Cloud service development?

- Software as a Service
- Platform as a Service
- Web Services
- On-Demand Computing

7. List the companies who offer cloud service development?

- Amazon
- Google App Engine
- IBM
- Salesforce.com

8. What is precloud computing?

Precloud computing is an email access was via a single computer, which also stores all email messages. Example: Microsoft Outlook or Outlook Express.

9. Why is Cloud computing important?

There are many implications of cloud technology, for both developers and end users. For developers, cloud computing provides increased amounts of storage and processing power to run the applications they develop. Cloud computing also enables new ways to access information, process and analyze data, and connect people and resources from any location anywhere in the world. For users, documents hosted in the cloud always exist, no matter what happens to the user’s machine. Users from around the world can collaborate on the same documents, applications, and projects, in real time. And cloud computing does all this at lower costs, because the cloud enables more efficient sharing of resources than does traditional network computing.

10. What are the advantages and disadvantages of Cloud Computing?

Advantages

- Lower-Cost Computers for Users
- Improved Performance
- Lower IT Infrastructure Costs
- Fewer Maintenance Issues
- Lower Software Costs
- Instant Software Updates
- Increased Computing Power
- Unlimited Storage Capacity
- Increased Data Safety
- Improved Compatibility Between Operating Systems
- Improved Document Format Compatibility
- Easier Group Collaboration
- Universal Access to Documents
- Latest Version Availability
- Removes the Tether to Specific Devices

Disadvantages

- Requires a Constant Internet Connection
- Doesn’t Work Well with Low-Speed Connections
- Can Be Slow
- Features Might Be Limited
- Stored Data Might Not Be Secure
- If the Cloud Loses Your Data, You’re Screwed

11. Who get benefits from Cloud Computing?

- Collaborators
- Road Warriors
- Cost-Conscious Users
- Cost-Conscious IT Departments
- Users with Increasing Needs

12. How to manage the web based projects?

Many project management applications include additional functions useful in the management of group projects. These features may include group to-do lists, web-based file sharing, message boards, time and cost tracking.

13. What is virtual community.

A virtual community is an assemblage of like-minded individuals, from anywhere in the world, online. Pre-Internet, virtual communities sprung up on online bulletin board systems

(BBSs) that were accessible via dial-up connections. The biggest home of virtual communities today is Facebook, MySpace, and other social networking sites.

Part – B

1. Explain the characteristics and types virtualization in cloud computing.
2. Explain about cloud computing architecture with neat diagram.
3. Discuss about types of clouds in cloud computing.
4. Explain the NIST reference architecture of cloud computing in detail.
5. Explain cloud deployment models and the different layers of cloud computing.