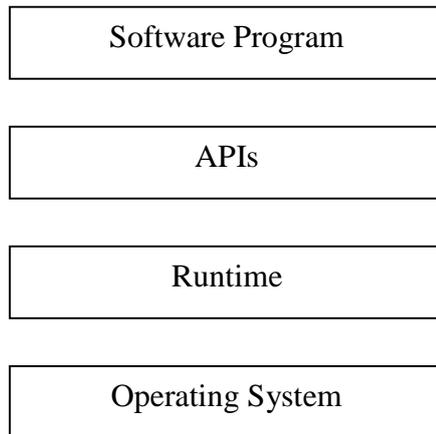


1. Draw the fundamental software technology architecture layers.**2. Give the architecture components of J2EE to SOA.**

- 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 from 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

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. Distinguish between WS-I and WSIT.

	WSIT (Web Services Interoperability Technology)	WS-I
Goal	An open source product-quality implementation of key enterprise web services technologies commonly known as WS-*	An industry organization to promote web services interoperability across platforms, operating systems and programming languages
Focus	Interoperability between Metro (where WSIT is a key component) and Microsoft .NET 3.0 framework	Vendor-neutral and produce profiles that contains clarifications on existing specifications to promote interoperability

20. Expand CLS and CLR.

CLS – Common Language Specification

CLR – Common Language Runtime

21. What is Common Language Runtime?

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

22. 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

23. What are three types of controls in asp.net?

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

24. Give the benefits of WSE.

- i. To build a wide range of application and infrastructure
- ii. Flexibility
- iii. Allows quick implementation
- iv. Code the low-level XML details

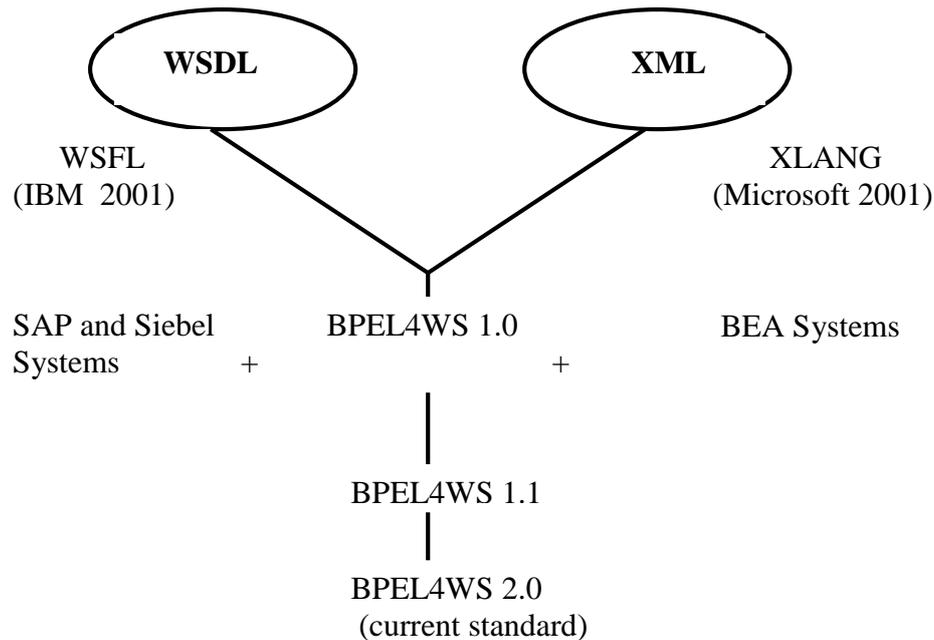
25. Expand WS-BPEL.

WS-BPEL stands for Web Services Business Process Execution Language.

26. What is WS-BPEL?

WS-BPEL is an XML based language (ie., it is described by a grammar) enabling users to describe business process activities as Web Services and define how they can be connected to accomplish specific tasks.

27. Draw the WS-BPEL family tree.



28. What is the process element?

An <process> element is the root element and must have a name attribute for assigning the name value. It is used to establish the process definition related namespace.

29. What does the partnerLink element define?

The partnerLink define the services that are orchestrated by the process. It contain a set of <partnerLink> element each represent the communication exchange between two partners ie., the process service being one partner and another service being the other.

30. What are the attributes in the partnerLink element?

- i. myRole
- ii. partnerRole

31. What is the use of myRole attribute?

- i. Used when the process service is invoked by a partner client service.
- ii. Process service acts as the service provider.

32. What is the use of partnerRole attribute?

- i. Identifies the partner service that the process service will be invoking
- ii. Partner service acts as the service provider

33. What is the use of the partnerLinkType elements?

The partnerLinkType elements are used to identify the WSDL portType elements referenced by the partnerLink elements within the process definition.

34. What does the variables element hold?

Variables hold the data that constitute the state of a BPEL business process during runtime.

35. List out the attributes of the variables element?

- MessageType
- Element
- Type

36. What is the getVariableProperty function?

The getVariableProperty function allows global property values to be retrieved from variables. It simply accepts the variable and property names as input and returns the requested value.

37. Give the syntax for getVariableProperty function.

Syntax:

```
getVariableProperty(variable name, property name)
```

Example

```
getVariableProperty("TicketApproval","class")
```

38. What is the getVariableData function?

The getVariableData function has a mandatory variable name parameter and two optional arguments that can be used to specify a part of the variable data.

39. Give the syntax for getVariableData function.

Syntax

```
getVariableData(variable name, part name, location path)
```

Example

```
getVariableData("input","payload","/tns:TimesheetType/Hours/...")
```

40. What is the use of invoke element?

The <invoke> activity is used to invoke the web service operations provided by partners.

41. What are five common attributes equipped with invoke element?

- i. partnerLink
- ii. portType
- iii. operation
- iv. inputVariable

v. outputVariable

42. What is the use of the receive element?

A <receive> activity is used to receive requests in a BPEL business process to provide services to its partners. The process block until the message is received.

43. What is the reply element?

A <reply> activity is used to send a response to a request previously accepted through a <receive> activity. Responses are used for synchronous request / reply interactions.

44. What is the purpose of the sequence element?

The sequence construct is to organize a series of activities so that they are executed in a predefined, sequential order, nesting of sequence is allowed.

45. Give the structure of sequence element.

```
<sequence>
  <receive>.....</receive>
  <assign>.....</assign>
  <invoke>.....</invoke>
  <reply>.....</reply>
</sequence>
```

46. Write down the syntax for switch case and otherwise element.

```
<switch>
  <case condition= "getVariableData('EmployeeResponseMessage',
    'ResponseParameter')=0">
    .....
  </case>
  <otherwise>
    .....
  </otherwise>
</switch>
```

47. What is the use of assign activity?

The <assign> activity is used to:

- Copy data from one variable to another
- Construct and insert new data using expressions and literal values
- Copy partner link endpoint references

48. Define fault handlers.

Fault handlers are used to react to faults that occur while the business process activities are executing. The faultHandlers construct contain multiple catch element and a catchAll child constructs.

49. What is catch element?

The <catch> activity is used to specify faults that are to be caught and handled. At least one <catch> activity needs to be specified.

50. What is CatchAll element?

The <catchAll> activity is used to catch all faults. It is optional.

51. Give the overview of WS-Coordination.

WS-Coordination is a framework for coordinating distributed activities

- Coordinator
 - Activation service for creating coordination instance
 - Registration service for registering participating application
 - Additional protocol specific service
- Set of coordination protocols

52. What is the use of CoordinationContext element?

The CoordinationContext is used to carry information about active coordination to participants

- Information inside context is coordination protocol specific
- Context format is not mandated by the standard
- Typically passed is SOAP headers

53. What is WS-choreography?

Web service choreography (WS-Choreography) is a XML based business process modeling language that describes collaboration protocols of cooperating web service participants, in which services act as peers, and interactions may be long lived and stateful.

54. How will you define the participant in WS-Choreography?

```
<participantType name="Buyer">
  <description type="documentation">
    Buyer Participant
  </description>
  <roleType typeRef="tns:BuyerRole"/>
</participantType>
```

55. How will you declare the relationship between the roles in WS-Choreography?

```
<relationshipType name="ncname">
  <role type="qname" behavior="list of ncname"?/>
  <role type="qname" behavior="list of ncname"?/>
</relationshipType>
```

56. What are channels?

Channels are the principle mechanism used to realize an interaction. A channel is named, described, and then related to the roles that realize its behavioral interface. A reference is provided to a service. The channel type will have the capability to derive its identity when in use.

57. What is WS-Policy?

WS-Policy defines a framework for allowing web services to express their constraints and requirements in relation to security, processing, or message content.

58. What is the goal of WS-Policy?

WS-Policy provides the mechanisms needed to enable web services application to specify policies.

59. Give the specifications of WS-Policy framework.

The WS-Policy framework is comprised of the following three specifications:

- WS-Policy
- WS-PolicyAssertions
- WS-PolicyAttachments

60. What is WS-Security?

WS-Security is known as Web Services Security is a flexible extensible framework to SOAP to apply security to web services.

61. Why is WS-Security needed?

The WS-Security is used to implement

- Message-level security measures
 - Protect message contents during transport and during processing by service intermediaries.
- Authentication and authorization control
 - Protect service provides from malicious requestors.

62. Give the specifications of WS-Security framework.

The WS-Security framework is comprised of the following specifications:

- WS-Security
- XML-Encryption
- XML-Signature

63. Give the syntax of WS-Security element.

```
<Envelope>  
  <Header>
```

```

.....
    <wsse:Security actor="..." mustUnderstand="...">
        .....
    </wsse:Security>
</Header>
<Body>
    .....
</Body>
</Envelope>

```

64. What does XML-Signature elements provide?

The XML-Signature elements provides message integrity and authentication information about the originator of the message.

65. Give the basic structure of the XML signature.

```

<Signature>
  <SignedInfo>
    <CanonicalizationMethod />
    <SignatureMethod />
    <Reference>
      <Transforms>
      <DigestMethod>
      <DigestValue>
    </Reference>
    <Reference />
  </SignedInfo>
  <SignatureValue />
  <KeyInfo />
  <Object />
</Signature>

```