Web Services

ELECTIVE COURSE I
1.2 WEB SERVICES

OBJECTIVES:

On successful completion of this subject student should have: understood how to build real world application using web services

Unit I

Introduction-What are web services-SOAP-WSDL-UDDI Basic web services standards, technologies and concepts: XML fundamentals: Documents-Namespaces-schema-processing XML-Simple API for XML(SAX)-Document object model(DOM)

Unit II

SOAP and WSDL: The SOAP model-SOAP messages-SOAP encoding WSDL: Structure-The types element-Managing WSDL descriptions-Using SOAP and WSDL Service implementation and Invoking web services.

Unit III

UDDI: Introduction- UDDI specification - UDDI and lifecycle management. Conversation: Overview-web services conversation language-WSCL interface components.

Workflow: Business process management-workflows and workflow management. Quality of Service: What is QoS- Why is QoS important for web services- QoS metrics for web services-QoS enabled web services. Mobile and Wireless mobile services- challenges with mobile.

Unit IV XML and HTML:

The limits of HTML-The scope of HTML-Structure-Structure and Content-Structure and Synthesis-Structure and Presentation-Representing Structure. **The XML Language**: Markup languages-Defining Markup Languages in XML.

Unit V Linking in XML:

Links(Information, Resources, and Hot Spots)-Link Management-Working with names-Choosing the linking methodology. **XML Style**: The publishing Process-At which stage do I structure my data?-Where do I process from one stage to the next?-When do I Convert?-Publishing data-Choosing a Client-side processing Application-Choosing a Server-side processing application

Text Book:

1. Sandeep Chatterjee, James Webber, "Developing Enterprise Web Services: An Architects Guide", Prentice Hall, Nov 2003

Unit I : Chapter 1 (Pg. 1 to 8), Chapter 2 (Pg.19 to 64)

Unit II : Chapter 3 (Pg. 71 to 86 , 100 – 119)

Unit III: Chapter 4 (Pg. 121 to 122, Pg. 129 , Pg. 137 to 139), Chapter 5 (Pg. 147-166)

Chapter 6 (Pg. 177-183), Chapter 9 (Pg. 345– 350, Pg. 364-368) Chapter 10 (Pg. 377-387)

2. Rick Darnell "HTML 4 Unleashed" Techmedia Publication, Second Edition,

Unit IV: Chapter 27 (Pg. 564-591), Chapter 28 (Pg. 606-626)

Unit V: Chapter 29 (Pg. 636-649,660-661) Chapter 30 (Pg. 664-670,672-674)

WEB SERVICES

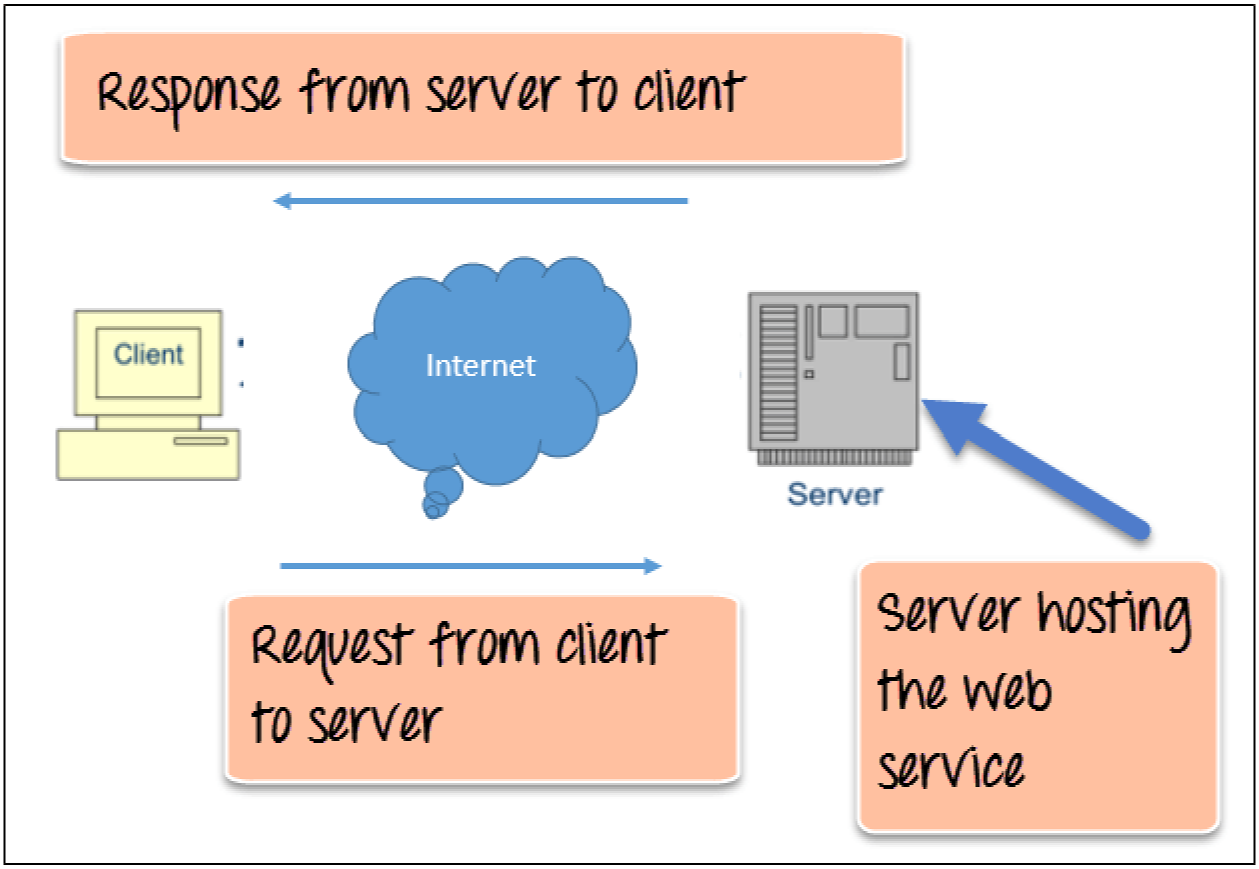
UNIT 1:

2 MARKS:

1.What are web services?

Web service is a standardized medium to propagate communication between the client and server applications on the World Wide Web.A web service is a software module which is designed to perform a certain set of tasks.The web services can be searched for over the network and can also be invoked accordingly.When invoked the web service would be able to provide functionality to the client which invokes that web service.

2.Draw the architecture of web services?


Response from server to client
Client
Internet
Server
Request from client to server
Server hosting the web service

Web Service Architecture Diagram

The above diagram shows a very simplistic view of how a web service would actually work. The client would invoke a series of web service calls via requests to a server which would host the actual web service.

These requests are made through what is known as remote procedure calls. Remote Procedure Calls(RPC) are calls made to methods which are hosted by the relevant web service.

3. What are the components of web services?

Components of Web Services

SOAP (Simple Object Access Protocol)
UDDI (Universal Description, Discovery and Integration)
WSDL (**Web Services** Description Language)

4. Define SOAP Web Services?

SOAP is an XML-based protocol for exchanging information between computers.

- SOAP is a communication protocol.
- SOAP is for communication between applications.
- SOAP is a format for sending messages.
- SOAP is designed to communicate via Internet.
- SOAP is platform independent.
- SOAP is language independent.
- SOAP is simple and extensible.
- SOAP allows you to get around firewalls.
- SOAP will be developed as a W3C standard.

5. Define WSDL Web Services?

WSDL is an XML-based language for describing web services and how to access them.

- WSDL stands for Web Services Description Language.
- WSDL was developed jointly by Microsoft and IBM.
- WSDL is an XML based protocol for information exchange in decentralized and distributed environments.
- WSDL is the standard format for describing a web service.
- WSDL definition describes how to access a web service and what operations it will perform.

- WSDL is a language for describing how to interface with XML-based services.
- WSDL is an integral part of UDDI, an XML-based worldwide business registry.
- WSDL is the language that UDDI uses.
- WSDL is pronounced as 'wiz-dull' and spelled out as 'W-S-D-L'.

6. Define UDDI Web Services?.

UDDI is an XML-based standard for describing, publishing, and finding web services.

- UDDI stands for Universal Description, Discovery, and Integration.
- UDDI is a specification for a distributed registry of web services.
- UDDI is platform independent, open framework.
- UDDI can communicate via SOAP, CORBA, and Java RMI Protocol.
- UDDI uses WSDL to describe interfaces to web services.
- UDDI is seen with SOAP and WSDL as one of the three foundation standards of web services.

.

7. Define XML Document?

An **XML document** is a basic unit of **XML** information composed of elements and other markup in an orderly package. An **XML document** can contains wide variety of data. For example, database of numbers, numbers representing molecular structure or a mathematical equation.

8. What the Example for XML Document?

```
<?xml version = "1.0"?>
<contact-info>
   <name>Tanmay Patil</name>
   <company>TutorialsPoint</company>
   <phone>(011) 123-4567</phone>
</contact-info>
```

9. What are the Rules for XML Document?

All XML Must Have a Root Element

All Tags Must Be Properly Nested

Tag Names Have Strict Limits

Tag Names Are Case Sensitive

Tag Names Cannot Contain Spaces

Attribute Values Must Appear Within Quotes

White Space Is Preserved

Avoid HTML Tags (Optional)

10. What are XML Namespaces?

XML Namespace is a mechanism to avoid name conflicts by differentiating elements or attributes within an **XML** document that may have identical names, but different definitions. We will be covering the basics of **namespace**, including declaration methods, scope, attribute **namespace**, and default **namespace**.

11. what are XML Schemas?

XML Schema is commonly known as **XML Schema Definition (XSD)**. It is used to describe and validate the structure and the content of XML data. XML schema defines the elements, attributes and data types. Schema element supports Namespaces. It is similar to a database schema that describes the data in a database.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema">
   <xs:element name = "contact">
      <xs:complexType>
         <xs:sequence>
            <xs:element name = "name" type = "xs:string" />
            <xs:element name = "company" type = "xs:string" />
            <xs:element name = "phone" type = "xs:int" />
         </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

12. Write the Simple Type Schema ?

Simple Type

Simple type element is used only in the context of the text. Some of the predefined simple types are: xs:integer, xs:boolean, xs:string, xs:date. For example −

```
<xs:element name = "phone_number" type = "xs:int" />
```

13. What are Complex type Schemas?

Complex Type

A complex type is a container for other element definitions. This allows you to specify which child elements an element can contain and to provide some structure within your XML documents. For example −

```
<xs:element name = "Address">
  <xs:complexType>
    <xs:sequence>
      <xs:element name = "name" type = "xs:string" />
      <xs:element name = "company" type = "xs:string" />
      <xs:element name = "phone" type = "xs:int" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

In the above example, *Address* element consists of child elements. This is a container for other **<xs:element>** definitions, that allows to build a simple hierarchy of elements in the XML document.

14.What are the Global Type Schemas?

Global Types

With the global type, you can define a single type in your document, which can be used by all other references. For example, suppose you want to generalize the *person* and *company* for different addresses of the company. In such case, you can define a general type as follows −

```
<xs:element name = "AddressType">
  <xs:complexType>
    <xs:sequence>
      <xs:element name = "name" type = "xs:string" />
      <xs:element name = "company" type = "xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Now let us use this type in our example as follows −

```
<xs:element name = "Address1">
  <xs:complexType>
    <xs:sequence>
      <xs:element name = "address" type = "AddressType" />
      <xs:element name = "phone1" type = "xs:int" />
      </xs:sequence>
   </xs:complexType>
</xs:element>

<xs:element name = "Address2">
   <xs:complexType>
      <xs:sequence>
         <xs:element name = "address" type = "AddressType" />
         <xs:element name = "phone2" type = "xs:int" />
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Instead of having to define the name and the company twice (once for *Address1* and once for *Address2*), we now have a single definition. This makes maintenance simpler, i.e., if you decide to add "Postcode" elements to the address, you need to add them at just one place.

15. Define The Any Element?

The <any> element enables us to extend the XML document with elements not specified by the schema.

The following example is a fragment from an XML schema called "family.xsd". It shows a declaration for the "person" element. By using the <any> element we can extend (after <lastname>) the content of "person" with any element:

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
      <xs:any minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

16. What is substitution group?

Substitution groups are a feature that allows you to declare that an element can be substituted for other elements in an instance document.

```
<xs:element name="name" type="xs:string"/>
<xs:element name="navn" substitutionGroup="name"/>
```

In the example above, the "name" element is the head element and the "navn" element is substitutable for "name".

Look at this fragment of an XML schema:

```
<xs:element name="name" type="xs:string"/>
<xs:element name="navn" substitutionGroup="name"/>

<xs:complexType name="custinfo">
  <xs:sequence>
    <xs:element ref="name"/>
  </xs:sequence>
</xs:complexType>



<xs:element name="customer" type="custinfo"/>
<xs:element name="kunde" substitutionGroup="customer"/>
```

17. What are SAX?

The **Simple API for XML** (SAX) is a programming interface that acts as an event-based sequential access parser application programming interface (**API**) **for XML** documents. It provides an abstract interface in which the XML information set (infoset) is processed in a linear sequence of method calls.

18. What is DOM?

The **Document Object Model** (**DOM**) is a cross-platform and language-independent interface that treats an XML or HTML document as a tree structure wherein each node is an object representing a part of the document. The DOM represents a document with a logical tree. Each branch of the tree ends in a node, and each node contains objects. DOM methods allow programmatic access to the tree; with them one can change the structure, style or content of a document. Nodes can have event handlers attached to them. Once an event is triggered, the event handlers get executed.

19. What are the Advantages of Web services?

- less expensive to use. This means that if we maintain less number of records or pages then we will require less no of vendors. Less number of vendors mean less efforts is used in maintainig and updating the records.

- Data quality Web services protect data from errors which usually comes when working with database. The data quality web services also improves and increase sales of web service business.
- One of the advantages of webservices is interoperability. Interoperability Similarly web service offer interoperability so that on can easily work.
- Implementation is another advantage of web services. If we talk about word implementation in web services we actually meant to say deployment. Web services are deployed or arranged over internet technology.
- Application to Application interaction are built on standards such as XML, WSDL, UDDI, HTTP. These standard solve many communication problems.

20. Why do we need Web Services?

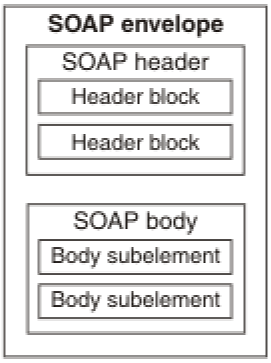
Web Services allows different applications to talk to each other and share data and **services** among themselves. Other applications can also use the **services** of the **web services**. ... So, **Web services** is used to make the application platform and technology independent.

UNIT 2:

1.Define SOAP Model?

SOAP is a communication protocol designed to communicate via Internet. **SOAP** can extend HTTP for XML messaging. **SOAP** provides data transport for **Web services**. ... **SOAP** enables client applications to easily connect to remote **services** and invoke remote methods.

2. Define the structure of SOAP message?



```
<?xml version='1.0' ?>

<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">

 <env:Header>

  <m:reservation
xmlns:m="http://travelcompany.example.org/reservation"

          env:role="http://www.w3.org/2003/05/soap-envelope/role/next"

           env:mustUnderstand="true">

   <m:reference>uuid:093a2da1-q345-739r-ba5d-
pqff98fe8j7d</m:reference>

   <m:dateAndTime>2001-11-29T13:20:00.000-05:00</m:dateAndTime>

  </m:reservation>

  <n:passenger xmlns:n="http://mycompany.example.com/employees"

          env:role="http://www.w3.org/2003/05/soap-envelope/role/next"

           env:mustUnderstand="true">
   <n:name>Åke Jógvan Øyvind</n:name>
  </n:passenger>
 </env:Header>
 <env:Body>
  <p:itinerary
    xmlns:p="http://travelcompany.example.org/reservation/travel">
   <p:departure>
     <p:departing>New York</p:departing>
     <p:arriving>Los Angeles</p:arriving>
     <p:departureDate>2001-12-14</p:departureDate>
     <p:departureTime>late afternoon</p:departureTime>
     <p:seatPreference>aisle</p:seatPreference>
   </p:departure>
   <p:return>
     <p:departing>Los Angeles</p:departing>
     <p:arriving>New York</p:arriving>
     <p:departureDate>2001-12-20</p:departureDate>
     <p:departureTime>mid-morning</p:departureTime>
     <p:seatPreference/>
   </p:return>
  </p:itinerary>
  <q:lodging
   xmlns:q="http://travelcompany.example.org/reservation/hotels">
   <q:preference>none</q:preference>
  </q:lodging>
 </env:Body>
</env:Envelope>
```

3. Define the SOAP Message?

SOAP message is an ordinary XML document containing the following elements:
An **Envelope** element that identifies the XML document as a **SOAP message**. A Header element that contains header information. A Body element that contains call and response information. A Fault element containing errors and status information.

4. Define SOAP Envelope?

SOAP provides the **envelope** for sending **Web Services** messages over the Internet/Internet. ... The **SOAP envelope** contains two parts: An optional header providing information on authentication, encoding of data, or how a recipient of a **SOAP** message should process the message. The body that contains the message.

```
<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
  ...
  Message information goes here
  ...
</soap:Envelope>
```

5. Define SOAP Header?

The **SOAP header**. The **SOAP** <**Header**> is an optional element in a **SOAP** message. It is used to pass application-related information that is to be processed by **SOAP** nodes along the message path. The

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <UserID xmlns="http://tempuri.org/">jsmith</UserID>
    <Password xmlns="http://tempuri.org/">25jeL3n</Password>
  </s:Header>
  <s:Body xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns1="http://tempuri.org/">
    <ns1:ProcessData></ns1:ProcessData>
  </s:Body>
</s:Envelope>
```

child elements of the <**Header**> element are called **header** blocks.

6. Define SOAP fault?

SOAP fault is an **error** in a **SOAP** (Simple Object Access Protocol) communication resulting from incorrect message format, header-processing problems, or incompatibility between applications.

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance" xmlns:xsd="http://www.w3.org/1999/XMLSchema">
      <SOAP-ENV:Body>
         <SOAP-ENV:Fault>
         <faultcode xsi:type="xsd:string">SOAP-ENV:Client</faultcode>
        <faultstring xsi:type="xsd:string">
            Failed to locate method (GetTutorialID) in class (GetTutorial)
         </faultstring>
    </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

7. Define SOAP encoding?

SOAP Encoding is an extension of the **SOAP** framework specification that defines how a data value should be **encoded** in an XML format. **SOAP** Data Model is defined as an adjunct in **SOAP** 1.2 specification. **SOAP encoding** offers the following rules to convert any data value defined in **SOAP** data model into XML format.

8. Define SOAP RPC?

SOAP-**RPC** defines a model for representing an **RPC** and an **RPC** response using the **SOAP** infrastructure. It is not necessarily bound tightly to a synchronous request/reply model, or to the HTTP protocol. ... When using **SOAP**-**RPC**, the body of the envelope contains the method name and the parameters for the procedure call.

9. Define WSDL Model?

WSDL is an XML format for describing network **services** as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint.

10. write elements of WSDL?

<types>	Defines the (XML Schema) data types used by the web service

<message>Defines the data elements for each operation

<portType> Describes the operations that can be performed and the messages involved.

<binding>   Defines the protocol and data format for each port type

11. Write the structure of WSDL ?

```
definitions>

<types>
  data type definitions........
</types>

<message>
  definition of the data being communicated....
</message>

<portType>
  set of operations......
</portType>

<binding>
  protocol and data format specification....
</binding>

</definitions>
```

12. Define binding?

Binding means coonecting your **web service** functions to execution of that function. You define your data types , elements type, complex element,simple elements in xsd etc. Then xsd is used in wsdl to announce the external world's that this is the types of data we accepts, and this is type of data we return.

Binding is an concrete part of **WSDL**. Describes how the operation is invoked by specifying concrete protocol and data format specifications for the operations and messages. **bindings** are three types. SOAP **Binding**: SOAP **binding** allows either document or rpc style with either encoding or literal .

13. Write the example for binding?

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>
```

```
<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>

<binding type="glossaryTerms" name="b1">
   <soap:binding style="document"
   transport="http://schemas.xmlsoap.org/soap/http" />
   <operation>
     <soap:operation soapAction="http://example.com/getTerm"/>
     <input><soap:body use="literal"/></input>
     <output><soap:body use="literal"/></output>
  </operation>
</binding>
```

14. Define binding types?

The **binding** element has two attributes - name and type.

The name attribute (you can use any name you want) defines the name of the binding, and the type attribute points to the port for the binding, in this case the "glossaryTerms" port.

UNIT-3

1. **What is UDDI?**

UDDI is an XML-based standard for describing, publishing, and finding web services. UDDI stands for Universal Description, Discovery, and Integration. UDDI is a specification for a distributed registry of web services. UDDI is a platform-independent, open framework.

2. **Define UDDI specification.**

The UDDI specification defines services that support the description and discovery of: Businesses, organizations and other providers of Web services; The Web services they make available; The technical interfaces which may be used to access and manage those services.

3. **What are the web services development lifecycle?**
   - **Planning**
   - **Analysis and design**
   - **Construction and testing**
   - **Provisioning**

- **Deployment**
- **Execution and monitoring**

4. **Define the conversation language in web services.**

The Web Service Conversation Language (WSCL) proposal defines the overall input and output message sequences for one web service using a finite state automaton FSA over the alphabet of message types.

5. **What is WSCL?**

Web Services Conversation Language (WSCL) Web Services Conversation Language (WSCL) allows the business level conversations or public processes supported by a Web service to be defined. WSCL specifies the XML documents being exchanged, and the allowed sequencing of these document exchanges.

6. **Define WSCL interface components.**

Each interaction in a WSCL message consists of a parent Interaction element that has attributes interactionType (one of those permitted interaction types: Send, Receive, SendReceive, and ReceiveSend) and a unique identifier.

7. **Define workflow.**

A workflow is a sequence of tasks that processes a set of data. Workflows occur across every kind of business and industry. Anytime data is passed between humans and/or systems, a workflow is created. Workflows are the paths that describe how something goes from being undone to done, or raw to processed.

8. **What is a workflow process?**

Workflow is the definition, execution and automation of business processes where tasks, information or documents are passed from one participant to another for action, according to a set of procedural rules.

9. **What are the 3 basic components of workflow?**

Each workflow component or step may be described by three parameters: input, transformation, and output.

- **Input: The materials and resources that are required to complete a step.**
- **Transformation: A specific set of rules that dictate how the input is received and what is done to it.**

10. What are workflows used for?

A workflow application is a software tool that automates the tasks involved in a business process. The workflow application routes data along a predefined path until an item in the process is completed. Tasks in the workflow may include approvals, adding information, or data transfers

11. What are the types of workflows?

Here are three major types of workflow:

Process Workflow.

Case Workflow.

Project Workflow

12. What are the components of workflow?

An activity is a single step in the workflow process. An activity can contain multiple components, including transitions, variables, and actions, but must contain start and end activities.

13. Define quality of service.

Quality of service is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. Quality of service is particularly important for the transport of traffic with special requirements.

14. What is quality of service in computer network?

Quality of service (QoS) refers to any technology that manages data traffic to reduce packet loss, latency and jitter on the network. QoS controls and manages network resources by setting priorities for specific types of data on the network

15. Why is Quality of Service QoS important?

Providing sufficient Quality of Service (QoS) across IP networks is becoming an increasingly important aspect of today's enterprise IT infrastructure. Not only is QoS necessary for voice and video streaming over the network, it's also an important factor in supporting the growing Internet of Things (IoT).

16. Discuss the metrics of QoS.

QoS Metrics for Web Services. QoS metrics are a set of qualitative measures of how appealing a Web service is for applications that may potentially consume it.

17. QoS enabled Web services

The interface definition (WSDL) specifies the syntactic signature for a service but does not specify any semantics or non-functional aspects. QoS enabled Web services require a separate QoS language for Web services

18. what is mobile wireless communication?

Wireless communication networks offer transmission of signals, such as voice, data, and multimedia, without using wires, which is the crucial part of mobile communications. ... Wireless communication is the fastest growing segment of the communication industry.

19. Is wireless mobile or is mobile wireless?

The terms "mobile" and "wireless" are often used interchangeably but in reality, they are two very different concepts applied to modern computing and technology. Mobile is a word that is commonly used to describe portable devices. ... Traditional computers or other non-mobile devices can access wireless networks.

20. What is mobile connectivity?

MOBILE CONNECTIVITY. Description of the Application. The ability to connect mobile devices, such as a handheld computer or

a cellular telephone, to the network is no longer a “nice-to-have”; it is quickly becoming a requirement for organizations worldwide.

21. Challenges in Mobile Testing

Major Roadblocks While Testing Mobile Apps

- **User Experience & Issues with App Performance. ...**
- **Touch Screen. ...**
- **Context Awareness. ...**
- **Lesser Time to Market the App. ...**
- **Security Issues. ...**
- **Variations of Mobile User Interfaces. ...**
- **Lack of Access to Multiple Devices. ...**
- **Device Based Testing Approach.**

UNIT-4

1. **What is the limits of html is not used?**

HTML Does Not:

- **Recognize spaces, tabs or margins.**
- **Translate easily from other types of documents.**
- **Support all font styles.**
- **Recognize line breaks, or paragraph breaks (unless you code it in).**

2. **What is the limits of html do?**

HTML Does:

Support multiple text sizes	**<FONT SIZE=X> </FONT>**
Support multiple fonts	**<FONT Face="XXXX"> </FONT>**
Support bold, underline, and italic version of text	**<B></B>, <U></U>, <I></I>**
Have multiple Text colors	**<FONT COLOR="#XXXXXX"> </FONT>**
Support Background Colors.	**<BODY BGCOLOR = "#XXXXXX"> </BODY>**
Support tables, the only way you can "fake" tabs and margins.	**<TABLE> <TR><TD></TD></TR></TABLE>**
Allow you to insert graphics, sound, videos.	**<IMG SRC = "#XXXXXX">**

Have built-in graphics for line rules	<HR>
Supports itemized lists with dots or numbers	<UL><LI>...<LI>...<LI> </UL> like this bulleted list.

3. **What is the scope of html?**

The scope attribute specifies whether a header cell is a header for a column, row, or group of columns or rows.

The scope attribute has no visual effect in ordinary web browsers, but can be used by screen readers.

4. **What is the basic structure of HTML?**

Basic structure of an HTML document. An HTML document has two* main parts: head. The head element contains title and meta data of a web document.

5. **What is the content of the structure in html?**

The content attribute gives the value associated with the http-equiv or name attribute.

6. **What is the structure and presentation in html?**

Structure consists of the mandatory parts of an HTML document plus the semantic and structured markup of its contents. Presentation is the style you give the content. In most cases presentation is about the way a document looks, but it can also affect how a document sounds – not everybody uses a graphical web browser.

7. **Which element represents a section of a document that links to other documents?**

    The HTML <nav> element represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents.
    Common examples of navigation sections are menus, tables of contents, and indexes.

8. **What is a markup language?**

Markup languages are designed for presentation of text in different formats, and it can also be used for transporting and storing data. This markup language specifies the code for formatting, layout and style of data .This markup code is called Tag.

HTML and XML are examples of Markup Language.

9. What is XML?

XML is called Extensible Markup Language which is designed to carry or transport and store data. XML tags are not as predefined as HTML, but we can define our own user tags for simplicity. It mainly concentrates on storing of data, not on displaying of data.

10. What are the features of XML?

Main features of XML are:

• Very easy to learn and implement

• XML files are text files, and no editor is required

• Minimal and a limited number of syntax rules in XML

• It is extensible, and it specifies that structural rules of tags

11. What are the differences between HTML and XML?

Following are the differences between HTML and XML:

HTML	XML
Markup language used to display data	Markup language used to store da

Case Insensitive	Case sensitive
Designing web pages	Used to transport and store data
Predefined Tags	Custom Tags
Does not Preserve white spaces	Preserve white spaces
Static	Dynamic

12. What do you understand by XML-RPC?

RPC is Remote Procedure Call and as the name suggests, it is the method of calling a procedure or function available on any remote computer.

XML stands for Extensible Markup Language. Thus XML-RPC represents a simple protocol that performs RPCs by using XML messaging. This has been considered as an excellent tool for connecting different environments and also establishing connections between wide varieties of computers.

13. Explain features of XML-RPC?

The major features of XML-RPC are enlisted below

- **RPCs are performed using simple XML language.**
- **XML encoded Requests are sent via HTTP POST.**
- **XML Response is embedded in HTTP response.**

- **It is considered as platform-independent.**
- **It allows communication between diverse applications.**
- **It uses HTTP protocol for passing information between client and server computers.**
- **It has small XML vocabulary for describing request and response's nature.**

UNIT-5

1. **Why XML is used in Web services?**

XML is used to encode all communications to a web service. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer.

2. **What is the purpose of XML in a Web service?**
   By using Web services, your application can publish its function or message to the rest of the world. Web services use XML to code and to decode data, and SOAP to transport it (using open protocols).

3. **XML Web Services**

Web services are web application components.

Web services can be published, found, and used on the Web.

This tutorial introduces WSDL, SOAP, RDF, and RSS.

4. **XML WSDL**

   WSDL stands for Web Services Description Language

   WSDL is used to describe web services

   WSDL is written in XML

   WSDL is a W3C recommendation from 26. June 2007

5. **XML Soap**

- **SOAP stands for Simple Object Access Protocol**
- **SOAP is an application communication protocol**
- **SOAP is a format for sending and receiving messages**
- **SOAP is platform independent**
- **SOAP is based on XML**

- **SOAP is a W3C recommendation**

6. **What is an XML Element?**

An XML element is everything from (including) the element's start tag to (including) the element's end tag.

An element can contain:

- **text**
- **attributes**
- **other elements**
- **or a mix of the above**

7. **What is XML publishing?**

XML is considered an ideal technology around which to build a publishing process. It is a platform and software independent language that can be transformed into a variety of

common publishing formats, including content formats used on the Web.

8. **An XML-based Publishing Process**

XML Content Creation.

- **an XML-based publishing process is the creation or acquisition of content in an appropriate XML vocabulary. ...**
- **Input Validation. ...**
- **Producing electronic publication formats from XML.**

9. **What is XML data?**

Extensible Markup Language (XML) is used to describe data. The XML standard is a flexible way to create information formats and electronically share structured data via the public Internet, as well as via corporate networks. ... Both XML and HTML contain markup symbols to describe page or file contents.

10. **What is the format of XML file?**

XML is a file extension for an Extensible Markup Language (XML) file format used to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere using standard ASCII text. XML is similar to HTML.

11. What is a client side application?

Client-side means that the action takes place on the user's (the client's) computer. Server-side means that the action takes place on a web server. ... For this reason, most things that can be accomplished using JavaScript can be done without needing access to a web server.

12. What are the benefits of client side processing?

Client-side scripts offer numerous advantages, including:

Allow for more interactivity by immediately responding to users' actions.

Execute quickly because they don't require a trip to the server.

May improve the usability of Web sites for users whose browsers support scripts.

13. What are the client side technologies?

Client-side Programming : It is the program that runs on the client machine (browser) and deals with the user interface/display and any other processing that can happen on client machine like reading/writing cookies. Refer JavaScript articles for example Client Side codes that run in browser.

14. What is the need for client side scripting?

The Need for Client-Side Scripting

The basic architecture for a web application is that most of the source code and assemblies reside and are processed on a web server. The sole task of a web server is to accept incoming HTTP requests and to return the requested resource in an HTTP response

15. What are the client side technologies?

Client-side Programming : It is the program that runs on the client machine (browser) and deals with the user interface/display and any other processing that can happen on client machine like reading/writing cookies. Refer

JavaScript articles for example Client Side codes that run in browser.

16. what is client side development?

Client side programming has mostly to do with the user interface, with which the user interacts. In web development it's the browser, in the user's machine, that runs the code, and it's mainly done in javascript, flash, etc. This code must run in a variety of browsers. Its main tasks are: ... manipulating UI elements.

17. What is server side application?

Server side and Client side Programming. Server-side Programming : It is the program that runs on server dealing with the generation of content of web page. ... For example if user input is a text in search box, run a search algorithm on data stored on server and send the results.

18. What is server side processing?

Server-side processing is used to interact with permanent storage like databases or files. The server will also render pages to the client and process user input. Server-side processing happens when a page is first requested and when pages are posted back to the server.

19. What is client side application server side application?

Client Side vs. Server Side. Website scripts run in one of two places – the client side, also called the front-end, or the server side, also called the back-end. The client of a website refers to the web browser that is viewing it. The server of a website is, of course, the server that hosts it.

20. What is the server side code main function?

Server-side scripting is a method of designing websites so that the process or user request is run on the originating server. Server-side scripts provide an interface to

the user and are used to limit access to proprietary data and help keep control of the script source code.