Assignment - 4

1) "SOAP" Fault is caused due to client or server failure". State T/F with justifies.

→ True.

2. Give the use of SOAP actor attribute?

→ SOAP actor element is specifically used to address the header element to a specific endpoint this attribute is used to annotate an extension element.

3. What do you mean by wire protocol & transport protocol.

→ it generally refers to communication protocol higher than the physical layer, in contrast to transport protocol at the protocol level (like TCP or UDP) the term wire protocol is used to describe a common way to exchange information at the application level.

4. What is SOAP message path?

→ it is the set of SOAP nodes through which a single SOAP message passes, including the initial SOAP sender, zero or more SOAP intermediaries and an altimates SOAP receiver.

5. Give the use of SOAP must understand attribute.

→ To indicate whether a header entry is mandatory or optional for the recipient to process.

Q Explain short Apache Axis environment.

→ Apache Axis is an implementation of the SOAP is a lightweight protocal for enhancing structured information in a decentralized distributed environment it is an XML based protocol. Apache axis is an open-source it consist of a Java and C++ implementation of SOAP server and various utilities and API's for, generating & deploying web service application.

Q How error are handled using SOAP Faults, give an example for adding fault in XML of SOAP messages.

→ SOAP provides a model for handling situation when fault is occure in the processing of message

1) The SOAP Fault model requires that all SOAP specific and application specific Fault be reported using a special purpose element called env fault.
2) The env:Fault element is a reserved element predefined by the SOAP specification whoes purpose is to provide an extensible mechanism for transporting structure & unstructured info about problem that have arisen the processing of SOAP message.

e.g.:-

```
<env:Body>
  <env:Fault>
```

```
<env:Code>
 <env:value> env:Sender </env:value>
  </env:code>
   <env:Reason>
   <env:Text xml:long = "en-us">
   processing error </env:Text>
<env:text> XML:long = "da"> Processerings
                          -fell
 </env:Text>
   <env:Reason>
   </env:Fault>
   </env:Body>
   </env:Envelope>
```

8 What are the advantages & disadvantages of SOAP

→ advantages:-

1) Simplicity - SOAP is simple as it is based on XML, which is highly structured and easy to parse.

2) Use of open standard - SOAP is use the open standard of XML to format the data which makes it easily extendable and well supported.

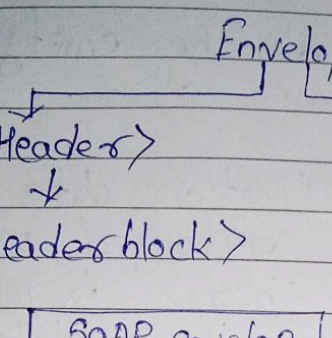
3) Universal acceptance - SOAP is the most widely accepted standard in the message common domain.

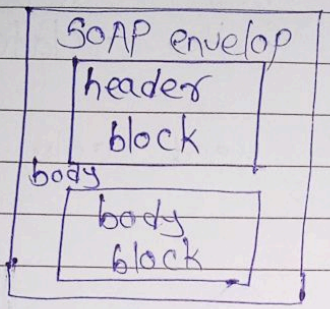
4) Portability - SOAP is portable without any dependancies on the underlying platform like byte ordering issues or machine-word widths.

disadvantages-
1) SOAP was initially tied to http and this mandated a request/response architecture that was not appropriate for all situation.
2) SOAP is stateless.
3) SOAP serializes by value & does not support serialization by references.

Q What is SOAP? give the structure of SOAP message?

→ 1) SOAP is based on message exchanges messages are seen as envelopes where the application encloses the data to be sent.
2) A SOAP message consists of an envelop element containing an optional <Header> and a mandatory <Body> element.
3) The SOAP <Header> contains blockes of information relevant to how the message is to be processed this helps pass information in SOAP message that is not for the application but the SOAP engine.
4) The SOAP <Body> is contain end-to-end info of the message.





- SOAP headers have been designed in anticipation of participant of other SOAP processing nodes. Called SOAP intermediaries - along a messages path from an initial SOAP sender which creates to an ultimate SOAP receiver.
- SOAP message travel along the message path from a sender to receiver.
- All SOAP messages start with an initial sender which creates the SOAP message and end with an ultimate receiver.

SOAP Body -

- The SOAP Body is the area of the SOAP message where the application specific XML data being exchanged in the message is placed.
- The <Body> element must be present and is an immediate child of the envelop it may contain number of child elements called body.

Entries but it may also be empty. The <Body> element contain other.
Following -
application - specification data - it is the information that is enchanged with a web service. The SOAP <body> is where the method call info. & its related arguments are encoded.
Fault message - is used only when an error occur.

Envelope

```
<env:Envelope xmlns:env="http://www.w3.org/
        2022/001 soap-envelope">
Header ← <env:Header>
  <t:transaction ID
   xmlns t="http://intermediatary.example.com
            procedurement"
   env:role="http://www.w3.org/2002/06/soap-
   envlope/role/next"
   env:must understand="true">
     5739
   </t transaction ID>
  </env:Header>
Block
Body
  <env:Body>
    <m:order Goods
    env:encoding style="http://www.w3.org/
          2002/06/soap encoding"
```

Body —

Q. In
→

```
xmlns:m=http//example.com/procurement">
<m:product Item>
<name> ACME softner </name>
</m:product Item>
<m:quantity>
35
</m:quantity>
</m:order Goods>
</env:Body>
<env:envelope>
```

Body

Q. Write an example of Document-style web service.

→

```
<env:Envelope>
XMLNS:SOAP = "http://www.w3.org/2003/05/soap-
envelop>
<env:Header>
<t:Transaction-id
xmlns:t="http://www.transaction.com/transaction
env:mustunderstand="1">
512
</env:Header>
<env:Body>
<po:purches order order date="2004-12-02"
xmlns:m="http://www.plastic-supply.com/pas">
<po:from>
<po:accName> Right plastics </po:accName>
<po:accNumber> PSC-0343-02 </po:accNumber>
</po:From>
<po:to>
<po:SuppName> plastic supplies inc. </po.SuppName>
```

<Po: SuppAddress> yara valley Melbourne
</Po: SuppAddress> </Po: to>
<Po : product>
<Po : proud - name> injection molder </Po:
prod - name>
<Po: prod-model> G-100 T </Po: prod - model>
<Po: qty> 2 </Po: qty>
</Po: product>
</Po: purchase order>
</env: Body>
</env: envelope>