



# Arcitura Education

**S90.05 Exam**

**Arcitura Education SOA Technology Lab Exam**

**Thank you for Downloading S90.05 exam PDF Demo**

**You can Buy Latest S90.05 Full Version Download**

<https://www.certkillers.net/Exam/S90.05>

<https://www.certkillers.net>

## Version: 5.0

---

### Question: 1

---

Your company has developed a service that allows your customers to check the status of an order. The schema for the service is shown here:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="OrderStatusLookup">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="customerID" type="xsd:string"/>
        <xsd:element name="orderID" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="OrderStatusResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="orderID" type="xsd:string"/>
        <xsd:element name="status" type="xsd:string"/>
        <xsd:element name="comments" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

A new customer plans to use your service, but they want to test it first. They have asked for two sample XML documents that show typical input and output message content. Which of the following pairs of XML elements will validate with the schema shown above?

- A. 

```
<OrderStatusLookup>
  <customerID>12345</customerID>
  <orderID>44F12</orderID>
</OrderStatusLookup>

<OrderStatusResponse>
  <orderID>44F12</orderID>
  <status>PENDING</status>
  <comments>Waiting for back order to be filled</comments>
  <comments>Will advise when in stock.</comments>
</OrderStatusResponse>
```
- B. 

```
<OrderStatusLookup>
  <customerID>12345</customerID>
  <orderID>44F12</orderID>
</OrderStatusLookup>

<OrderStatusResponse>
  <orderID>44F12</orderID>
  <comments>Waiting for back order to be filled</comments>
</OrderStatusResponse>
```
- C. 

```
<OrderStatusLookup>
  <orderID>44F12</orderID>
  <customerID>12345</customerID>
</OrderStatusLookup>

<OrderStatusResponse>
  <orderID>44F12</orderID>
  <status>PENDING</status>
  <comments>Waiting for back order to be filled</comments>
</OrderStatusResponse>
```
- D. None of the above.

- A. Option A  
B. Option B  
C. Option C  
D. Option D

---

**Answer: D**

---

---

### Question: 2

---

A partner organization has given you samples of XML documents retrieved from their product information database. The three samples they have sent you are:

```
<product>
  <number>123DD</number>
  <name>3-Compartment Travel Kit</name>
  <quantity>2</quantity>
</product>

<product>
  <number>445AB</number>
  <quantity>1</quantity>
  <name>13 inch. Rolling Suitcase</name>
</product>

<product>
  <number>943AC</number>
  <quantity>1</quantity>
  <name>Deluxe Travel Pillow</name>
</product>
```

Because they are an important partner, you have agreed to write a service to accept XML documents in their format. However, they did not send you a schema, so you are tasked with defining a schema that will correctly validate their documents. Which of the following XML schemas will successfully validate each of the above XML document fragments?

- A. 

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="product">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="number" type="xsd:string"/>
        <xsd:element name="name" type="xsd:string"/>
        <xsd:element name="quantity" type="xsd:integer"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```
- B. 

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="product">
    <xsd:complexType>
      <xsd:all>
        <xsd:element name="quantity" type="xsd:integer"/>
        <xsd:element name="number" type="xsd:string"/>
        <xsd:element name="name" type="xsd:string"/>
      </xsd:all>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```
- C. 

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="product">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="number" type="xsd:string"/>
        <xsd:element name="name" type="xsd:string"/>
        <xsd:element name="quantity" type="xsd:integer"/>
        <xsd:element name="name" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```
- D. None of the above.

- A. Option A  
B. Option B  
C. Option C  
D. Option D

---

**Answer: B**

---

---

### Question: 3

---

Your company has developed a PO service that allows customers to submit purchase order documents. The message sent by a customer must be based on a predefined "purchaseOrder" element and the message received by the customer is based on a "messageAcknowledgement" element. These two elements are defined in the following two separate XML Schema definitions:

The "purchaseOrder.xsd" schema:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.org/po">
  <xsd:element name="purchaseOrder">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="number" type="xsd:string"/>
        <xsd:element name="date" type="xsd:date"/>
        <xsd:element name="amount" type="xsd:decimal"/>
        <xsd:element name="description" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

The "ack.xsd" schema:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.org/ack">
  <xsd:element name="messageAcknowledgement">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="number" type="xsd:string"/>
        <xsd:element name="dateReceived" type="xsd:date"/>
        <xsd:element name="status" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Note that these two schemas have different namespaces. That is because the "purchaseOrder.xsd" schema is specific to purchase order-related functions, while the "ack.xsd" schema is more generic and used for a variety of different purposes.

- C A. `<definitions name="POService" targetNamespace="http://www.example.org/wsdl/po" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:po="http://www.example.org/po" xmlns:ack="http://www.example.org/ack" xmlns:xsd="http://www.w3.org/2001/XMLSchema">`  
`<types>`  
`<xsd:schema targetNamespace="http://www.example.org">`  
`<xsd:import namespace="http://www.example.org/po" schemaLocation="purchaseOrder.xsd"/>`  
`<xsd:import namespace="http://www.example.org/ack" schemaLocation="ack.xsd"/>`  
`</xsd:schema>`  
`</types>`  
`<message name="submitPOMessage">`  
`<part name="PO" element="po:purchaseOrder"/>`  
`</message>`  
`<message name="POAcknowledgement">`  
`<part name="PO" element="ack:messageAcknowledgement"/>`  
`</message>`  
`...`  
`</definitions>`
- C B. `<definitions name="POService" targetNamespace="http://www.example.org/wsdl/po" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:po="http://www.example.org/po" xmlns:ack="http://www.example.org/ack" xmlns:xsd="http://www.w3.org/2001/XMLSchema">`  
`<types>`  
`<xsd:schema targetNamespace="http://www.example.org">`  
`<xsd:import namespace="http://www.example.org/po" schemaLocation="purchaseOrder.xsd"/>`  
`<xsd:import namespace="http://www.example.org/po" schemaLocation="ack.xsd"/>`  
`</xsd:schema>`  
`</types>`  
`<message name="submitPOMessage">`  
`<part name="PO" element="po:purchaseOrder"/>`  
`</message>`  
`<message name="POAcknowledgement">`  
`<part name="PO" element="ack:messageAcknowledgement"/>`  
`</message>`  
`...`  
`</definitions>`
- C C. `<definitions name="POService" targetNamespace="http://www.example.org/wsdl/po" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:po="http://www.example.org/po" xmlns:ack="http://www.example.org/ack" xmlns:xsd="http://www.w3.org/2001/XMLSchema">`  
`<types>`  
`<xsd:schema targetNamespace="http://www.example.org">`  
`<xsd:import namespace="http://www.example.org/po" schemaLocation="purchaseOrder.xsd"/>`  
`<xsd:import namespace="http://www.example.org/ack" schemaLocation="ack.xsd"/>`  
`</xsd:schema>`  
`</types>`  
`<message name="submitPOMessage">`  
`<part name="PO" element="po:purchaseOrder"/>`  
`</message>`  
`<message name="POAcknowledgement">`  
`<part name="PO" element="po:messageAcknowledgement"/>`  
`</message>`  
`...`  
`</definitions>`
- C D. None of the above.

- A. Option A
- B. Option B
- C. Option C
- D. Option D

---

**Answer: A**

---

---

**Question: 4**

---

Your developer is asked to build a Timesheet service that is able to receive timesheet submissions for an internal payroll system. Service consumers need to be able to send this service request message based on a root "employeeHours" element within the SOAP message body. The "employeeHours" element will contain "empID" and "hours" child elements. Furthermore, the service needs to return a message based on a "response" element that contains "empID" and "status" child elements. Your developer comes to you with the following WSDL definition:

CertKillers.net



```

<definitions name="Timesheet"
  targetNamespace="http://www.example.org/wsdl/timesheet"
  xmlns:tns="http://www.example.org/wsdl/timesheet"
  xmlns:timesheet="http://www.example.org/timesheet"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  <types>
    <xsd:schema
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://www.example.org/timesheet">
      <xsd:element name="employeeHours">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="empID"
              type="xsd:integer"/>
            <xsd:element name="hours"
              type="xsd:decimal"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="response">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="empID"
              type="xsd:integer"/>
            <xsd:element name="status"
              type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="TimesheetMessage">
    <part name="Timesheet"
      element="timesheet:employeeHours"/>
  </message>
  <message name="TimesheetResponse">
    <part name="Response"
      element="timesheet:response"/>
  </message>
  <portType name="TimesheetInterface">
    <operation name="SubmitTimesheet">
      <input message="tns:TimesheetMessage"/>
      <output message="tns:TimesheetResponse"/>
    </operation>
  </portType>
  <binding name="TimesheetBinding"
    type="tns:TimesheetInterface">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="SubmitTimesheet">
      <soap:operation
        soapAction="http://www.example.org/SubmitTimesheet"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  ...
</definitions>

```

The only revision you suggest is that the value of the "style" attribute of the "soap: binding" element be changed to "document" instead of "rpc". Why?

- A. With a value of "rpc", the root element within the SOAP message body will have the name of the operation, and therefore cannot be named "employeeHours" as indicated by the requirements.
- B. With a value of "rpc", the root element within the SOAP message body is unable to be part of an operation that contains both "input" and "output" elements.
- C. The value of "rpc" is not a valid value for the "style" attribute.
- D. None of the above.

---

**Answer: A**

---

CertKillers.net

## Thank You for trying S90.05 PDF Demo

To Buy Latest S90.05 Full Version Download visit link below

<https://www.certkillers.net/Exam/S90.05>

## Start Your S90.05 Preparation

**[Limited Time Offer]** Use Coupon “CKNET” for Further discount on your purchase. Test your S90.05 preparation with actual exam questions.

<https://www.certkillers.net>