Introduction to Web Services (SOAP & REST)

Working with JAX-WS

Lesson Objectives

- Working with JAX-WS
  - WSDL
  - SOAP
- Creating JAX-WS service
- Consuming web service

Working with JAX - WS

- Java API for XML web services (JAX-WS), is a set of APIs for creating web services in XML format

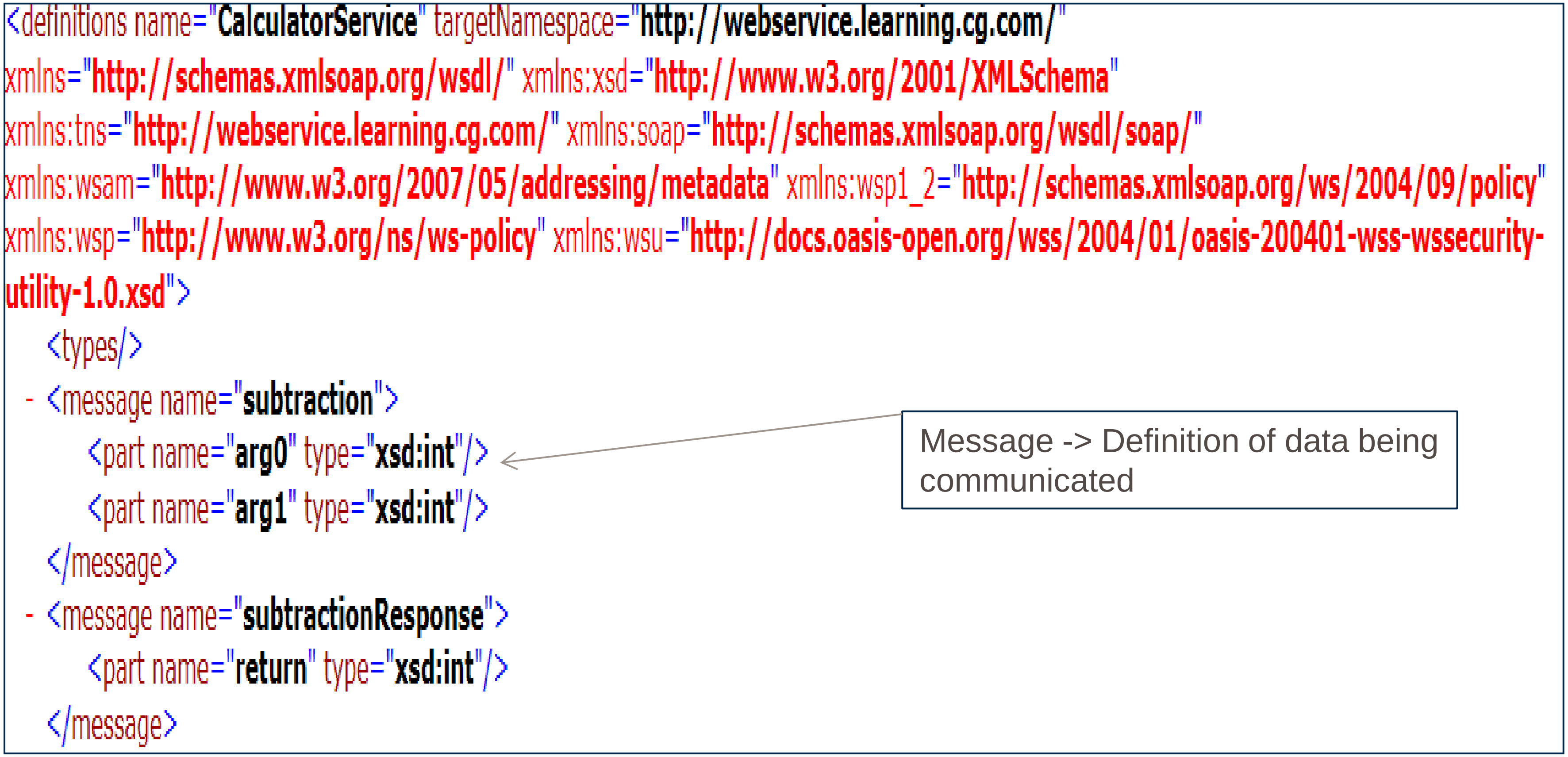
- In JAX-WS, a web service operation invocation is represented by an XML-based protocol, such as SOAP. The SOAP specification defines the envelope structure, encoding rules, and conventions for representing web service invocations and responses. These calls and responses are transmitted as SOAP messages (XML files) over HTTP
- Although SOAP messages are complex, the JAX-WS API hides this complexity from the application developer. On the server side, the developer specifies the web service operations by defining methods in an interface written in the Java programming language.

What is WSDL ?

- WSDL stands for Web Services Description Language
- WSDL is a document written in XML. This document describes a Web service. It specifies the location of the service and the operations (or the methods) the service exposes
- WSDL documents uses these major elements
  - <types>
    - Defines the datatypes (XML Schems) used by the web service
  - <message>
    - Defines the data elements for each operation
  - <portType>
    - Describes the operations that can be performed and messages involved
  - <binding>
    - Defines the protocol and data format for each port type

WSDL Elements

- Below is the snap shot of the WSDL file:

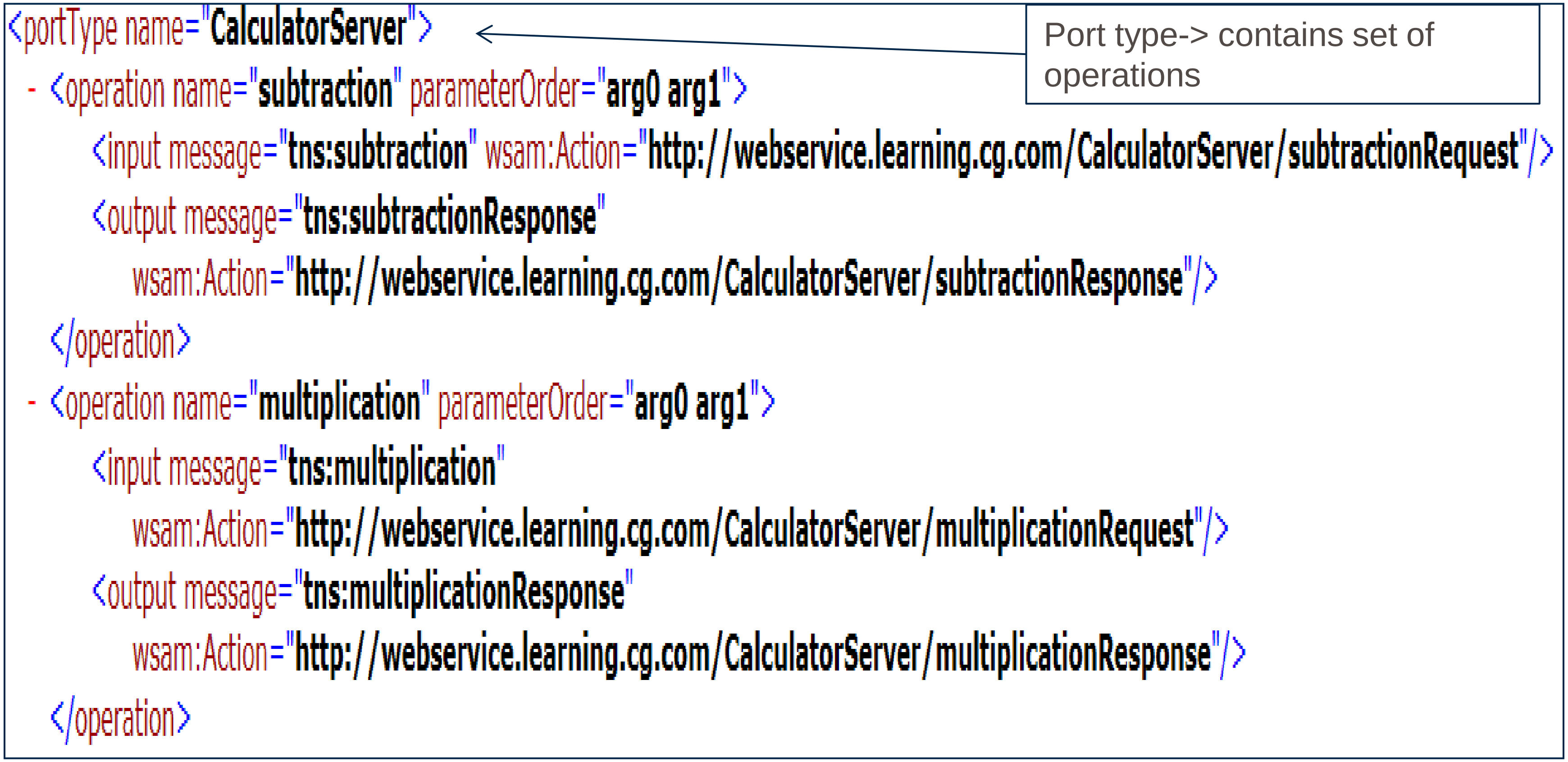


```
<definitions name="CalculatorService" targetNamespace="http://webservice.learning.cg.com/"
xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://webservice.learning.cg.com/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
   <types/>
 - <message name="subtraction">
      <part name="arg0" type="xsd:int"/>
      <part name="arg1" type="xsd:int"/>
   </message>
 - <message name="subtractionResponse">
      <part name="return" type="xsd:int"/>
   </message>
```

Message -> Definition of data being communicated

WSDL Elements

- Below is the snap shot of the WSDL file:

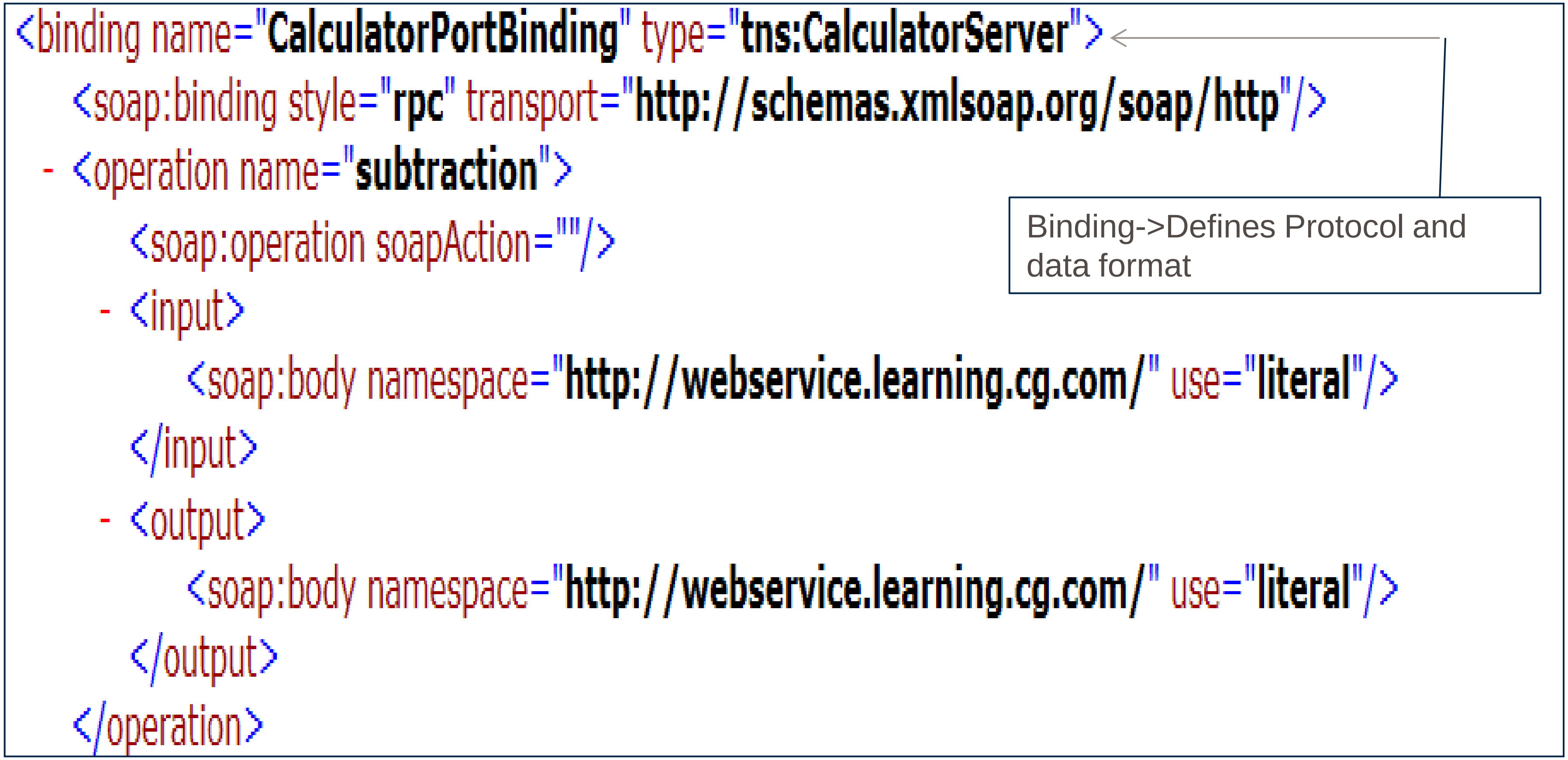


```
<portType name="CalculatorServer">
 - <operation name="subtraction" parameterOrder="arg0 arg1">
      <input message="tns:subtraction" wsam:Action="http://webservice.learning.cg.com/CalculatorServer/subtractionRequest"/>
      <output message="tns:subtractionResponse"
         wsam:Action="http://webservice.learning.cg.com/CalculatorServer/subtractionResponse"/>
   </operation>
 - <operation name="multiplication" parameterOrder="arg0 arg1">
      <input message="tns:multiplication"
         wsam:Action="http://webservice.learning.cg.com/CalculatorServer/multiplicationRequest"/>
      <output message="tns:multiplicationResponse"
         wsam:Action="http://webservice.learning.cg.com/CalculatorServer/multiplicationResponse"/>
   </operation>
```

Port type-> contains set of operations

WSDL Elements

- Below is the snap shot of the WSDL file:

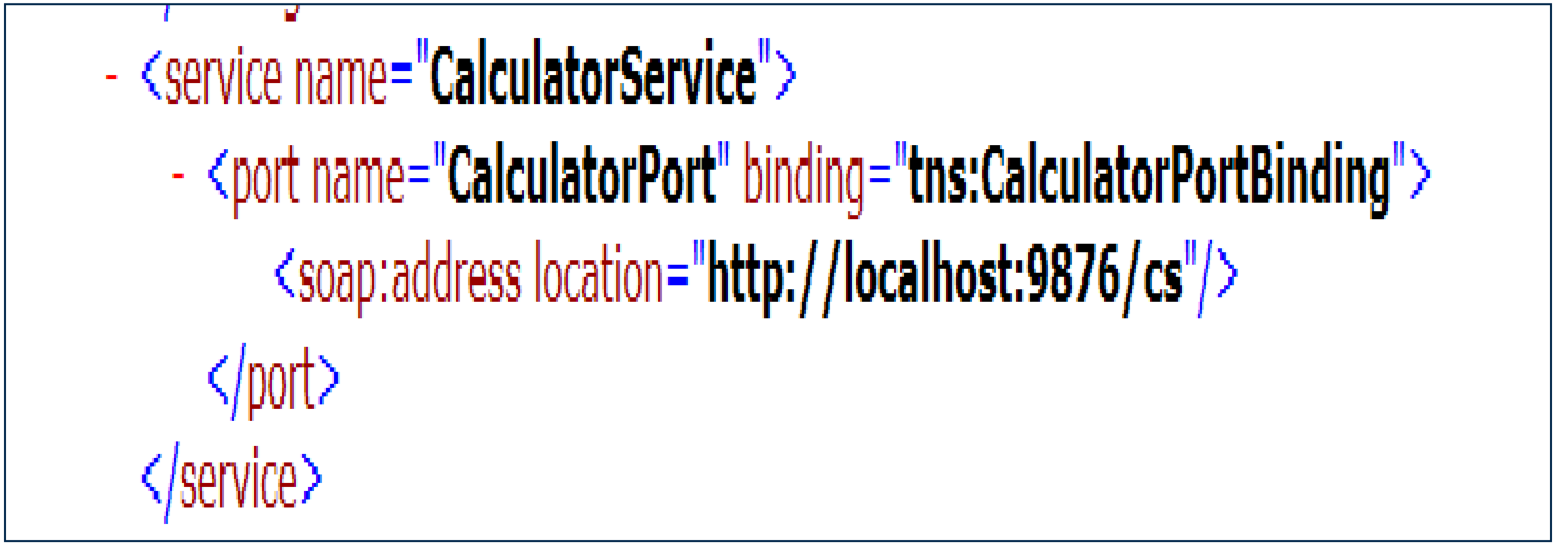


```
<binding name="CalculatorPortBinding" type="tns:CalculatorServer">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  - <operation name="subtraction">
        <soap:operation soapAction=""/>
      - <input>
            <soap:body namespace="http://webservice.learning.cg.com/" use="literal"/>
        </input>
      - <output>
            <soap:body namespace="http://webservice.learning.cg.com/" use="literal"/>
        </output>
    </operation>
```

Binding->Defines Protocol and data format

WSDL Elements

- Below is the snap shot of the WSDL file:



```
- <service name="CalculatorService">
   - <port name="CalculatorPort" binding="tns:CalculatorPortBinding">
       <soap:address location="http://localhost:9876/cs"/>
     </port>
  </service>
```

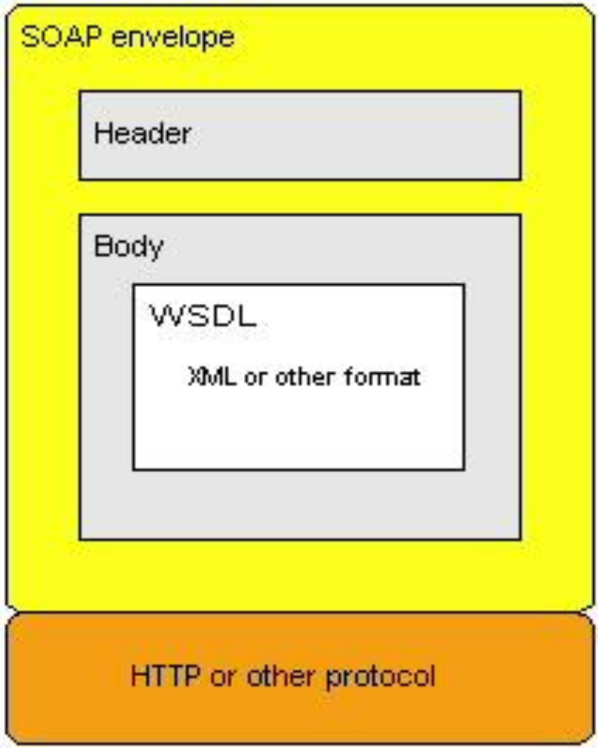
Service: Describes the service name to be used , port name and the soap address, where the web service will be available for consumption

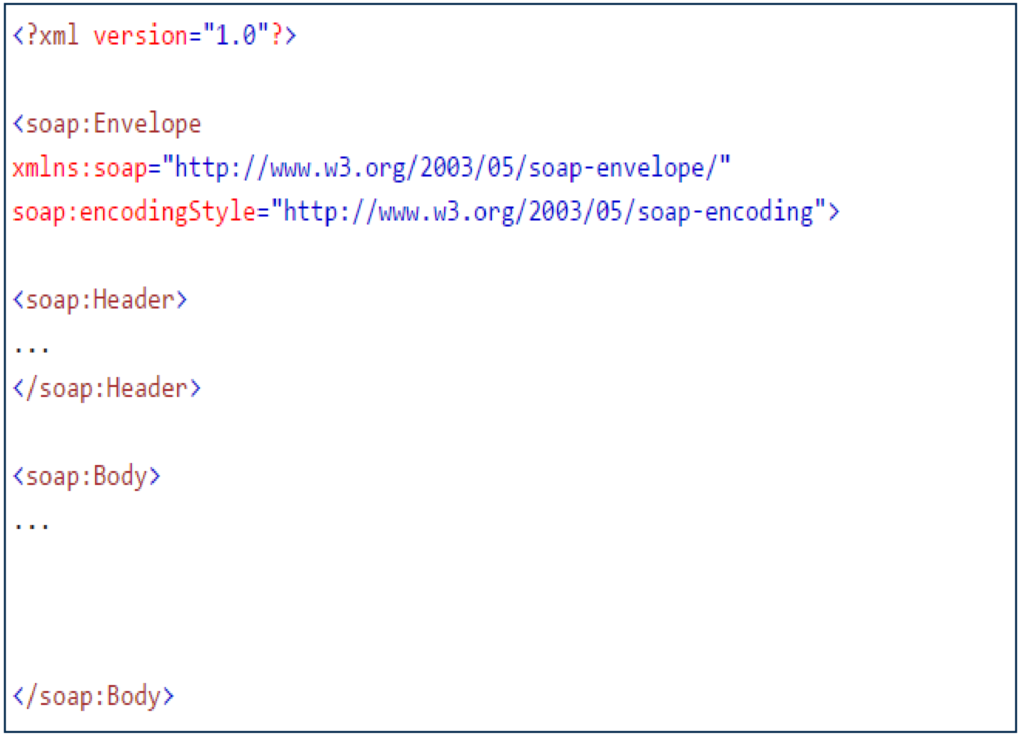
What is SOAP

- SOAP is Simple Object Access Protocol
- Is a format for sending and receiving messages
- It is important for web applications to be able to communicate over the internet
- The best way to communicate between applications is over HTTP, because HTTP is supported by all Internet browsers and servers. SOAP was created to accomplish this
- SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages

SOAP Building Blocks

- Basic SOAP consists of Envelope, Header and Body:

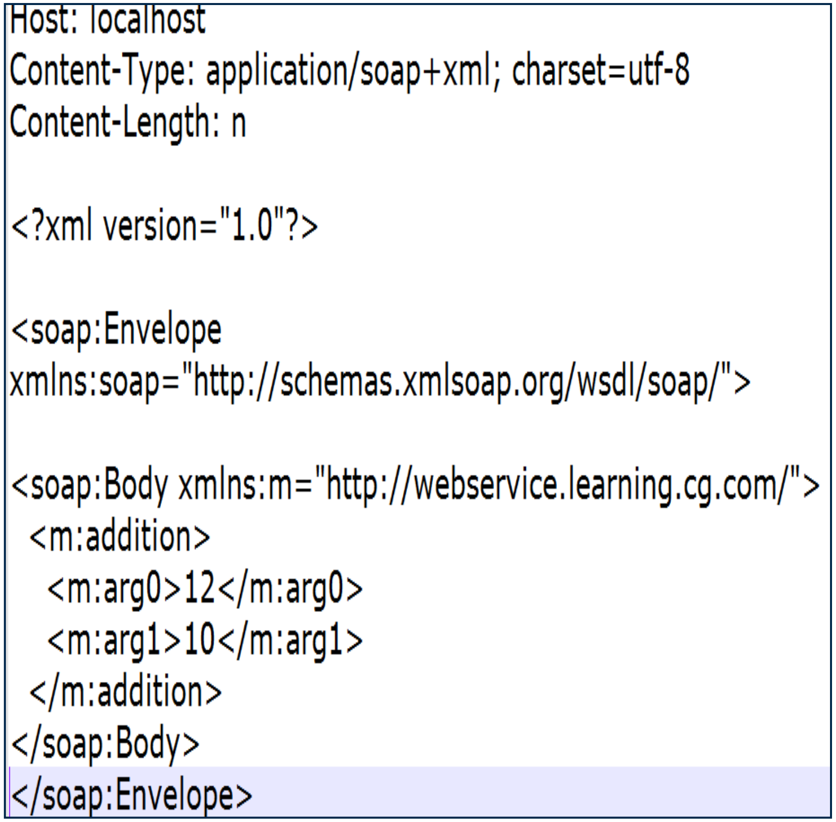




```
<?xml version="1.0"?>

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

<soap:Header>
...
</soap:Header>

<soap:Body>
...



</soap:Body>
```

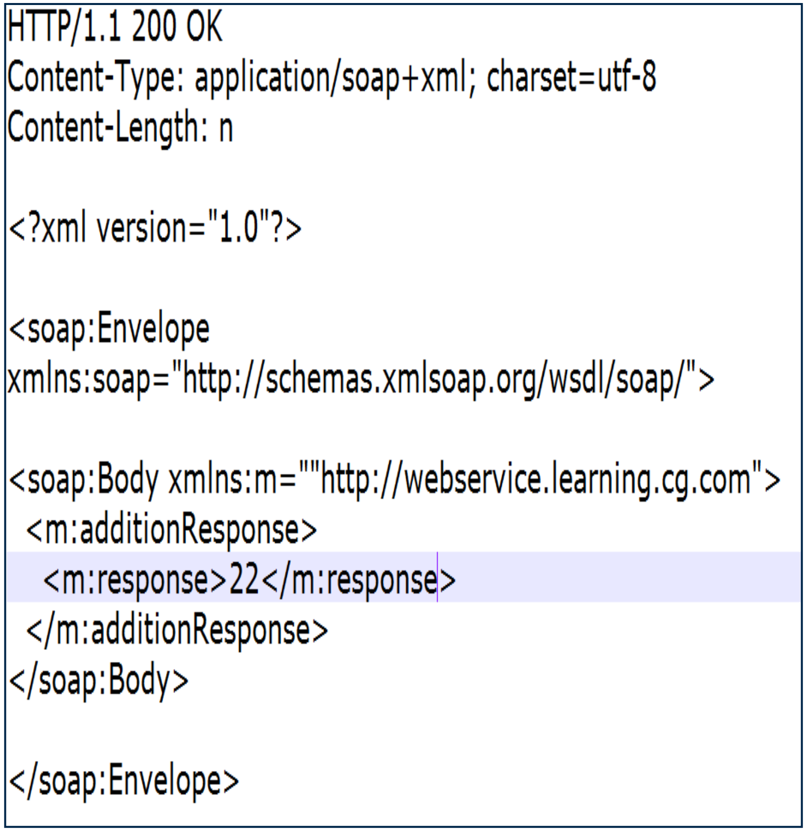
SOAP Request and Response

- Following demonstrates a SOAP request



```
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: n

<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

<soap:Body xmlns:m="http://webservice.learning.cg.com/">
 <m:addition>
  <m:arg0>12</m:arg0>
  <m:arg1>10</m:arg1>
 </m:addition>
</soap:Body>
</soap:Envelope>
```

- Following demonstrates a SOAP response

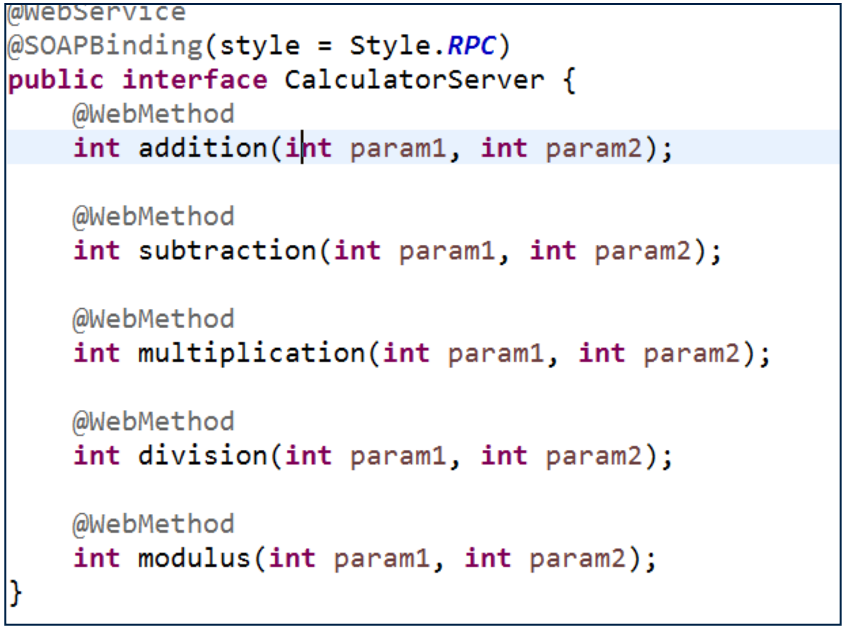


```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: n

<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

<soap:Body xmlns:m=""http://webservice.learning.cg.com">
 <m:additionResponse>
  <m:response>22</m:response>
 </m:additionResponse>
</soap:Body>

</soap:Envelope>
```

2.2: Creating Service

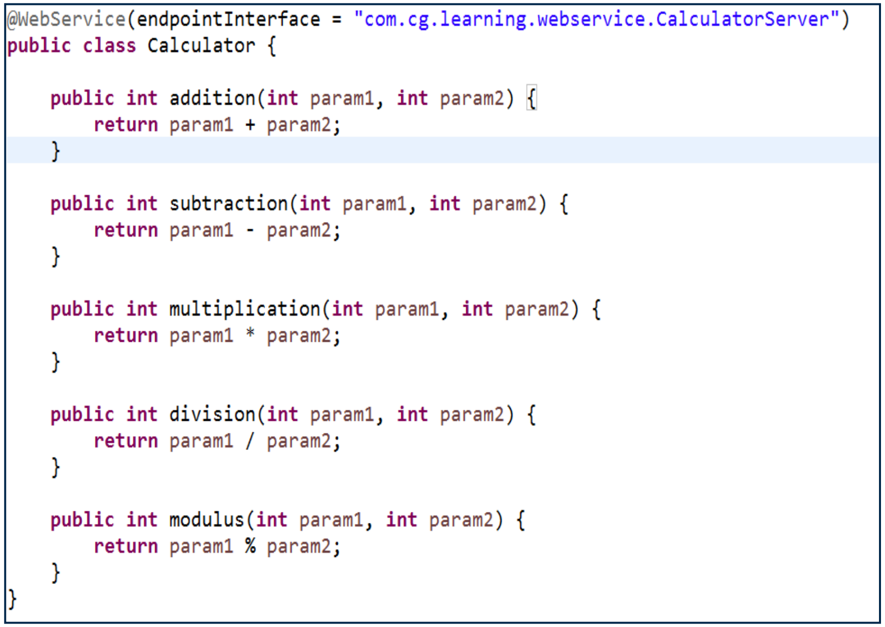
Creating JAX- WS Service

End Point Interface



```
@WebService
@SOAPBinding(style = Style.RPC)
public interface CalculatorServer {
    @WebMethod
    int addition(int param1, int param2);

    @WebMethod
    int subtraction(int param1, int param2);

    @WebMethod
    int multiplication(int param1, int param2);

    @WebMethod
    int division(int param1, int param2);

    @WebMethod
    int modulus(int param1, int param2);
}
```

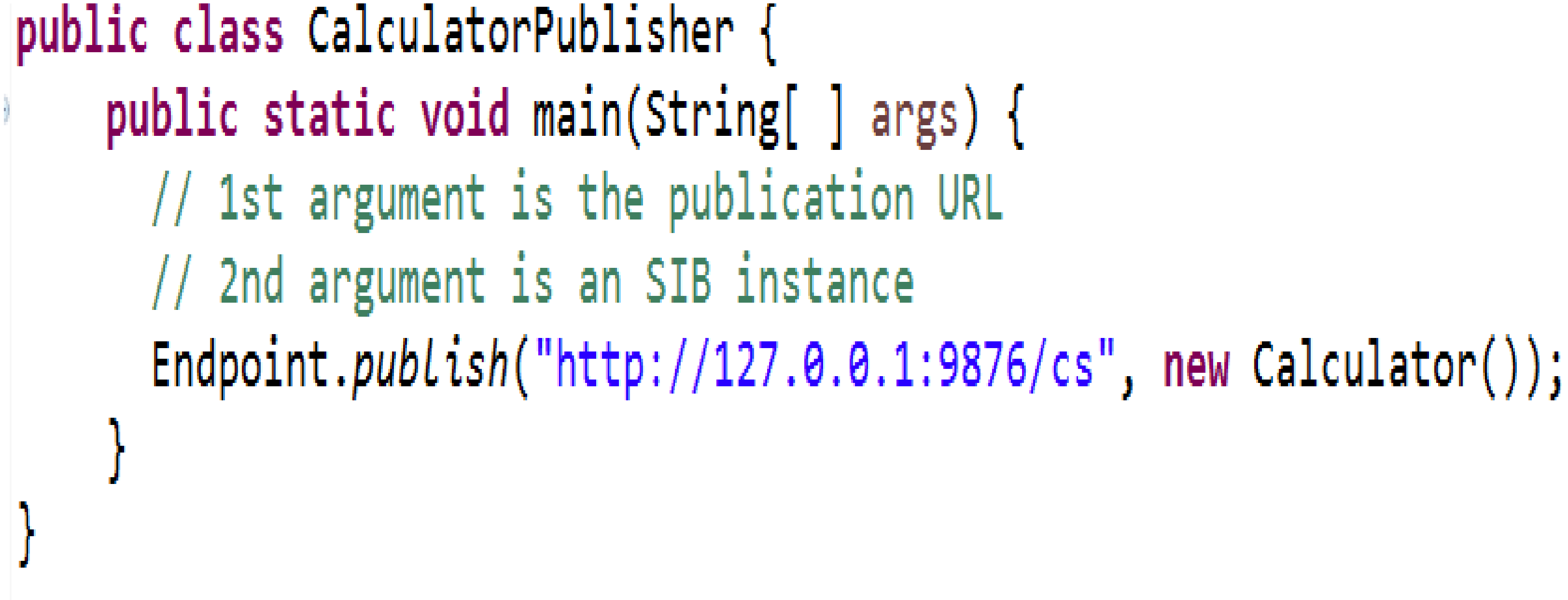
Implementation class



```
@WebService(endpointInterface = "com.cg.learning.webservice.CalculatorServer")
public class Calculator {

    public int addition(int param1, int param2) {
        return param1 + param2;
    }

    public int subtraction(int param1, int param2) {
        return param1 - param2;
    }

    public int multiplication(int param1, int param2) {
        return param1 * param2;
    }

    public int division(int param1, int param2) {
        return param1 / param2;
    }

    public int modulus(int param1, int param2) {
        return param1 % param2;
    }
}
```

Creating JAX – WS Service

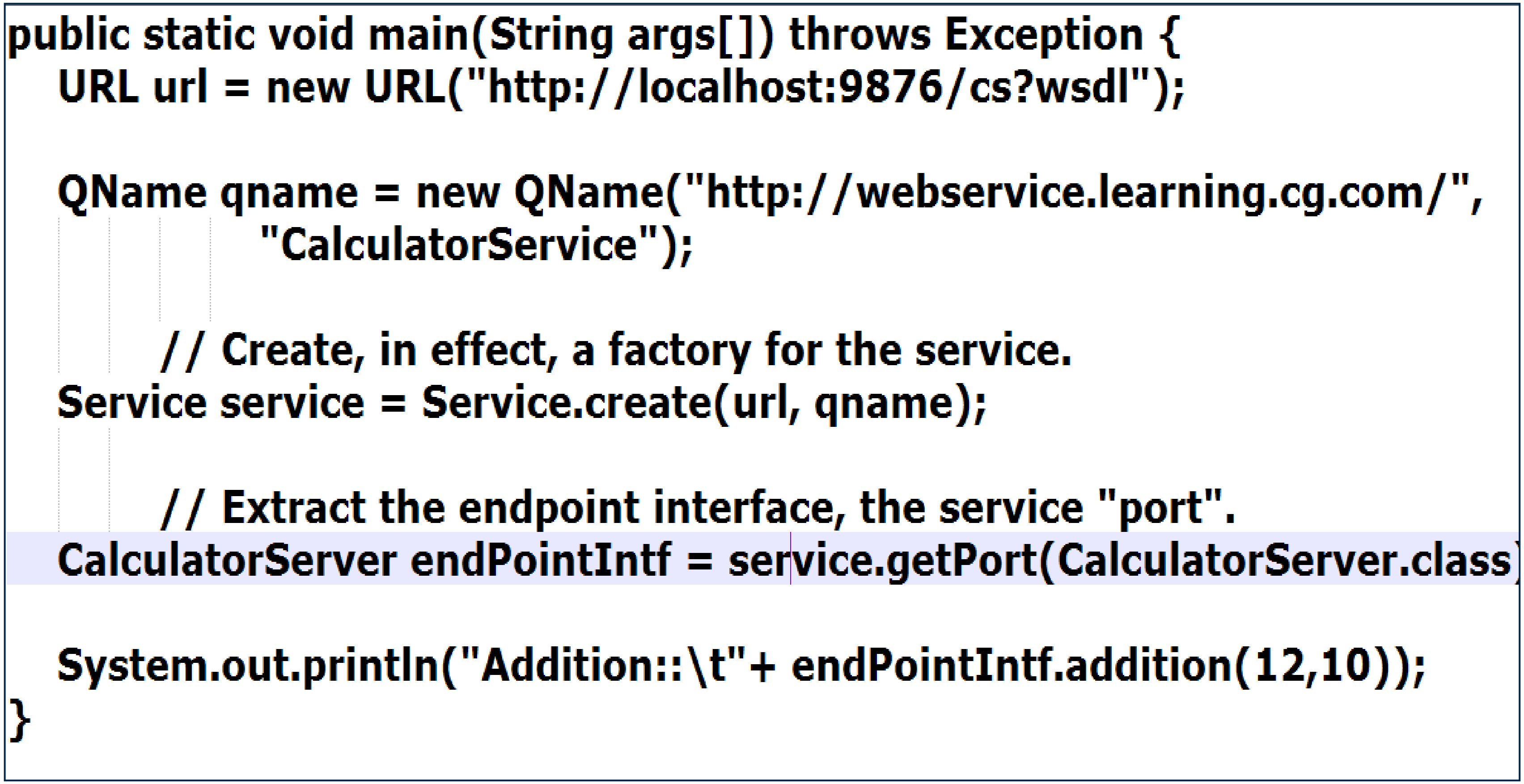
- The web service needs to be published so that it can await service requests
- Refer below screen shot for same



```
public class CalculatorPublisher {
    public static void main(String[ ] args) {
      // 1st argument is the publication URL
      // 2nd argument is an SIB instance
      Endpoint.publish("http://127.0.0.1:9876/cs", new Calculator());
    }
}
```

Consuming JAX – WS Service

- Refer below screen shot to consume a service:



```
public static void main(String args[]) throws Exception {
   URL url = new URL("http://localhost:9876/cs?wsdl");

   QName qname = new QName("http://webservice.learning.cg.com/",
               "CalculatorService");

         // Create, in effect, a factory for the service.
   Service service = Service.create(url, qname);

         // Extract the endpoint interface, the service "port".
   CalculatorServer endPointIntf = service.getPort(CalculatorServer.class

   System.out.println("Addition::\t"+ endPointIntf.addition(12,10));
}
```

Demo

- JAX-WS-Calculator

Summary

- We have so far learnt
  - What are JAX-WS
  - What are wsdl documents and their representation
  - What are soap messages and their request and response structure
  - How to create a JAX – WS service
  - How to consume a JAX – WS service

Lab

- Lab 1

Review Question

- Question 1: What gives information about the web service location and operation?
  - SOAP message structure
  - WSDL document
  - Service End Point interface
- Question 2: @WebMethod signals that each method is a service operation (True or False)
  - True
  - False
- Question 3:____________ is used to define protocol and data format?
  - Message
  - Port type
  - Binding