

## Category “wsrpc 2010/2011”

### Q1 original and generated WSDL

(15 pts)

In one or two sentences, describe the most important difference between the two WSDL documents that you compared in exercise 7.3.(b) and state why the change has been made by Axis.

*The WSDL generated by Axis contains the correct URL in the service definition.*

### Q2 Canvas GUI field decl

(5 pts)

After exercise 7.3.(d), locate the line in the original service implementation class that **declares** the field in which it stores a reference to the canvas GUI. Cut and paste this line into the box below.

(100) equals (case sensitive):

```
private static FrmMain GUI;
```

### Q3 why GUI static

(20 pts)

Explain why the GUI field located the previous question is declared as static.

*This class is instantiated with every request and no such instance is carried over to subsequent requests. The field is static because the GUI is needed across multiple requests.*

### Q4 SOAP tags

(10 pts)

In exercise 7.3.(e) you observed a SOAP envelope and its contents. State the names of all elements the SOAP envelope consists of. (Do not include any elements that relate to the XML types defined in files `LineCanvas.wsdl` or `Geom.xsd`.)

(100) equals (case sensitive):

```
envelope, body
```

(100) equals (case sensitive):

```
soapenv:envelope, soapenv:body
```

(100) equals (case sensitive):

```
body, envelope
```

(100) equals (case sensitive):

```
soapenv:body, soapenv:envelope
```

**Q5 DrawPoint**

(25 pts)

Copy and paste in the box below the complete code for class `DrawPoint` that you wrote in exercise 7.4.(d).

Your answer will be assessed according to how well it performs the task stated by the exercise text.

```
package points;

import java.rmi.RemoteException;

import wsrpc.canvas.LineCanvasProxy;
import wsrpc.geom.Line;
import wsrpc.geom.Point;
import wsrpc.geom.holders.PointHolder;

public class DrawPoints
{
    public static void main(String[] args) throws RemoteException
    {
        LineCanvasProxy canvas = new LineCanvasProxy();

        PointHolder topLeft = new PointHolder();
        PointHolder botRight = new PointHolder();
        canvas.getCanvasSize(topLeft, botRight);

        int centre_X = (topLeft.value.getX() + botRight.value.getX())/2;
        int centre_Y = (topLeft.value.getY() + botRight.value.getY())/2;

        Point point1 = new Point(centre_X + 5, centre_Y + 5);
        Point point2 = new Point(centre_X + 5, centre_Y - 5);
        Point point3 = new Point(centre_X - 5, centre_Y - 5);
        Point point4 = new Point(centre_X - 5, centre_Y + 5);

        canvas.drawLine(new Line(point1, point2));
        canvas.drawLine(new Line(point2, point3));
        canvas.drawLine(new Line(point3, point4));
        canvas.drawLine(new Line(point4, point1));

    }
}
```

**Q6 PointCanvas.wsdl**

(25 pts)

Copy and paste the complete XML text of the document `PointCanvas.wsdl` that you created in exercises 7.5.(a)–(h).

Your answer will be assessed according to how well it performs the task stated by the exercise text.

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="ptcanvas.wsrpc" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="PointCanvas"
  targetNamespace="ptcanvas.wsrpc">
  <wsdl:types>
    <xsd:schema targetNamespace="ptcanvas.wsrpc" xmlns:geom="geom.wsrpc">
      <xsd:import namespace="geom.wsrpc" schemaLocation="Geom.xsd" />
      <xsd:element name="drawPoint">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="point" type="geom:point" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="drawPointResponse">
        <xsd:complexType/>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="drawPointRequest">
    <wsdl:part element="tns:drawPoint" name="parameters" />
  </wsdl:message>
  <wsdl:message name="drawPointResponse">
    <wsdl:part element="tns:drawPointResponse" name="parameters" />
  </wsdl:message>
  <wsdl:portType name="PointCanvas">
    <wsdl:operation name="drawPoint">
      <wsdl:input message="tns:drawPointRequest" />
      <wsdl:output message="tns:drawPointResponse" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="PointCanvasSOAP" type="tns:PointCanvas">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="drawPoint">
      <soap:operation soapAction="ptcanvas.wsrpc/DrawPoint" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="PointCanvas">
    <wsdl:port binding="tns:PointCanvasSOAP" name="PointCanvasSOAP">
      <soap:address location="http://www.example.org/" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

Total of 100 pt.
------------------