

COP 4856 exercises

Consider the following (well-formed) XML expression and a corresponding XSLT fragment (next page), then draw a sketch how the result would look like on the screen.

```
<camera>
  <brand>Nikon</brand>
  <mp>24</mp>
  <sensor>apsc</sensor>
  <viewfinder>pentaprism</viewfinder>
</camera>
```

```
<xsl:template match="/">
  <html>
    <head>
      <title>I want this for Christmas!</title>
    </head>
    <body>
      <h1>I want this for Christmas!</h1>
      <table border="1" cellpadding="6"
        cellspacing="3" width="30%" >
        <thead>
          <tr>
            <td>Brand</td>
            <td>Megapixel</td>
            <td>Sensor</td>
            <td>Viewfinder</td>
          </tr>
        </thead>
        <tr>
          <td><xsl:value-of select="camera/brand" /></td>
          <td><xsl:value-of select="camera/mp" /></td>
          <td><xsl:value-of select="camera/sensor" /></td>
          <td><xsl:value-of select="camera/viewfinder" /></td>
        </tr>
      </table>
    </body>
  </html>
</xsl:template>
```

Sketch goes here:

Rewrite this code with a prepared statement:

```
// assume bookId is set with a primary key for a Book
Statement stmt = conn.createStatement();
String sql = "SELECT * from Books " +
    "WHERE id = \"" + bookId + "\"";
ResultSet rs = stmt.executeQuery(sql);
```

Provide annotations for JPA and JAXB to this POJO (you might have to add one field!). Assume that JAXB annotations are handled by field names.

```
public class Book {
```

```
    String author;
```

```
    String title;
```

```
    int pages;
```


These are the headers for the doGet and doPost methods of a Servlet.

```
protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException,
    IOException
```

```
protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException,
    IOException
```

Write the important parts of a Servlet that

- reacts to both GET and POST requests the same way
- it extracts the "message" parameter value out of the request
- it produces HTML like this

```
<html>
  <head><title>Message of the day</title></head>
  <body>
    <h1>Message of the day</h1>
    <p>XXXXXXXXXX</p>
  </body>
</html>
```

where XXXXXXXXXXXX is replaced by the parameter value (Don't worry about HTML indentation!)