

19. Save your changes to the productlist.xml file.
20. Use your XSLT 2.0 processor to transform the productlist.xml file using the productlist.xsl style sheet, storing the result in the **productlist.html** file.

APPLY

Case Problem 1

Data Files needed for this Case Problem: alltxt.xml, horizonstxt.xml, +6 XML files, +1 CSS file, +1 PNG file

Horizons TechNet Lucy di Prima is the employment officer at Horizons TechNet, a technology firm located in Provo, Utah. Periodically, she has to compile reports on employee compensation by gender, where compensation is defined as the sum of each employee's salary, bonuses, and any sales commissions.

Lucy has received sample employee data from several departments in the form of XML documents. She wants to compile these XML documents into a single XML document and then she wants to display the total compensation for each employee and calculate the average compensation across all employees by gender. Figure 8-44 shows a preview of her report.

Figure 8-44 Horizons TechNet employment report

|  Horizons TechNet Forward-Looking Solutions for Integrated Technology | | | | |
|--|------------|-------------|-----------------|--------------------|
| Employment Report | | | | |
| Gender | | F | | |
| Employees | | 22 | | |
| Average Compensation | | \$72,122.73 | | |
| ID | Department | Title | Education Level | Total Compensation |
| 10 | A00 | President | 18 | \$133,300.00 |
| 520 | A00 | Sales Rep | 18 | \$119,000.00 |
| 30 | C01 | Manager | 20 | \$96,800.00 |
| 40 | A00 | Manager | 18 | \$93,600.00 |
| 70 | D21 | Manager | 16 | \$91,500.00 |
| 220 | D11 | Designer | 18 | \$75,500.00 |
| 610 | D11 | Designer | 18 | \$75,500.00 |
| 90 | E11 | Manager | 16 | \$75,300.00 |
| 140 | C01 | Analyst | 18 | \$72,000.00 |
| 550 | C01 | Analyst | 18 | \$72,000.00 |
| 270 | D21 | Clerk | 15 | \$69,200.00 |
| 280 | E11 | Programmer | 17 | \$66,400.00 |
| 640 | E11 | Programmer | 17 | \$66,400.00 |
| 780 | E21 | Field Rep | 16 | \$64,400.00 |
| 760 | E21 | Field Rep | 14 | \$64,300.00 |
| 130 | C01 | Analyst | 16 | \$60,300.00 |
| 790 | E21 | Field Rep | 14 | \$56,800.00 |
| 160 | D11 | Designer | 17 | \$56,200.00 |
| 180 | D11 | Designer | 17 | \$54,200.00 |
| 260 | D21 | Clerk | 16 | \$43,600.00 |
| 310 | E11 | Programmer | 12 | \$40,200.00 |
| 660 | E11 | Programmer | 12 | \$40,200.00 |

Lucy has asked you to use XSLT 2.0 and XPath 2.0 to write this report. Complete the following:

1. Using your text editor, open the **alltxt.xml** and **horizonstxt.xml** files from the xml08 ► case1 folder. Enter **your name** and the **date** in the comment section of each file, and save them as **alldepartments.xml** and **horizons.xml**, respectively.
2. Go to the **alldepartments.xml** file in your text editor. The purpose of this template is to create a single XML document listing all of the employees from each department. First, create a template named **getEmployees**.
3. Within the getEmployees template, create a variable named **depts** containing a sequence of the following text strings representing the department codes for Lucy's sample data: 'a00', 'c01', 'd11', 'd21', 'e11', and 'e21'.

4. After the line to create the *depts* variable, create the **departments** element.
5. Within the departments element, insert a for-each loop that loops through each entry in the *depts* sequence.
6. For each entry in the *depts* sequence do the following:
 - a. Create a variable named **currentDept** equal to the current item in the *depts* sequence.
 - b. Create an element named **department** with an attribute named **deptID** whose value is equal to the value of the *currentDept* variable.
 - c. Use the `doc()` function to reference the "deptcurrent.xml" file, where *current* is the value of the *currentDept* variable. (Hint: Use the `concat()` function to combine the text strings for "dept", the *currentDept* variable, and the text string ".xml".)
 - d. Use the `copy-of` element to copy the contents of the employees element and its descendants to the department element.
7. Save your changes to the file and then use your XSLT 2.0 processor to generate the result document **horizons.xml** by applying the getEmployees template within the alldepartments.xsl style sheet.
8. Next, you'll display the employee data in an HTML file. Go to the **horizons.xsl** file in your text editor.
9. Directly below the `<h1>Employment Report</h1>` tag, insert a `for-each-group` element that selects all of the employee elements in the source document and groups them by the gender element.
10. Within the `for-each-group` loop, write the following code to the result document:

```
<table id="summary">
  <tr>
    <th>Gender</th>
    <td>key</td>
  </tr>
  <tr>
    <th>Employees</th>
    <td>count</td>
  </tr>
  <tr>
    <th>Average Compensation</th>
    <td>average</td>
  </tr>
</table>
```

where *key* is the value of the current grouping key, *count* is the number of items in the current group, and *average* uses the XPath 2.0 `avg()` function to calculate the average value of the sum of the salary + bonus + commission elements within the current group. Format the average compensation as a currency value.

11. Next, write the following HTML table to the result document:

```
<table id="emptable">
  <tr>
    <th>ID</th>
    <th>Department</th>
    <th>Title</th>
    <th>Education Level</th>
    <th>Total Compensation</th>
  </tr>
  <current group template>
</table>
```

where *current group template* is application of the template for the current group, displaying information on each employee in a separate table row sorted in descending order of the sum of salary + bonus + commission. (Hint: Be sure to specify the data type for the `sort` element as `xs:number`.)

12. Create the template for the employee element. The purpose of this template is to display information on each employee. Within the template, write the following table row:

```
<tr>
  <td>empID</td>
  <td>department</td>
  <td>title</td>
  <td>edLevel</td>
  <td>compensation</td>
</tr>
```

where *empID* is the value of the empID attribute, *department* is the value of the department element, *title* is the value of the title element, *edLevel* is the value of the edLevel element, and *compensation* is the sum of the salary, bonus, and commission elements.

13. Save your changes to the file.
14. Use your XSLT 2.0 processor with the horizons.xml source document and the horizons.xsl style sheet to generate an output file named **horizons.html**.
15. Open the **horizons.html** file in your web browser and verify that its contents and layout resemble that shown in Figure 8-44.


Case Problem 2

Data Files needed for this Case Problem: boise.txt, readtabtxt.xsl. +1 CSS file, +1 XSLT file, +1 PNG file

Idaho Climate Research Council Paul Rao works for the Idaho Climate Research Council, a research institute located in Boise, Idaho, that is investigating issues surrounding climate and weather. Paul has the job of developing a report summarizing temperature, precipitation, wind, and other weather factors taken from a weather research station outside of Boise.

His first job is to summarize daily temperature readings from 1995 to 2013. As an initial step, he wants to create a table showing average, maximum, and minimum temperature values as summary statistics for each month. He also wants his table to calculate the range of temperatures (equal to the maximum value minus the minimum value). Figure 8-45 shows a preview of the completed report.

Figure 8-45 Boise, Idaho temperature summary



Idaho Climate Research Council

Boise, Idaho

Annual Temperature Summary

| Month | Average | Minimum | Maximum | Range |
|-----------|---------|---------|---------|-------|
| January | 32.12 | 6.0 | 54.0 | 48.0 |
| February | 36.52 | 7.3 | 53.1 | 45.8 |
| March | 44.19 | 26.7 | 62.8 | 36.1 |
| April | 50.07 | 35.4 | 75.7 | 40.3 |
| May | 58.97 | 40.4 | 80.2 | 39.8 |
| June | 67.57 | 44.0 | 88.6 | 44.6 |
| July | 78.20 | 56.8 | 94.2 | 37.4 |
| August | 75.97 | 54.2 | 89.2 | 35.0 |
| September | 66.23 | 46.2 | 85.1 | 38.9 |
| October | 52.42 | 23.4 | 79.1 | 55.7 |
| November | 40.83 | 15.2 | 63.1 | 47.9 |
| December | 32.10 | 3.1 | 55.8 | 52.7 |