

Section1

Exercise 12.1.1

a) //PC/RAM

<RAM>1024</RAM>

<RAM>512</RAM>

<RAM>2048</RAM>

<RAM>1024</RAM>

b) //@price

2114

995

1150

2500

959

649

3673

239

100

200

c) //Printer

<Printer model="3002" price="239">...</Printer>

<Printer model="3006" price="100">...</Printer>

<Printer model="3007" price="200">...</Printer>

d) /Products/Maker[Printer/Type = "laser"]/@name

E

H

e) /Products/Maker[PC or Laptop]/@name

A

E

f) //PC[HardDisk >= 200]/@model

1001

1002

g) /Products/Maker[count(PC) >= 2]/@name

A

E

Exercise 12.1.2

a) //Ship/@name

Kongo

Hiei

Kirishima

Hurana

North Carolina

Washington

Tennessee

California

King George V

Prince of Wales

Duke of York

Howe

Anson

b) /Ships/Class[@displacement > 35000]

```
<Class name="North Carolina" type = "bb" country="USA" numGuns="9" displacement="37000">...</Class>
```

C) //Ship[@launched < 1917]

```
<Ship name="Kongo" launched="1913" />
```

```
<Ship name="Hiei" launched="1914" />
```

```
<Ship name="Kirishima" launched="1915">...</Ship>
```

```
<Ship name="Hurana" launched="1915" />
```

D) //Ship [Battle/@outcome = "sunk"]

```
<Ship name="Kirishima" launched="1915">...</Ship>
```

```
<Ship name="Prince of Wales" launched="1941">...</Ship>
```

E) /Ships/Class/Ship [@name = ../@name]/@launched

1913

1941

1920

1940

F) /Ships/Class/Ship [Battle]/@name

Kirishima

Washington

Tennessee

California

Prince of Wales

Duke of York

g) /Ships/Class/Ship [count(Battle) >= 2]

```
<Ship name="Prince of Wales" launched="1941">...</Ship>
```

Section 2

Exercise 12.2.1

a)

```
let $products := doc("products.xml")
for $printer in $products//Printer
where $printer/@price lt '100'
return $printer
```

b)

```
<CheapPrinters>
{
let $products := doc("products.xml")
for $printer in $products//Printer
where $printer/@price lt '100'
return $printer
}
</CheapPrinters>
```

c)

```
let $products := doc("products.xml")
for $maker in $products//Maker
where $maker/Printer and $maker/Laptop
return <Maker name="{ $maker/@name}" />
```

D)

```
let $products := doc("products.xml")
for $maker in $products//Maker
where count($maker/PC[Speed > 3.00]) >= 2
return <Maker name="{ $maker/@name}" />
```

E)

```
let $products := doc("products.xml")
for $maker in $products//Maker[PC]
where every $pc in $maker/PC satisfies $pc/@price <= 1000
return <Maker>{ $maker/@name}</Maker>
```

note: without [PC] from line 2 will qualify makers that do not produce any PC

F)

```
let $products := doc("products.xml")
for $laptop in $products//Laptop
return
<Lap-
top><Model>{data($laptop/@model)}</Model><Maker>{data($laptop/../@name)}</Maker></
Laptop>
```

Exercise 12.2.2

a)

```
let $ships := doc("Ships.xml")
for $class in $ships/Ships/Class
where $class/@numGuns >= 10
return data($class/@name)
```

b)

```
let $ships := doc("Ships.xml")
for $class in $ships/Ships/Class
where $class/@numGuns >= 10
return data($class/Ship/@name)
```

c)

```
let $ships := doc("Ships.xml")
for $ship in $ships//Ship[Battle]
where $ship/Battle/@outcome = "sunk"
return data($ship/@name)
```

d)

```
let $ships := doc("Ships.xml")
for $class in $ships//Class
where count($class/Ship) >= 3
return data($class/@name)
```

e)

```
let $ships := doc("Ships.xml")
for $class in $ships//Class
where every $ship in $class/Ship satisfies count($ship/Battle) = 0
return data($class/@name)
```

or

```
let $ships := doc("Ships.xml")
for $class in $ships//Class
where count($class//Battle) = 0
return data($class/@name)
```

f)

```
let $ships := doc("Ships.xml")
for $class in $ships//Class
where some $ship in $class/Ship satisfies count($class/Ship[@launched = $ship/@launched]) >= 2
return data($class/@name)
```

```

g)
let $ships := doc("Ships.xml")
let $battles := distinct-values(data($ships//Battle))
for $battle in $battles
return <Battle name = "{$battle}">
{
    for $ship in $ships//Ship[Battle]
    where data($ship/Battle) = $battle
    return <Ship name = "{data($ship/@name)}" />
}</Battle>

```

or

```

let $ships := doc("Ships.xml")
let $battles := distinct-values(data($ships//Battle))
for $battle in $battles
return <Battle name = "{$battle}">
{
    for $ship in $ships//Ship[data(Battle) = $battle]
    return <Ship name = "{data($ship/@name)}" />
}</Battle>

```

Exercise 12.2.3

```

let $stars := doc("stars.xml")
for $s in $stars/Stars/Star
where $s/Address[data(Street) eq "123 Maple St."] and $s/Address[data(City) eq "Malibu"]
return data($s/Name)

```

Exercise 12.2.4

It is not possible to have expression E and F such that the expression every \$x in E satisfies F is true, but some \$x in E satisfies F is false. The existential qualifier ‘some’ is a subset of the universal qualifier ‘every.’ If all values in E satisfy F, then there is certainly at least one value in E that satisfies F.

Section 3

Exercise 12.3.1

a)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
    <head>
      <title>Manufacturers</title>
    </head>
    <body>
      <h1>Manufacturers</h1>
      <ol>
        <xsl:for-each select="Products/Maker">
          <li>
            <xsl:value-of select="@name" />
          </li>
        </xsl:for-each>
      </ol>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

b)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="/">
  <html>
    <head>
      <title></title>
    </head>
    <body>
      <table border="1">
        <tbody>
          <tr>
            <th>Model</th>
            <th>Price</th>
          </tr>
          <xsl:for-each select="//PC">
            <tr>
              <td>
                <xsl:value-of select="@model" />
              </td>
              <td>
                <xsl:value-of select="@price" />
              </td>
            </tr>
          </xsl:for-each>
        </tbody>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

c)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="/">
  <html>
    <head>
      <title></title>
    </head>
    <body>
      <b>Laptops</b>
      <table border="1">
        <tbody>
          <tr>
            <th>Model</th>
            <th>Price</th>
            <th>Speed</th>
            <th>Ram</th>
          </tr>
          <xsl:for-each select="//Laptop">
            <tr>
              <td><xsl:value-of select="@model" /></td>
              <td><xsl:value-of select="@price" /></td>
              <td><xsl:value-of select="Speed" /></td>
              <td><xsl:value-of select="RAM" /></td>
            </tr>
          </xsl:for-each>
        </tbody>
      </table>
      <b>PC's</b>
      <table border="1">
        <tbody>
          <tr>
            <th>Model</th>
            <th>Price</th>
            <th>Speed</th>
            <th>Ram</th>
          </tr>
          <xsl:for-each select="//PC">
            <tr>
              <td><xsl:value-of select="@model" /></td>
              <td><xsl:value-of select="@price" /></td>
              <td><xsl:value-of select="Speed" /></td>
              <td><xsl:value-of select="RAM" /></td>
            </tr>
          </xsl:for-each>
        </tbody>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```


d)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="/">
  <xsl:element name="PCs">
    <xsl:for-each select="//PC">
      <xsl:element name="PC">
        <xsl:attribute name="model">
          <xsl:value-of select="@model" />
        </xsl:attribute>
        <xsl:attribute name="price">
          <xsl:value-of select="@price"/>
        </xsl:attribute>
        <xsl:attribute name="speed">
          <xsl:value-of select="Speed"/>
        </xsl:attribute>
        <xsl:attribute name="ram">
          <xsl:value-of select="RAM"/>
        </xsl:attribute>
      </xsl:element>
    </xsl:for-each>
  </xsl:element>
</xsl:template>
</xsl:stylesheet>
```

e)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="/">
  <xsl:element name="Products">
    <xsl:for-each select="//PC">
      <xsl:element name="Product">
        <xsl:attribute name="type">
          <xsl:value-of select="name()" />
        </xsl:attribute>
        <xsl:attribute name="maker">
          <xsl:value-of select="../@name"/>
        </xsl:attribute>
        <xsl:attribute name="model">
          <xsl:value-of select="@model"/>
        </xsl:attribute>
        <xsl:attribute name="price">
          <xsl:value-of select="@price"/>
        </xsl:attribute>
      </xsl:element>
    </xsl:for-each>
    <xsl:for-each select="//Laptop">
      <xsl:element name="Product">
        <xsl:attribute name="type">
          <xsl:value-of select="name()" />
        </xsl:attribute>
        <xsl:attribute name="maker">
          <xsl:value-of select="../@name"/>
        </xsl:attribute>
      </xsl:element>
    </xsl:for-each>
  </xsl:element>
</xsl:template>
</xsl:stylesheet>
```

```

        </xsl:attribute>
        <xsl:attribute name="model">
            <xsl:value-of select="@model"/>
        </xsl:attribute>
        <xsl:attribute name="price">
            <xsl:value-of select="@price"/>
        </xsl:attribute>
    </xsl:element>
</xsl:for-each>
<xsl:for-each select="//Printer">
    <xsl:element name="Product">
        <xsl:attribute name="type">
            <xsl:value-of select="name()" />
        </xsl:attribute>
        <xsl:attribute name="maker">
            <xsl:value-of select="../@name"/>
        </xsl:attribute>
        <xsl:attribute name="model">
            <xsl:value-of select="@model"/>
        </xsl:attribute>
        <xsl:attribute name="price">
            <xsl:value-of select="@price"/>
        </xsl:attribute>
    </xsl:element>
</xsl:for-each>
</xsl:element>
</xsl:template>

```

```

f)
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:output method="text"/>
    <xsl:template match="/">
        <xsl:text disable-output-escaping="yes">
            \begin{table}[ht]
            \caption{PCs}
            \begin{tabular}{c c}
            Model & Price \\
            \hline
        </xsl:text>
        <xsl:for-each select="//PC">
            <xsl:value-of select="@model" />
            <xsl:text> & </xsl:text>
            <xsl:value-of select="@price" />
            <xsl:text> \\ </xsl:text>
        </xsl:for-each>
        <xsl:text>
            \hline
            \end{tabular}
            \end{table}
        </xsl:text>
    </xsl:template>
</xsl:stylesheet>

```

Exercise 12.3.2

a)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="/">
  <html>
    <head>
      <title>Ships</title>
    </head>
    <body>
      <xsl:for-each select="Ships/Class">
        <h1>
          <xsl:value-of select="@name" />
        </h1>
        <table border="1">
          <tbody>
            <tr>
              <th>Name</th>
              <th>Launched</th>
            </tr>
            <xsl:for-each select="Ship">
              <tr>
                <td><xsl:value-of select="@name"/></td>
                <td><xsl:value-of select="@launched"/></td>
              </tr>
            </xsl:for-each>
          </tbody>
        </table>
      </xsl:for-each>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

b)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="/">
  <html>
    <head>
      <title></title>
    </head>
    <body>
      <xsl:text>&lt;Losers&gt;</xsl:text>
      <br></br>
      <xsl:for-each select="//Battle[@outcome = 'sunk']">
        <xsl:text>&lt;Ship&gt;</xsl:text>
        <xsl:value-of select="../@name"/>
        <xsl:text>&lt;/Ship&gt;</xsl:text>
        <br></br>
      </xsl:for-each>
      <xsl:text>&lt;/Losers&gt;</xsl:text>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

```

        </body>
    </html>
</xsl:template>
</xsl:stylesheet>

```

c)

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="/">
    <xsl:element name="Ships">
        <xsl:for-each select="//Ship">
            <xsl:element name="Ship">
                <xsl:attribute name="name">
                    <xsl:value-of select="@name"/>
                </xsl:attribute>
                <xsl:attribute name="class">
                    <xsl:value-of select="../@name"/>
                </xsl:attribute>
                <xsl:attribute name="country">
                    <xsl:value-of select="../@country"/>
                </xsl:attribute>
                <xsl:attribute name="numGuns">
                    <xsl:value-of select="../@numGuns"/>
                </xsl:attribute>
            </xsl:element>
        </xsl:for-each>
    </xsl:element>
</xsl:template>
</xsl:stylesheet>

```

d)

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="/">
    <xsl:element name="Ships">
        <xsl:for-each select="//Ship[Battle]">
            <xsl:element name="Ship">
                <xsl:attribute name="name">
                    <xsl:value-of select="@name"/>
                </xsl:attribute>
                <xsl:attribute name="class">
                    <xsl:value-of select="../@name"/>
                </xsl:attribute>
                <xsl:attribute name="country">
                    <xsl:value-of select="../@country"/>
                </xsl:attribute>
                <xsl:attribute name="numGuns">
                    <xsl:value-of select="../@numGuns"/>
                </xsl:attribute>
            </xsl:element>
        </xsl:for-each>
    </xsl:element>
</xsl:template>
</xsl:stylesheet>

```

```

    </xsl:element>
</xsl:template>
</xsl:stylesheet>

```

e)

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="/">
<xsl:element name="Ships">
  <xsl:for-each select="Ships/Class">
    <xsl:element name="Class">
      <xsl:attribute name="name">
        <xsl:value-of select="@name"/>
      </xsl:attribute>
      <xsl:attribute name="type">
        <xsl:value-of select="@type"/>
      </xsl:attribute>
      <xsl:attribute name="country">
        <xsl:value-of select="@country"/>
      </xsl:attribute>
      <xsl:attribute name="numGuns">
        <xsl:value-of select="@numGuns"/>
      </xsl:attribute>
      <xsl:attribute name="bore">
        <xsl:value-of select="@bore"/>
      </xsl:attribute>
      <xsl:attribute name="displacement">
        <xsl:value-of select="@displacement"/>
      </xsl:attribute>
      <xsl:for-each select="Ship">
        <xsl:element name="Ship">
          <xsl:attribute name="name">
            <xsl:value-of select="@name"/>
          </xsl:attribute>
          <xsl:attribute name="launched">
            <xsl:value-of select="@launched"/>
          </xsl:attribute>
          <xsl:for-each select="Battle">
            <xsl:element name="Battle">
              <xsl:attribute name="name">
                <xsl:value-of select="."/>
              </xsl:attribute>
              <xsl:attribute name="outcome">
                <xsl:value-of select="@outcome"/>
              </xsl:attribute>
            </xsl:element>
          </xsl:for-each>
        </xsl:element>
      </xsl:for-each>
    </xsl:element>
  </xsl:for-each>
</xsl:element>
</xsl:template>
</xsl:stylesheet>

```