CSC343 Worksheet 10 Solution

June 29, 2020

1. a) /Products/Maker/PC/RAM

   Notes:

   - XPATH and Selecting Nodes
     - nodename
       * Selects all nodes with the name "nodename"
     - /
       * Selects from the root node
     - //
       * Selects node in the document from the current node that match the selection no matter where they are
     - .
       * Select the current node
     - ..
       * Selects the parent of the current node
     - @
       * Selects attributes

   Example:

   ```
   1    /StarMovieData/Star//City
   2
   ```

   - selects all City element in

     <StarMovieData>
       <Star>
         Here :)
       </Star>
     </StarMovieData>
   - Wildcards *

    – Is used to say 'any tag'

Example:

```
1      /StarMovieData/*/@*
2
```

    – '@*' means any attributes
    – '*' means any tag
- Context of Expressions
  – [...] means that exists or there exists
  – [*integer*] selects ith child of its parent
  – [*Tag*] selects elements that have one or more sublements with 'Tag'
  – [*Attribute*] selects elements that have attribute 'Attribute'

Example:

```
1      /StarMovieData/Star[//City = "Malibu"]/Name
2
```

    ∗ Means select all Star Name that contains City with value 'Malibu'

Example:
```
1      /Movies/Movie/Version[1]/@year
2
```

    ∗ Returns value of 'year' attribute of first 'Version' tag in 'Movie'
    ∗ e.g. 1933 and 1984

```
1)   <? xml version="1.0" encoding="utf-8" standalone="yes" ?>
2)   <Movies>
3)       <Movie title = "King Kong">
4)           <Version year = "1933">
5)               <Star>Fay Wray</Star>
6)           </Version>
7)           <Version year = "1976">
8)               <Star>Jeff Bridges</Star>
9)               <Star>Jessica Lange</Star>
10)          </Version>
11)          <Version year = "2005" />
12)      </Movie>
13)      <Movie title = "Footloose">
14)          <Version year = "1984">
15)              <Star>Kevin Bacon</Star>
16)              <Star>John Lithgow</Star>
17)              <Star>Sarah Jessica Parker</Star>
18)          </Version>
19)      </Movie>
20)  </Movies>
```

Result of
/Movies/Movie/Version[1]/@year

Example 2:

```
1      /Movies/Movie/Version
2
```

∗ Returns all 'Version' tag in 'Movie'

∗ e.g. lines 4 through 6, 7 through 10, line 11, lines 14 through 18

```
1)   <? xml version="1.0" encoding="utf-8" standalone="yes" ?>
2)   <Movies>
3)       <Movie title = "King Kong">
4)           <Version year = "1933">
5)               <Star>Fay Wray</Star>
6)           </Version>
7)           <Version year = "1976">
8)               <Star>Jeff Bridges</Star>
9)               <Star>Jessica Lange</Star>
10)          </Version>
11)          <Version year = "2005" />
12)      </Movie>
13)      <Movie title = "Footloose">
14)          <Version year = "1984">
15)              <Star>Kevin Bacon</Star>
16)              <Star>John Lithgow</Star>
17)              <Star>Sarah Jessica Parker</Star>
18)          </Version>
19)      </Movie>
20)  </Movies>
```

Result of
/Movies/Movie/Version

Example 3:

```
1      /Movies/Movie/Version[Star]
2
```

∗ Selects all 'Version' tag with one or more 'Star' tag inside

∗ e.g lines 4 through 6, 7 through 10, 14 through 18

```
1)   <? xml version="1.0" encoding="utf-8" standalone="yes" ?>
2)   <Movies>
3)      <Movie title = "King Kong">
4)         <Version year = "1933">
5)            <Star>Fay Wray</Star>
6)         </Version>
7)         <Version year = "1976">
8)            <Star>Jeff Bridges</Star>
9)            <Star>Jessica Lange</Star>
10)        </Version>
11)        <Version year = "2005" />
12)     </Movie>
13)     <Movie title = "Footloose">
14)        <Version year = "1984">
15)           <Star>Kevin Bacon</Star>
16)           <Star>John Lithgow</Star>
17)           <Star>Sarah Jessica Parker</Star>
18)        </Version>
19)     </Movie>
20)  </Movies>
```

Result of /Movies/Movie/Version[Star]

  b) /Products/Maker/*/@price

  c) /Products/Maker/Printer

  d) /Products/Maker[/Printer/Type/text() = 'ink-jet']

  > **Correct Solution:**
  > /Products/Maker[Printer/Type/text() = 'ink-jet']

  e) /Products/Maker[/PC | /Laptops]

  Notes:

  - XPATH and OR
    - **Syntax:** (*xpath expression 1*) | (*xpath expression 2*) [1]

  References:

  1) Stack Overflow, XPath OR operator for different nodes, link

  f) /Products/Maker/*[HardDisk/text() > 200]/@model

2. a) /Ships/Class/Ship/@name

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

   c) /Ships/Class/Ship[@launched < 1917]

   d) /Ships/Class/Ship[Battle/@outcome = 'sunk']/@name

   e) /Ships[Class/@name = Class/Ship/@name]/Ship/@name

   f) /Ships[Class/@name = Class/Ship/@name]/Class/Ship/@launched