University of Toronto
CSC343, Summer 2019

XML and DTDs: Solutions

XML document

The following XML document is well-formed

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE DATA SYSTEM "blah.dtd">
<DATA>
   <ANIMALS>
      <COW name = "Snowball" home = "Red Barn Acres"/>
      <HEN name = "Henrietta" birthdate = "120724" home = "Red Barn Acres"/>
   </ANIMALS>
   <FARMS>
      <FARM owner = "Old MacDonald" name = "Red Barn Acres">
         A picturesque hobby farm on 150 acres
      </FARM>
      <FARM owner = "Wiloughby Clive" name = "Weatherby Farm">
      </FARM>
      <FARM owner = "Egg Masters Inc" name = "Farm 23">
         A factory farm on a quarter section of land
      </FARM>
      <FARM owner = "Wiloughby Clive" name = "">
      </FARM>
   </FARMS>
</DATA>
```

Questions

1. Let's recap some terminology.

   (a) What is the root element of this XML file?

   Solution: DATA

   (b) Name three tags in the XML file.

   Solution: any 3 of DATA, ANIMALS, COW, HEN, FARMS, FARM

   (c) Name an empty element in the XML file.

   Solution: HEN or COW

   (d) Name three attributes in the XML file.

   Solution: name, home, owner, birthdate

   (e) Is this XML document well-formed? Explain.

   Solution: yes, it meets all the criteria (tree w/ proper nesting etc.)

   (f) Is this XML document valid? Explain.

   Solution: We can't tell because we don't know what is in "blah.dtd".

2. Suppose this XML is valid with respect to its DTD. For each of the following rules, circle Yes or No to indicate whether it could be part of that DTD.

`<!ELEMENT DATA (ANIMALS+, FARMS*)>`	[✓ Yes]	No
`<!ATTLIST COW birthdate CDATA #IMPLIED>`	[✓ Yes]	No
`<!ELEMENT HEN EMPTY>`	[✓ Yes]	No
`<!ELEMENT ANIMALS (HEN | COW)*>`	[✓ Yes]	No

3. Write a DTD definition for element FARMS that accepts the above instance document and enforces this rule: There must be at least four farms in the file. If this is not possible, explain why

   Solution: `<!ELEMENT FARMS (FARM, FARM, FARM, FARM+)>`

4. Write a DTD definition for attribute `name` of element `FARM` that accepts the above instance document and enforces this rule: No two farms have the same name. If this is not possible, explain why.

   Solution: It's not possible because blanks are not allowed in an ID attribute.

5. Suppose our DTD includes a rule defining an element called `DOG` — we just didn't happen to engage it in this XML file. Write a new DTD rule for element `ANIMALS` that enforces the following: there must be at least one `DOG`, and the order of the animals is all `DOG`s first, then `HEN`s and `COW`s in any order.

   Solution: `<!ELEMENT ANIMALS (DOG+, (HEN | COW)*) >`