

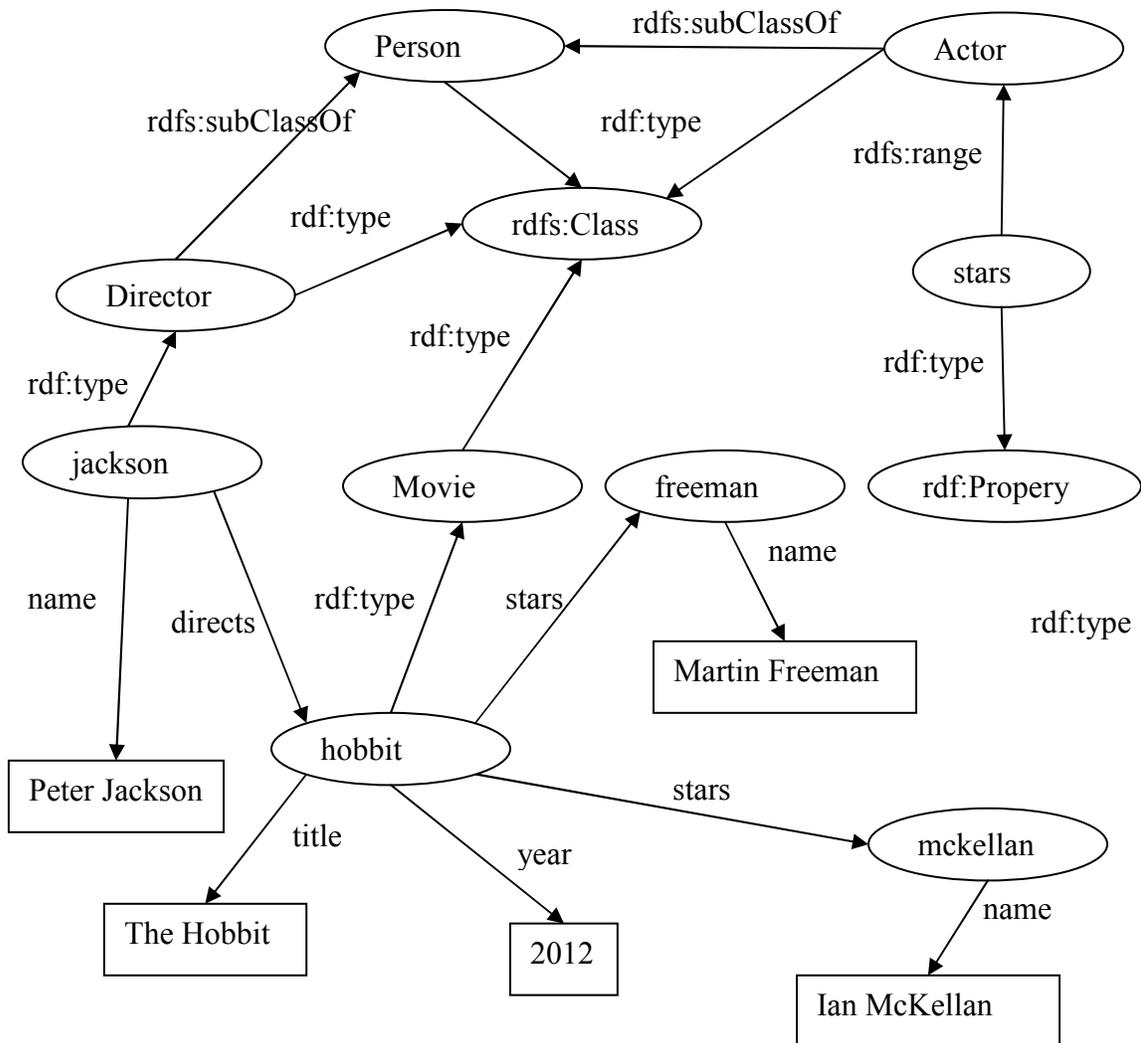
Homework #1: RDF

The following exercises are due at the beginning of class on Tuesday, Feb. 5. There are two sections: written exercises and electronic exercises. Note, one exercise requires CSE 428 students to do more work, and another is only assigned to them. CSE 398 students can receive extra credit for doing the work expected of CSE 428 students. This will count for 10% of your overall grade.

Written Exercises:

The exercises in this section should be completed and turned in on paper.

1. [10 pts.] Translate the following RDF Graph into the Turtle syntax. For nodes and arcs labeled with qnames, assume the standard namespaces apply. For other names, assume they are all local to the document you are writing.



2. [10 pts.] Using the RDFS entailment rules rdfs2, rdfs3, rdfs5, rdfs7, rdfs9, and rdfs11 (see the RDF Semantics recommendation [<http://www.w3.org/TR/rdf-mt/>], Section 7), determine what triples can be inferred from the example in Problem #1. You may ignore the other entailment rules because they do not add anything particularly interesting. Give you answer by listing additional triples in Turtle.

3. [10 pts.] Consider the following RDF document using the XML syntax. Draw the equivalent graph. For convenience, you may use qnames for your node and edge labels.

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:u="http://www.example.org/uni#"
        xml:base="http://www.example.org/uni#">
  <rdfs:Class rdf:ID="Person" />
  <rdfs:Class rdf:ID="Student">
    <rdfs:subClassOf="#Person" />
  </rdfs:Class>
  <rdfs:Class rdf:ID="Professor">
    <rdfs:subClassOf="#Person" />
  </rdfs:Class>
  <rdfs:Class rdf:ID="Course" />
  <rdf:Property rdf:ID="advises">
    <rdfs:domain rdf:resource="#Professor" />
    <rdfs:range rdf:resource="#Student" />
    <rdfs:subPropertyOf="#knows">
  </rdf:Property>
  <rdf:Property rdf:ID="takes">
    <rdfs:domain rdf:resource="#Student" />
    <rdfs:range rdf:resource="#Course" />
  </rdf:Property>
  <rdf:Property rdf:ID="teaches">
    <rdfs:domain rdf:resource="#Professor" />
    <rdfs:range rdf:resource="#Course" />
  </rdf:Property>
  <rdf:Property rdf:ID="knows" />
  <u:Professor rdf:ID="alan">
    <u:teaches rdf:resource="#cs100" />
    <u:advises rdf:resource="#rob" />
    <u:advises rdf:resource="#sarah" />
  </u:Professor>
  <u:Student rdf:ID="rob">
    <u:takes rdf:resource="#cs100" />
    <u:takes rdf:resource="#cs200" />
  </u:Student>
</rdf:RDF>

```

4. [10 pts.] (Required for CSE 428 students, extra credit for CSE 398 students) According to the RDFS Schema recommendation, a property can have multiple rdfs:range statements. When this occurs the range of the property is treated as the intersection of the individual ranges. What advantage does this have over using the union of the individual ranges to determine the actual range? Hint: Consider the distributed design of RDF and the inferences that are sanctioned by rdfs:range.

Electronic Exercises:

The following exercises require both submission of files and hardcopies of these files. The specified files should be included as attachments to a single e-mail sent to heflin@cse.lehigh.edu with subject line “CSE 398/428 – Homework #1 Submission”.

5. [20 pts.] Consider the Semantic Web reading list posted at: <http://www.cse.lehigh.edu/~heflin/courses/sw-readlist.htm>. Your task is to write the equivalent RDF/XML for some of these references. If you are a **CSE 398 student**, you are responsible for describing **5 papers**, if you are a **CSE 428 student**, you are responsible for **10 papers**. See the table below for the specific papers you are responsible for.

Student	Papers
Daniels (398)	[SGK08] to [TCK05]
Dannenhauer	[HD05] to [ABL07]
Hatalis	[KFN04] to [LN04]
Kaplan (398)	[FFR97a] to [CLS05]
Lathrop (398)	[UC106] to [BG04]
Luo	[CHS04] to [KWD97]
Reynolds	[HPP10] to [MSS05]
Sambhawa Priya	[GPH04] to [QHC06]

Student	Papers
Spector (398)	[ESY05] to [NM01b]
Suess	[MBS01] to [GPS04]
Tate	[HIS03] to [SS04]
Tessitore	[WPS03] to [LMU06]
West	[HMK05] to [AKT07]
Wuensch (398)	[VKV06] to [SSG04]
Xia	[GS04] to [JDW01]
<i>Extra Credit</i>	[HH01] to [Hol00]

Note, I use the citation ids to specify your assigned papers. You are responsible for the two papers mentioned and all papers between them, considering the ordering of papers as they appear on the web page. You can receive extra credit by doing more papers than assigned; please choose any additional papers from the Extra Credit set (see the end of the table). If I accidentally assigned you too many papers, feel free to ignore any extras. If you have too few, choose some from the Extra Credit set to round out your tally.

You must write a single file named *your-user-id.rdf* (e.g., xxx013.rdf) that contains RDF/XML descriptions using the vocabulary from the posted **swpub.rdf** schema. Each paper should have a URI with namespace <http://www.cse.lehigh.edu/~heflin/courses/sw-readlist.rdf> and a local part that matches its citation id (the string in square brackets). Categorize the paper as journal article, in conference proceedings, etc. to the best of your ability, and simply use the **PublishedWork** class if nothing else fits. Put as much of the citation information as you can in the RDF file. Also, be sure to use the **topic** property to signify which section of the reading list the paper came from and the **electronicCopy** property to specify the location of the PDF, if it is given. See the file **foundations.rdf** on the course website for examples. You do not need to read the paper, nor do you have to include any information that is not available in the readlist.html file, although you may receive extra credit if you provide additional, correct reference data (e.g., by locating full names, adding page numbers, etc.). If you provide additional data, mark it clearly in your submission. Print a copy of your file and turn it in with the rest of your written answers. Attach the file to the e-mail mentioned above.

6. [40 pts.] Using Jena, write a class **ReadPubRdf** that can read in an RDF file that was written using the vocabulary from **swpub.rdf** and output a list of the publications, one per line. This program is expected to be somewhere between 150 and 200 lines long, so plan your time accordingly. From the command-line, your program should run as:

```
java ReadPubRdf rdf-filename
```

Each entry in the output should be of the form:

```
pub-type:      “title” by author1, author2, ... and authorn (year)
```

where each italicized symbol represents information from a triple. The *pub-type* placeholder should be one of: **Article**, **In Proceedings**, **Book Chapter**, or **Tech Report**. You can ignore any resources that are not instances of one of the corresponding classes. You should use the **foundations.rdf** file mentioned above to test your program, and should also test with the file you created for Problem #5 above. If we ignore the linebreak below, the first output line from the file **foundations.rdf** might look like:

```
In Proceedings: "A Conjunctive Query Language for Description
Logic ABoxes" by Ian Horrocks and Sergio Tessaris (2000)
```

Note, the input files will use the `<rdf:Seq>` and `<rdf:li>` elements to provide ordered lists of authors. Your code must appropriately navigate the resulting Jena model constructs.

For this project, you must use Jena 2.7.4 to parse the input files. The Jena distribution includes a number of JAR files, and many (but not all) of these will need to be in your classpath for your program to compile and run. You will need to import classes from the **com.hp.hpl.jena.rdf.model** package, and may find the **com.hp.hpl.jena.vocabulary.RDF** class useful as well. On the course website, I provide a file **SwPub.java** that includes useful constants for the various classes, properties and other resources, defined in **swpub.rdf**.

Your program will be graded on functionality and style. Style includes modularity (avoid repeated code when possible, keep methods under ~60 lines, use multiple classes when appropriate), commenting (all of your files should be reasonably commented, including an initial comment that identifies you as the author and descriptive comments for each class and method), proper indentation, clear names, and use of standard naming conventions. The program should terminate successfully for any syntactically correct RDF/XML file. You should ignore any triples that do not match the expected vocabulary, and you should ignore any publications that are missing the required **rdf:type**, **title**, **authorList** or **year** triples or that have authors that are missing **fullName** triples.

Print out your .java file(s) and turn in the hardcopy with the rest of your written answers. Create a zip file *your-user-id-prog.zip* that contains both your source code (.java) and compiled (.class) files (but do not include any of the Jena files in it). Attach the zip file to the e-mail mentioned above.