The first practical assignment consists of two main tasks. The first assesses your knowledge of RDF(S) as a modelling language. The second allows you to explore the practical use of the [SPARQL](http://www.w3.org/TR/rdf-sparql-query/) query language and how to process SPARQL queries in Jena.  Jena is a Java framework for building [Semantic Web](http://www.w3.org/2001/sw/) and [linked-data](http://linkeddata.org/) applications that provides a programmatic environment for  RDF, RDFS, OWL and SPARQL.

**Part 1:**

Consider the following RDF graph G:

**@prefix rdf:  www.w3.org/1999/02/22-rdf-syntax-ns#> .**

**@prefix rdfs: www.w3.org/2000/01/rdf-schema#> .**

**@prefix docs:  org/docs/> .**

**@prefix aut:  org/authors/> .**

**@prefix voc:  org/vocab/> .**

**(1)  voc:inJournal rdfs:domain voc:Article .**

**(2)  voc:inJournal rdfs:range voc:Journal .**

**(3)  voc:Book rdfs:subClassOf voc:Publication .**

**(4)  voc:Article rdfs:subClassOf voc:Publication .**

**(5)  voc:Monograph rdfs:subClassOf voc:Book .**

**(6)  voc:hasFirstAuthor rdfs:subPropertyOf voc:hasAuthor .**

**(7)  voc:hasAuthor rdfs:range voc:Author .**

**(8)  docs:a voc:hasFirstAuthor aut:tim .**

**(9)  docs:a voc:hasAuthor aut:james .**

**(10)  docs:a voc:inJournal docs:sciAm .**

**(11) docs:a voc:cites \_:x .**

**(12) \_:x a voc:Book .**

**(13) \_:x voc:hasFirstAuthor aut:ora .**

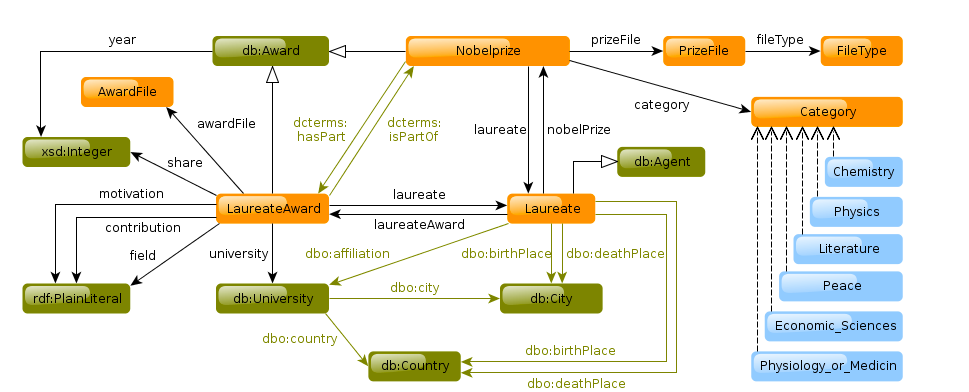
**(14) \_:x voc:cites docs:a .**

For each of the triples (or set of triples) below, verify whether they can be either RDFS or simple entailed from G and show the process for deriving them, or give a short explanation of why they cannot be derived. Each derivation is worth 5 marks, 2 marks for the correct answer and 3 marks for the proof.

1. **docs:a a voc:Publication .**
2. **aut:ora a voc:Author .**
3. **voc:hasFirstAuthor rdfs:domain voc:Publication .**
4. **voc:hasFirstAuthor rdfs:range voc:Author .**
5. **\_:y a voc:Publication .**
6. **voc:Monograph rdfs:subClassOf voc:Publication .**
7. **\_:n voc:cites \_:m .  
   \_:m voc:cites \_:p .  
   \_:p a voc:Article .**
8. **voc:inJournal rdfs:domain voc:Publication .**

**Part 2:**

The second task consists in writing a Java programme that uses Jena to edit an RDF file and answers queries expressed in SPARQL. The [dataset](https://vital.liv.ac.uk/bbcswebdav/pid-2140653-dt-content-rid-13848249_1/xid-13848249_1) used models data about Nobel Laureates and their prizes from the first edition of the Nobel prize in 1901 until recent years. The RDF file with the dump of the dataset (in N-Triple format) is in attachment, and is modelled according to the ontology (schema) specified in the Nobel Prize Linked Data Vocabulary illustrated in the figure below.



Your programme should consists of one class only, named ***RDFprocessing.java***, that reads in the RDF dataset of the Nobel laureates, stores the graph in a triple store, and answers some SPARQL queries against the triple store.

For your convenience, here is a fragment of the Nobel Prize data dump, that describes nobel laureate ["Barack H. Obama"](https://vital.liv.ac.uk/bbcswebdav/pid-2140653-dt-content-rid-10236277_1/xid-10236277_1)

Write a programme that uses the Jena API to carry out the following tasks, in order:

* + Load the data in the Nobel prize data dump file, and create a triple store to host it locally in a directory called NobelDB that should be a sub-directory of your Assignment working directory. Write and execute a SPARQL query against the triple store that prints the first 20 elements in the triple store.
  + Formulate the following SPARQL queries and execute them against the newly created database. Pretty print the results as in the Jena SPARQL tutorial:
    - **Query 1:** ~~Find the name of most recent Nobel Chemistry award winner in the dataset~~ ;
      * Clarification posted on Mon March 2nd: The query should be "Find the name of one of the most recent Nobel Chemistry award winners in the dataset".
    - **Query 2:**Find the categories awarded during the first edition of the Nobel prize (1901);
    - **Query 3:** List, in ascending order of award number, the countries who have Physiology or Medicine prize winners affiliated to one of their universities together with the number of awards received by the country;
    - **Query 4:** Find all the Nobel Laureates, with the year of the award, who were born either in Germany (present or at the time and denoted as “now-Germany”) or in a country that is now known as Germany.

**Useful resources**

**Snorql: Exploring http://data.nobelprize.org/sparql (try the queries here first!)**http://data.nobelprize.org/snorql/

**RDF primer**https://www.w3.org/2007/02/turtle/primer/

**SPARQL specification**

http://www.w3.org/TR/rdf-sparql-query

**Jena SPARQL tutorial**

http://jena.sourceforge.net/ARQ/Tutorial/

**Querying remote SPARQL services with ARQ**http://jena.sourceforge.net/ARQ/sparql-remote.html

**A comprehensive JENA tutorial**http://jena.sourceforge.net/tutorial/RDF\_API/

**Tutorial on how to write/update an RDF file**http://etutorials.org/Misc/Practical+resource+description+framework+rdf/Chapter+8.+Jena+RDF+in+Java/8.2+Creating+and+Serializing+an+RDF+Model/