

## **i Plagiarism and declaration**

### **Plagiarism**

It is not accepted to submit papers/ essays/ exams that have previously been submitted for grading or evaluation, either by you or another student. Resubmitting a paper/ essay/ exam will be regarded as plagiarism, and can lead to serious consequences.

### **Declaration**

I hereby declare that the assignment which I am submitting is my own work and that

- it has not been used in another examination or been submitted or published at another educational institution in Norway or abroad
- it does not contain other people's work without this being stated
- it does not contain my own previous work without this being stated
- the bibliography contains all the literature and all the sources that I have used in my assignment, and that all references refer to this bibliography.

I am aware that any violation of these rules will be considered cheating.

All work you submit at the University of Bergen may be sent for an electronic plagiarism check.

## **i Part 1 - General theory**

**In this task, you will answer 25 questions.**

Scoring: each correct answer usually gives +0.5 points (\*).

Suggested time: 30 minutes.

Weight: ca 12.5% of the final grade.

(\*) A few questions allow multiple choices. Each correct choice then gives +0.25 and each incorrect choice gives -0.25. But the score for everything correct is still 0.5 (even if only one choice is correct) and the minimum is 0.0 (so no negative overall score on any question).

## 1 INFO216 - RDFS axioms

### Axioms in RDFS

Select one alternative:

- Are built into the semantics of RDFS ✔
- There are 22 axioms in RDFS
- Are not a part of the RDFS semantics
- Are based on description logic (DL)
- Must be defined by the user before executing entailment rules

---

Maximum marks: 0.5

## 2 INFO216 - link prediction

When talking about graph embeddings, link prediction is to:

Select one alternative:

- Given two nodes and an edge type, to decide whether they form a plausible triple.
- Given a node, to find semantically similar nodes.
- Given two nodes, to find all property paths between them.
- Given a node and an edge type, to decide nodes that are candidates to form a plausible triple. ✔

(Here, plausible means "likely to be true".)

---

Maximum marks: 0.5

### 3 INFO216 - hasFlightTo property

The hasFlightTo object property between two airports is  
Select one or more alternatives:

- Asymmetric
- Transitive
- Functional
- Irreflexive
- Reflexive
- Inverse functional



---

Maximum marks: 0.5

### 4 INFO216 - JSON-LD forms

JSON-LD is  
Select one alternative:

- A tractable fragment of JSON logic
- W3C's JSON Data Loader standard
- A JSON format for serialising linked data
- Logic description for JSON
- Description logic reasoning in JSON



---

Maximum marks: 0.5

## 5 INFO216 - JSON-LD keywords

Which is NOT a reserved keyword in JSON-LD?

Select one alternative:

- @rule: signifies an entailment rule that applies to the object ✓
- @value: signifies that a value is a literal
- @context: signifies a JSON object that contains the context (or semantic mapping) for the other objects in the same JSON array
- @type: signifies that the JSON object with the @type key has a particular RDF type (or several types)
- @id: signifies that the JSON object with the @id key is identified by a particular URI

---

Maximum marks: 0.5

## 6 INFO216 - RDFS expressiveness

Which one can be expressed in plain RDFS?

Select one alternative:

- Two individuals with different URIs are actually different
- A class is a negation of another class
- The BirthNumber of a Person is unique
- A Republic has exactly one President
- Everyone who receives medical treatment is a patient
- A FootballTeam has 11 players, a VolleyballTeam only 6
- A class is a union (or intersection) of other classes
- A StringQuartet has two violins but only one viola and one cello
- Every ancestor of an ancestor is an ancestor too
- Two properties with different URIs are actually the same



---

Maximum marks: 0.5

## 7 INFO216 - RDFS resource classes

What is true about RDFS resource classes?

Select one alternative:

- The properties of a resource determines its RDFS class ✓
- The properties of a resource are only visible to its neighbours
- Provides Information hiding
- Resources have the same RDFS class throughout their lifetime
- Classes are templates for instantiating objects

---

Maximum marks: 0.5

## 8 INFO216 - why RDFS classes

What is NOT a reason that RDFS has resource classes?

Select one alternative:

- Classes are important for defining and using other RDFS concepts
- RDFS classes restrict which properties RDF resources can have ✓
- We can describe the class formally using RDFS and OWL DL
- Knowing the type (class) of a resource often means we can infer additional information about it (entailment)
- The type (class) of a resource is an important part of its semantics

---

Maximum marks: 0.5

## 9 INFO216 - RDFS Schema

**RDF Schema (RDFS) is NOT**

**Select one alternative:**

- Used for defining other vocabularies
- Used to query RDF graphs
- The foundation for SKOS, OWL and OWL2
- A small RDF vocabulary for more expressive graphs

---

Maximum marks: 0.5

## 10 INFO216 - hasLocation property

**The isLocatedIn object property (between spatial regions) is**

**Select one or more alternatives:**

- Transitive
- Irreflexive
- Asymmetric
- Reflexive
- Inverse functional
- Symmetric

---

Maximum marks: 0.5

## 11 INFO216 - relation prediction

When talking about graph embeddings, relation prediction is to:

**Select one alternative:**

- Given a node, determine whether it plausibly represents a relation or not.
- Given two nodes, to find edge types that are candidates to form a plausible triple. ✓
- Given two nodes and an edge type, to decide whether they form a plausible triple.
- Given a node, to find semantically similar nodes.

(Here, plausible means "likely to be true".)

---

Maximum marks: 0.5

## 12 INFO216 - RDF expressiveness

Which of these are supported by the RDF semantics?

**Select one alternative:**

- Legally owning a gun means owning a licensed weapon
- The object in a hasWorkHomepage triple is a URL
- Everything that is used as a predicate in a triple is an rdf:Property ✓
- A Motorbike is a Vehicle
- The subject in a hasLicensePlate triple is a Vehicle

---

Maximum marks: 0.5



### 13 INFO216 - reification

Reification is that

Select one alternative:

- A resource represents a material thing or place
- A triple is unpacked into four new triples
- A URI answers HTTP requests and returns more information about a resource
- 303 redirection is used to return information about a resource

---

Maximum marks: 0.5

### 14 INFO216 - RDF lists

It is true about an `rdf:List` (collection) that

Select one alternative:

- New members cannot be added without deleting triples
- Cannot contain the same resource several times
- It is easy to add new members
- It is typically used to represent alternatives

---

Maximum marks: 0.5

**15 INFO216 - RDFS containers****An RDFS container CANNOT****Select one alternative:**

- Be an rdfs:Alt, rdfs:Bag or rdfs:Seq
- Have duplicate members
- Be an rdf:List ✓
- Be extended without deleting triples

---

Maximum marks: 0.5

**16 INFO216 - RDF resource types****An RDF resource****Select one alternative:**

- may or may not have an rdf:type
- always has at least one rdf:type ✓
- must have exactly one rdf:type
- always has rdfs:Class as its rdf:type

---

Maximum marks: 0.5

## 17 INFO216 - triple classificaiotn

When talking about graph embeddings, triple classification is to:

**Select one alternative:**

- Given two nodes, to find edge types that are candidates to form a plausible triple.
- Given two nodes and an edge type, to decide whether they form a plausible triple. ✓
- Given a node and an edge type, to find nodes that are candidates to form a plausible triple.
- Given a node, to find semantically similar nodes.

(Here, plausible means "likely to be true".)

---

Maximum marks: 0.5

## 18 INFO216 - RDF resources can be

**An RDF resource can be**

**Select one alternative:**

- a concept
- an information resource
- a property
- a material phenomenon (including people and artefacts)
- any of these ✓

---

Maximum marks: 0.5

## 19 INFO216 - not a LOD best practice

Which is NOT a best practice for data provisioning in the LOD cloud?

Select one alternative:

- Provide dataset-level metadata (e.g., VANN, VS)
- Use terms from widely deployed vocabularies
- Refer to additional access methods (e.g., SPARQL)
- Provide provenance metadata (e.g., PROV)
- Provide licensing metadata (e.g., CC)
- Map proprietary vocabulary terms to other vocabularies
- Make proprietary vocabulary terms dereferencable
- Use URIs that are standardised by the W3C ✓

---

Maximum marks: 0.5

## 20 INFO216 - not a core LOD principle

Which is NOT one of the four core LOD principles?

Select one alternative:

- URIs return information about resources on standard semantic formats
- URIs return information that contain URIs of related resources
- Use URIs that answer to HTTP requests
- Use URIs to identify resources
- Use URIs that are language-independent. ✓

---

Maximum marks: 0.5

## 21 INFO216 - LOD acronym

LOD is an acronym for

Select one alternative:

- Live Online Data
- Lean Open Data
- Linked Open Data ✓
- Linguistic Online Documents

---

Maximum marks: 0.5

## 22 INFO216 - hasSibling property

When we exclude half-siblings, the hasSibling object property between two people is

Select one or more alternatives:

- Symmetric ✓
- Asymmetric
- Transitive ✓
- Inverse functional
- Functional
- Irreflexive ✓
- Reflexive

---

Maximum marks: 0.5

**23 INFO216 - connectedByRoad property (one-way streets)**

The `isConnectedByRoad` object property between two locations (there can be one-way streets) is

Select one or more alternatives:

Functional

Inverse functional

Reflexive

Asymmetric

Irreflexive



Transitive



Symmetric

---

Maximum marks: 0.5

**24 INFO216 - hasMother property**

The hasBirthMother object property between two persons is  
Select one or more alternatives:

- Transitive
- Reflexive
- Functional ✓
- Irreflexive ✓
- Symmetric
- Asymmetric ✓
- Inverse functional

---

Maximum marks: 0.5

**25 INFO216 - hasNeighbour property**

The locatedNextTo object property between two locations is  
Select one or more alternatives:

Asymmetric

Symmetric



Inverse functional

Transitive

Reflexive

Irreflexive



Functional

---

Maximum marks: 0.5



## i Part 2 - Programming

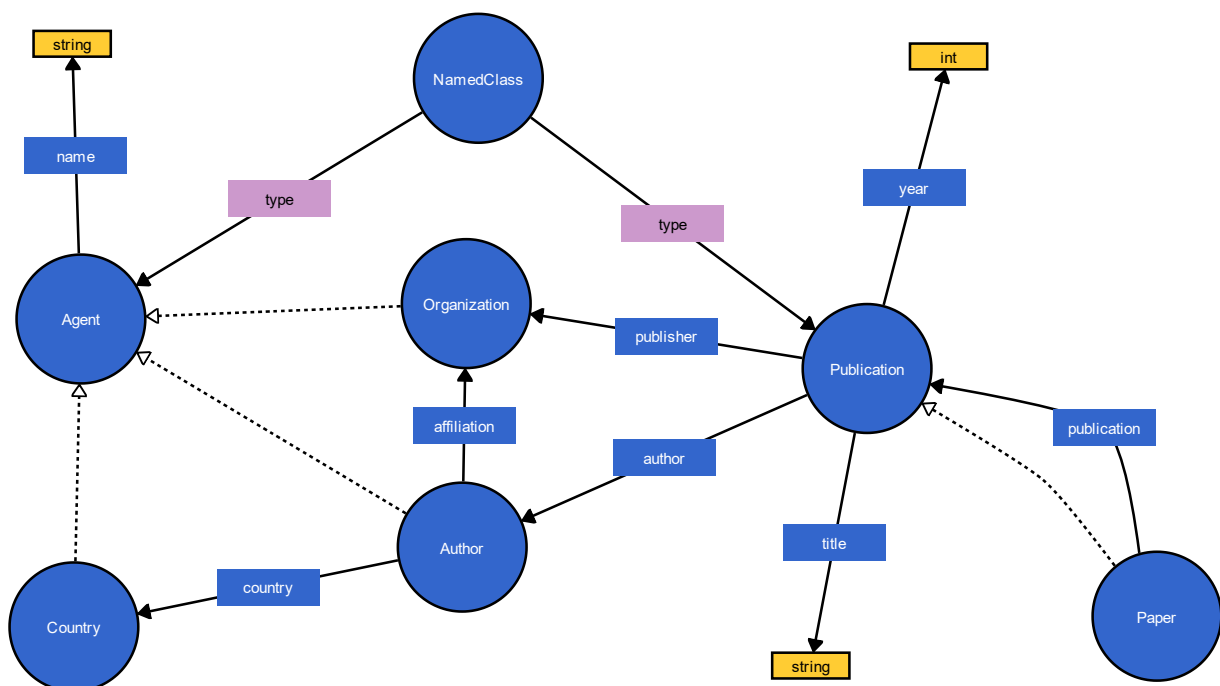
In this task, you will write a program that creates an ontology/knowledge graph, as described in the text and illustrated in the figure.

### Description:

Author-s, Country-s and Organization-s are Agent-s, which have name-s that are string-s. An Author can have affiliation (organization he/she works for) and country.

Paper-s are a type of Publication, and they can also be published in other Publication-s (such as books or journals). Publication-s have string title-s. They also have author-s, and they are published in a year that is described by an integer. A Publication can have a publisher, which is an Organization.

### Illustration:



## 26 Part 2a - Ontology programming

Write a program that creates the ontology using either `rdflib` or `owlready2`.  
(If you use `rdflib`, SPARQL Updates are ok.)

Use the same example prefix `<http://ex.org#>` for all domain-specific classes and properties. In addition, you will need basic terms from RDF, RDFS, OWL, etc., *which should have the correct prefix*.

Scoring:

- you get up to 2 points for overall idea
- you get up to 4 points extra for code that compiles and runs
- you get up to 6 points for getting the ontology contents right

Suggested time: 30 minutes.

Weight: ca 12% of the final grade.

Here is a list of domain-specific classes and properties used, if you want to copy-and-paste:

Classes:	Properties:
:Agent :Author :Organization :Country :Paper :Publication	:name :affiliation :title :publisher :publication :author :country :year

Fill in your answer here

1

Maximum marks: 12



## 27 Part 2b: Knowledge-graph programming

Write a program that adds the following individuals and relations to the ontology/knowledge graph you just created. (When there are several triples of the same type, it is not critical that you add all of them.)

Use the same example prefix `<http://ex.org#>` for all domain-specific classes, properties, and individuals. In addition, you will need basic terms from RDF, RDFS, OWL, etc., *which should have the correct prefix*.

Title	Authors	Publication	Publisher	Year
:The_semantic_web	:Tim_Berners-Lee, :James_Hendler	:Scientific_American	:Springer_Nature	2001
:DBpedia_A_nucleus	:Soren_Auer, :Christian_Bizer	:The_semantic_web_book	:Springer_Nature	2007
:Linked_data_The_story_so_far	:Christian_Bizer, :Tim_Berners-Lee	:Semantic_services_interoperability_and_web_applications	:IGI_Global	2011

Author	Affiliation	Country
:Tim_Berners-Lee	:Massachusetts_Institute_of_Technology	:United_States
:Soren_Auer	:Leibniz_University_Hannover	:Germany
:Christian_Bizer	:University_of_Mannheim	:Germany
:James_Hendler	:Rensselaer_Polytechnic_Institute	:United_States

Scoring: you get up to 4 points for correct triples.

Suggested time: 15 minutes.

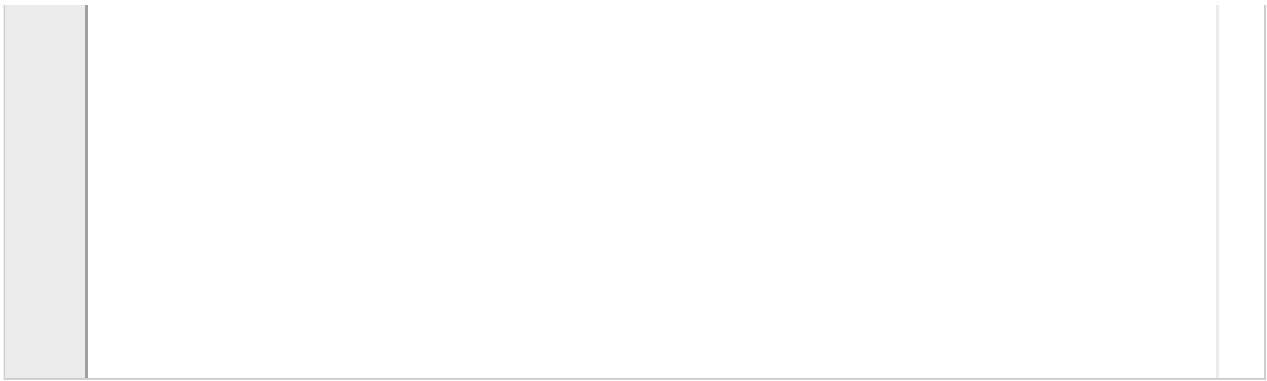
Weight: ca 4% of the final grade.

*Note* - in the table above, these URLs were split over two lines, but they are only two, not four resources:

- :Linked\_data\_The\_story\_so\_far
- :Semantic\_services\_interoperability\_and\_web\_applications

**Fill in your answer here**

1	
---	--



---

Maximum marks: 4

## i Part 3 - Standard terms

The classes and properties in the description and illustration in part 2 lack prefixes. But many similar classes and properties are already defined (exactly or closely matching) in standard vocabularies and ontologies.

**Which of these 12 standard terms do you think are best?**

Scoring: you get 0.8 points for each correct prefix or name.

Suggested time: 15 minutes.

Weight: ca 10% of the final grade.

## 28 INFO216 - Terms 1

For Agent I would use:

**Select one alternative:**

dc:Agent

sioc:Agent

foaf:Agent



(In this part, you can assume that 'dc:' is either 'dce:' or 'dcterms:'.)

---

Maximum marks: 0.8

**29 INFO216 - Terms 2**

For type I would use:

**Select one alternative:**

- dc:type
- owl:type
- rdf:type



---

Maximum marks: 0.8

**30 INFO216 - Terms 3**

For NamedClass I would use:

**Select one alternative:**

- dbpedia-owl:NamedClass
- rdfs:NamedClass
- owl:NamedClass



---

Maximum marks: 0.8

**31 INFO216 - Terms 4**

For subClassOf I would use:

**Select one alternative:**

- rdf:subClassOf
- owl:subClassOf
- rdfs:subClassOf



---

Maximum marks: 0.8

**32 INFO216 - Terms 5**

For Organization I would use:

**Select one alternative:**

- foaf:Organization
- dc:Organization
- sioc:Organizarion



---

Maximum marks: 0.8

**33 INFO216 - Terms 6**

For name I would use:

**Select one alternative:**

skos:name

rdfs:name

foaf:name



---

Maximum marks: 0.8

**34 INFO216 - Terms 7**

For string I would use:

**Select one alternative:**

xsd:string

schema:string

rdf:string



---

Maximum marks: 0.8



**35 INFO216 - Terms 8**

For affiliation I would use:

**Select one alternative:**

- rdfs:affiliation
- schema:affiliation
- dc:affiliation



---

Maximum marks: 0.8

**36 INFO216 - Terms 9**

For title I would use:

**Select one alternative:**

- foaf:title
- dc:title
- skos:title



---

Maximum marks: 0.8

**37 INFO216 - Terms 10**

For publisher I would use:

**Select one alternative:**

- foaf:publisher
- bibo:publisher
- dc:publisher



---

Maximum marks: 0.8

**38 INFO216 - Terms 11**

For publication (i.e., that a Paper is published in another Publication) I would use:

**Select one alternative:**

- dc:isPartOf
- foaf:publication
- bibo:inCollection



---

Maximum marks: 0.8

**39 INFO216 - Terms 12**

For author I would use:

**Select one alternative:**

- rdfs:author
- dc:contributor
- foaf:author



---

Maximum marks: 0.8

**i Part 4 - Restrictions and reasoning**

Part 4 has two sub-parts.

**First you will write restrictions in RDFS, and then in OWL.**

Suggested time for *both sub-parts*: 40 minutes.

Weight: ca 16% of the final grade.

**i Part 4a - Simple restrictions in RDFS/OWL**

**Write the following 10 statements as RDFS/OWL triples in Turtle.**

Continue to use the same example prefix as in Part 2: `<http://ex.org#>` for all domain-specific classes and properties, along with the basic terms from RDF, RDFS, OWL, etc.

*You do not have to write the prefix definitions yourself.* You can assume that the prefixes ':', 'rdf:', 'rdfs:', 'owl:', etc. are already defined.

Scoring: you get 0.8 points for each statement.

Suggested time: 20 minutes.

Weight: ca 8% of the final grade.

**40 INFO216 - Simple restrictions 1**

An Organization is a kind of Agent.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 0.8

**41 INFO216 - Simple restrictions 2**

The object in an affiliation-triple is an Organization.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 0.8

**42 INFO216 - Simple restrictions 3**

The subject in an affiliation-triple is an Author.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 0.8

**43 INFO216 - Simple restrictions 4**

A Paper is published in only one Publication.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 0.8

**44 INFO216 - Simple restrictions 5**

A Paper cannot have more than one publication year.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 0.8



**45 INFO216 - Simple restrictions 6**

A Paper cannot be its own publication.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 0.8

**46 INFO216 - Simple restrictions 7**

publication is a transitive relation.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 0.8

**47 INFO216 - Simple restrictions 8**

Different Author-s have different name-s.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 0.8

**48 INFO216 - Simple restrictions 9**

An Author is not an Organization.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 0.8

**49 INFO216 - Simple restrictions 10**

A Publication title is a string.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 0.8

**i Part 4b - Complex restrictions in OWL/RDFS**

**Write the following 5 statements as OWL/RDFS expressions in Turtle.**

Continue to use the same example prefix as in Part 2: `<http://ex.org#>` for all domain-specific classes and properties, along with basic terms from RDF, RDFS, OWL, etc.

*You do not have to write the prefix definitions yourself.* You can assume that the prefixes ':', 'rdf:', 'rdfs:', 'owl:', etc. are already defined.

Scoring: you get 1.6 points for each statement.

Suggested time: 20 minutes.

Weight: ca 8% of the final grade.

**50 INFO16 - Complex restrictions 1**

A Paper must have at least one author.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 1.6

**51 INFO216 - Complex restrictions 2**

A Paper has exactly one title.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 1.6



**52 INFO216 - Complex expressions 3**

A year must be in range from 1900 to 2050 (inclusive).

**Fill in your answer here**

1	
---	--

---

Maximum marks: 1.6

**53 INFO216 - Complex expressions 4**

Author-s, Organization-s, and Country-s are disjoint.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 1.6

**54 INFO216 - Complex expressions 5**

A Publisher must be either of ACM, IEEE CS, Springer Nature, or IGI Global.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 1.6

## i Part 5 - Reasoning

In this part, we will run reasoning over the classes, properties, and individuals from Part 2, with the restrictions from Part 4 added. Then you will be asked about the results of reasoning.

Continue to use the same example prefix as in Part 2: `<http://ex.org#>` for all domain-specific classes and properties, along with basic terms from RDF, RDFS, OWL, etc.

Assume we have run `RDFS_OWLRL_Semantics` to execute RDFS entailments.

Scoring: you get 1.5 points for each correct choice.

Suggested time: 30 minutes.

Weight: ca 13.5% of the final grade.

### 55 INFO216 - Reasoning 1

After running entailments, it is true that:  
Christian Bizer has country Germany,

Select one alternative:

Incorrect

Correct



---

Maximum marks: 1.5

**56 INFO216 - Reasoning 2**

After running entailments, it is true that:  
University of Mannheim has country Germany,

Select one alternative:

Incorrect

Correct



---

Maximum marks: 1.5

**57 INFO216 - Reasoning 3**

After running entailments, it is true that:  
The semantic web has type Paper,

Select one alternative:

Incorrect

Correct



---

Maximum marks: 1.5

**58 INFO216 - Reasoning 4**

After running entailments, it is true that:  
The semantic web has type Publication,

Select one alternative:

- Incorrect
- Correct



---

Maximum marks: 1.5

**59 INFO216 - Reasoning 5**

After running entailments, it is true that:  
The semantic web book has type Paper,

Select one alternative:

- Incorrect
- Correct



---

Maximum marks: 1.5

**60 INFO216 - Reasoning 6**

After running entailments, it is true that:  
The semantic web book has type Publication,

Select one alternative:

- Incorrect
- Correct



---

Maximum marks: 1.5

**61 INFO216 - Reasoning 7**

Assume the following statement has been added as a triple:

The semantic web is published in The semantic web book.

Which of the following statements are now true?

**Select one or more alternatives:**

- RDFS-OWLRL reasoning goes through with no errors or warnings.
- RDFS-OWLRL reasoning halts due to an exception..
- Christian\_Bizer is now the OWL-same author as Time\_Berners-Lee.
- The\_semantic\_web paper is now OWL-same as the DBpedia\_A\_nucleus paper. ✓
- The Scientific\_American publication is now OWL-same as The\_semantic\_web\_bo. ✓
- RDFS-OWLRL reasoning goes through, but it reports reasoning errors. ✓

(Because you can choose multiple alternatives, each correct answer gives 1.5 and each incorrect one gives -1.5, but you cannot score negative on the task as a whole.)

---

Maximum marks: 4.5

## i Part 6 - SPARQL

**In this part you will write 16 SPARQL queries and updates to the ontology and instances from Part 2.**

Assume we have run an RDFS-OWLRL reasoner as in Part 5. Also assume the KG is complete so we can query with negation and cardinality restrictions.

Use the same names of classes, properties, individuals, and prefixes as before. You will only write the SPARQL query or update itself, not any of the program code needed to execute it.

*You do not have to write the prefix definitions yourself.* You can assume that the prefixes ':', 'rdf:', 'rdfs:', 'owl:', etc. are already defined.

Scoring: 1.0 point per correct query/update

Suggested time: 40 minutes.

Weight: 16% of the final grade.



**62 INFO216 - SPARQL query 1**

List all Paper title-s. (Example output below.)

**Fill in your answer here**

1
---

Example output:

title

-----

"DBpedia A nucleus"

"Linked data The story so far"

"The semantic web"

---

Maximum marks: 1

**63 INFO216 - SPARQL query 2**

List all publisher Organization-s in alphabetical order.

**Fill in your answer here**

1	
---	--

Example output:

name

-----

"IGI Global"

"Springer Nature"

---

Maximum marks: 1

**64 INFO216 - SPARQL query 3**

List all Author-s along with the title-s of the Paper-s they have written.

**Fill in your answer here**

1	
---	--

Example output:

author | title

```
-----  
"Christian Bizer"|"DBpedia A nucleus"  
"Christian Bizer"|"Linked data The story so far"  
"James Hendler"|"The semantic web"  
"Jens Lehmann"|"DBpedia A nucleus"  
"Ora Lassila"|"The semantic web"  
"Soren Auer"|"DBpedia A nucleus"  
"Tim Berners-Lee"|"Linked data The story so far"  
"Tim Berners-Lee"|"The semantic web"  
"Tom Heath"|"Linked data The story so far"
```

---

Maximum marks: 1

**65 INFO216 - SPARQL query 4**

List all Country-s and numbers of Paper-s from that Country.

**Fill in your answer here**

1	
---	--

Example output:

country | number

-----  
"Germany" | "3"^^<...>

"United States" | "3"^^<...>

---

Maximum marks: 1

**66 INFO216 - SPARQL queries 5**

List all Author-s of more than one Paper in descending order (of paper numbers).

**Fill in your answer here**

1	
---	--

Example output.

author	number
"Christian Bizer"	"2"^^<...>
"Tim Berners-Lee"	"2"^^<...>

---

Maximum marks: 1

## 67 INFO216 - SPARQL queries 6

For each Author, write the first and last year they have published a Paper.

Fill in your answer here

1	
---	--

Example output:

name | min | max

```

-----
"Christian Bizer"|"2007"^^<...>|"2011"^^<...>
"James Hendler"|"2001"^^<...>|"2001"^^<...>
"Jens Lehmann"|"2007"^^<...>|"2007"^^<...>
"Ora Lassila"|"2001"^^<...>|"2001"^^<...>
"Soren Auer"|"2007"^^<...>|"2007"^^<...>
"Tim Berners-Lee"|"2001"^^<...>|"2011"^^<...>
"Tom Heath"|"2011"^^<...>|"2011"^^<...>

```

Maximum marks: 1

**68 INFO216 - SPARQL queries 7**

List all Authors that are not affiliated with an Organization in Germany.

**Fill in your answer here**

1	
---	--

Example output:

name

-----

"James Hendler"

"Tim Berners-Lee"

---

Maximum marks: 1

**69 INFO216 - SPARQL queries 8**

Is James Hendler the author of more than one Paper?

**Fill in your answer here**

1	
---	--

Example output:  
False

---

Maximum marks: 1



**70 INFO216 - SPARQL queries 9**

Create a new graph of all Author-s that have author-ed a Paper with Christian Bizer, including their affiliation-s and country-s. (Example output below.)

Fill in your answer here

1	
---	--

Partial example output:

```
@prefix : <http://ex.org#> .
:Christian_Bizer a :Author ;
  :affiliation :University_of_Mannheim ;
  :country :Germany ;
  :name "Christian Bizer" .
```

---

Maximum marks: 1

## 71 INFO216 - SPARQL queries 10

Create a new graph of all Author-s that have author-ed a Paper with Christian Bizer, including their Paper-s, affiliation-s and country-s.

Fill in your answer here

1	
---	--

Partial example output:

```
@prefix : <http://ex.org#> .
:Christian_Bizer a :Author ;
  :affiliation :University_of_Mannheim ;
  :country :Germany ;
  :name "Christian Bizer" .
:DBpedia_A_nucleus a :Paper ;
  :author :Christian_Bizer, :Soren_Auer ;
  :title "DBpedia A nucleus" .
```

---

Maximum marks: 1

**72 INFO216 - SPARQL updates 1**

An Organization that is an Author's affiliation is an Institution.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 1

**73 INFO216 - SPARQL updates 2**

If an Author has a country, then that Author's affiliation Organization is locatedIn the same Country.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 1

**74 INFO216 - SPARQL updates 3**

A Paper is producedBy the Organization-s its Author-s are affiliated with.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 1

**75 INFO216 - SPARQL updates 4**

A Paper is produced in the Country-s of its Author-s Institutions are located in.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 1

**76 INFO216 - SPARQL updates 5**

Delete all direct (country) connections between Author-s and Country-s.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 1

**77 INFO216 - SPARQL updates 6**

If a Paper has more than one year, delete the most recent ones.

**Fill in your answer here**

1	
---	--

---

Maximum marks: 1



## i Part 7 - rdflib understanding

In this part you are given a Python program that loads the ontology from Part 2a. It then reads the papers and authors from Part 2b from files. Finally, the program saves them together in a knowledge graph.

**Unfortunately, the program does not work yet. You must identify 8 errors that must be fixed before the program can run correctly.**

All the errors are somehow related to knowledge graphs and rdflib, none of them are "pure Python" errors.

Scoring: you get 2.0 point for each correctly identified error

Suggested time: 40 minutes.

Weight: 16% of the final grade.

Here are examples of input data to the program:

### **File papers.txt:**

The semantic web;Tim Berners-Lee,James Hendler;Scientific American;Springer Nature;2001  
DBpedia A nucleus;Soren Auer,Christian Bizer;The semantic web book;Springer Nature;2007  
Linked data The story so far;Christian Bizer,Tim Berners-Lee;Semantic services interoperability  
and web applications;IGI Global;2011

### **File authors.txt:**

Tim Berners-Lee;Massachusetts Institute of Technology;United States  
Soren Auer;Leibniz University Hannover;Germany  
Christian Bizer;University of Mannheim;Germany  
James Hendler;Rensselaer Polytechnic Institute;United States

**78 INFO216 - rdflib errors**

**For each error you find in the exam program, copy the line where the error first appears. If the same error is repeated, you only copy the first line.**

Leave a blank line between each error line you find.

Attached you can find a file with 8 errors to identify.

**Fill in your answer here**



---

Maximum marks: 16

**79 Part 8 - Comments to the exam**

If you have any comments to the exam, you can enter them here:

**Fill in your answer here**

---

Maximum marks: 0

**Question 78**  
Attached



```

from rdflib import Graph, Namespace, RDF, RDFS, OWL, XSD, Literal

# Define paths and file names
BASE = 'http://ex.org#'
ONT_FILE = 'ontology.ttl'
PAPER_FILE = 'papers.txt'
AUTHOR_FILE = 'authors.txt'
KG_FILE = 'created_kg.ttl'

# Create rdflib graph
Graph()

# Read ontology from file
g.parse(ONT_FILE, format='owl')

def str_to_id(str):
    """Make sure that the string is a legal part of a URL."""
    return str.replace(' ', '_')

def add_paper(g, line):
    """Add information about a paper to the graph."""
    if line.strip()=='0':
        return
    title, author_names, source, pub_name, year = line.split(';')

    paper = BASE['publ_'+str_to_id(title.strip())]
    g.add(paper, RDF.type, BASE.Paper)
    g.add(paper, BASE.title, title.strip())
    g.add(paper, BASE.year, int(year))

    publication = BASE['publ_'+str_to_id(source.strip())]
    g.add(publication, BASE.title, source.strip())
    g.add(paper, BASE.publication, publication)

    publisher = BASE['org_'+str_to_id(pub_name.strip())]
    g.add(publisher, BASE.name, pub_name.strip())
    g.add(paper, BASE.publisher, publisher)

    for auth_name in author_names.split(','):
        author = BASE['author_'+str_to_id(auth_name.strip())]
        g.add(author, BASE.name, auth_name.strip())
        g.add(paper, BASE.author, author)

# Read paper information from file and add to graph
with open(PAPER_FILE) as file:
    for line in file:
        add_paper(g, line)

def add_author(g, line):
    """Add information about an author to the graph."""
    if line.strip()=='0':
        return
    name, aff_name, country_name = line.split(';')

    author = BASE['author_'+str_to_id(name.strip())]

```

```

g.add(author, RDF.type, BASE.Author)
g.add(author, BASE.name, name.strip())

affiliation = BASE['org_'+str_to_id(aff_name.strip())]
g.add(affiliation, BASE.name, aff_name.strip())
g.add(author, BASE.affiliation, affiliation)

country = BASE['country_'+str_to_id(country_name.strip())]
g.add(country, BASE.name, country_name.strip())
g.add(author, BASE.country, country)

# Read author information from file and add to graph
with open(AUTHOR_FILE) as file:
    for line in file:
        add_author(g, line)

# Execute RDFS entailments
rdfs = RDFS_OWLRL_Semantics(g, False, False, False)

# Remove unnecessary triples
g.update("""
DELETE {
    ?s ?p rdfs:Resource .
}
""")

# Save to file
g.serialize(KG_FILE, format='ttl')

```