In [1]:

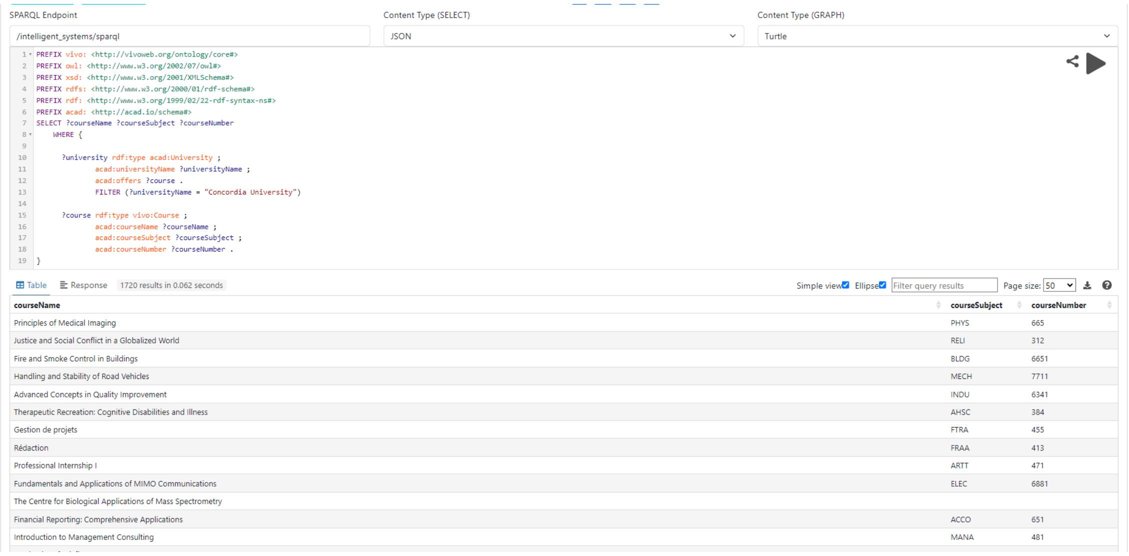
```
import requests
import json
# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'
```

1. List all courses offered by [university]

In [22]:

```
# Define the SPARQL query
university_name = "Concordia University"

sparql_query = f"""
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>
SELECT ?courseName ?courseSubject ?courseNumber
    WHERE {{

          ?university rdf:type acad:University ;
                          acad:universityName ?universityName ;
                          acad:offers ?course .
              FILTER (?universityName = "{university_name}")

          ?course rdf:type vivo:Course ;
              acad:courseName ?courseName ;
                          acad:courseSubject ?courseSubject ;
                          acad:courseNumber ?courseNumber .
}}
"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
print(len(response.json()['results']['bindings']))
print(response.json()['results']['bindings'][:5])
```

```
1720
[{'courseName': {'type': 'literal', 'value': 'Principles of Medical Imaging'}, 'cour
seSubject': {'type': 'literal', 'value': 'PHYS'}, 'courseNumber': {'type': 'litera
l', 'value': '665'}}, {'courseName': {'type': 'literal', 'value': 'Justice and Socia
l Conflict in a Globalized World'}, 'courseSubject': {'type': 'literal', 'value': 'R
ELI'}, 'courseNumber': {'type': 'literal', 'value': '312'}}, {'courseName': {'type':
'literal', 'value': 'Fire and Smoke Control in Buildings'}, 'courseSubject': {'typ
e': 'literal', 'value': 'BLDG'}, 'courseNumber': {'type': 'literal', 'value': '665
1'}}, {'courseName': {'type': 'literal', 'value': 'Handling and Stability of Road Ve
hicles'}, 'courseSubject': {'type': 'literal', 'value': 'MECH'}, 'courseNumber': {'t
ype': 'literal', 'value': '7711'}}, {'courseName': {'type': 'literal', 'value': 'Adv
anced Concepts in Quality Improvement'}, 'courseSubject': {'type': 'literal', 'valu
e': 'INDU'}, 'courseNumber': {'type': 'literal', 'value': '6341'}}]
```



SPARQL Endpoint: /intelligent_systems/sparql

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

```
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>
SELECT ?courseName ?courseSubject ?courseNumber
    WHERE {

      ?university rdf:type acad:University ;
              acad:universityName ?universityName ;
              acad:offers ?course .
              FILTER (?universityName = "Concordia University")

      ?course rdf:type vivo:Course ;
              acad:courseName ?courseName ;
              acad:courseSubject ?courseSubject ;
              acad:courseNumber ?courseNumber .
}
```

1720 results in 0.062 seconds

Simple view☑ Ellipse☑ Filter query results Page size: 50

courseName	courseSubject	courseNumber
Principles of Medical Imaging	PHYS	665
Justice and Social Conflict in a Globalized World	RELI	312
Fire and Smoke Control in Buildings	BLDG	6651
Handling and Stability of Road Vehicles	MECH	7711
Advanced Concepts in Quality Improvement	INDU	6341
Therapeutic Recreation: Cognitive Disabilities and Illness	AHSC	384
Gestion de projets	FTRA	455
Rédaction	FRAA	413
Professional Internship I	ARTT	471
Fundamentals and Applications of MIMO Communications	ELEC	6881
The Centre for Biological Applications of Mass Spectrometry		
Financial Reporting: Comprehensive Applications	ACCO	651
Introduction to Management Consulting	MANA	481

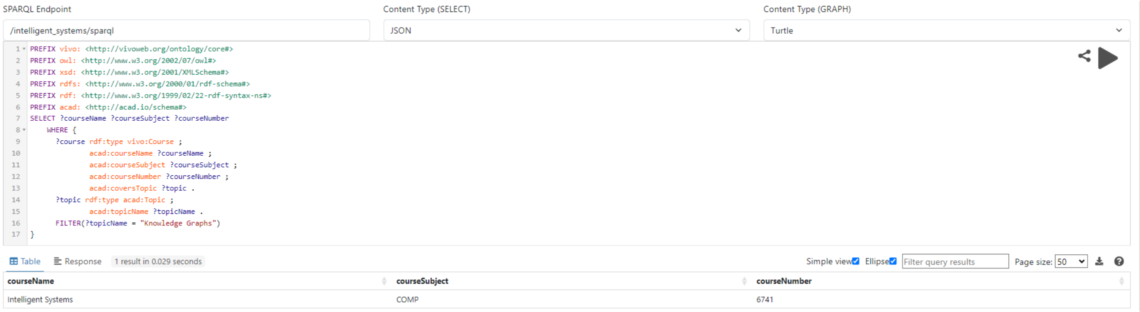
2. In which courses is [topic] discussed?

In [3]:

```
# Define the SPARQL query
topic_name = "Knowledge Graphs"

sparql_query = f"""
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>
SELECT ?courseName ?courseSubject ?courseNumber
    WHERE {{
          ?course rdf:type vivo:Course ;
              acad:courseName ?courseName ;
                          acad:courseSubject ?courseSubject ;
                          acad:courseNumber ?courseNumber ;
              acad:coversTopic ?topic .
      ?topic rdf:type acad:Topic ;
              acad:topicName ?topicName .
      FILTER(?topicName = "{topic_name}")
}}
```

```python
"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
print(json.dumps(response.json(), indent=1))
```

```
{
 "head": {
  "vars": [
   "courseName",
   "courseSubject",
   "courseNumber"
  ]
 },
 "results": {
  "bindings": [
   {
    "courseName": {
     "type": "literal",
     "value": "Intelligent Systems"
    },
    "courseSubject": {
     "type": "literal",
     "value": "COMP"
    },
    "courseNumber": {
     "type": "literal",
     "value": "6741"
    }
   }
  ]
 }
}
```



SPARQL Endpoint: /intelligent_systems/sparql

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

```sparql
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>
SELECT ?courseName ?courseSubject ?courseNumber
    WHERE {
      ?course rdf:type vivo:Course ;
              acad:courseName ?courseName ;
              acad:courseSubject ?courseSubject ;
              acad:courseNumber ?courseNumber ;
              acad:coversTopic ?topic .
      ?topic rdf:type acad:Topic ;
              acad:topicName ?topicName .
      FILTER(?topicName = "Knowledge Graphs")
}
```

1 result in 0.029 seconds

Simple view ☑ Ellipse ☑ Filter query results Page size: 50

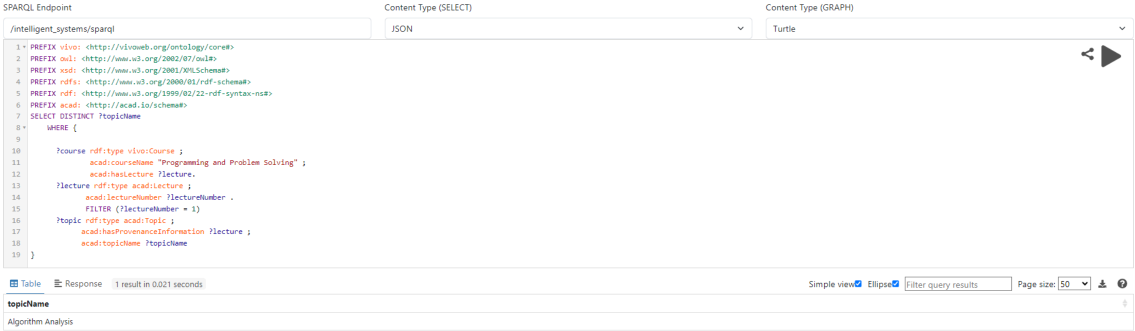
courseName	courseSubject	courseNumber
Intelligent Systems	COMP	6741

3. Which [topics] are covered in [course] during [lecture number]?

In [4]:

```
# Define the SPARQL query
course_name = "Programming and Problem Solving"
lecture_number = 1

sparql_query = f"""
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>
SELECT DISTINCT ?topicName
    WHERE {{

          ?course rdf:type vivo:Course ;
              acad:courseName "{course_name}" ;
                          acad:hasLecture ?lecture.
      ?lecture rdf:type acad:Lecture ;
             acad:lectureNumber ?lectureNumber .
                 FILTER (?lectureNumber = {lecture_number})
          ?topic rdf:type acad:Topic ;
            acad:hasProvenanceInformation ?lecture ;
            acad:topicName ?topicName
}}
"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
print(json.dumps(response.json(), indent=1))
```

```
{
 "head": {
  "vars": [
   "topicName"
  ]
 },
 "results": {
  "bindings": [
   {
    "topicName": {
     "type": "literal",
     "value": "Algorithm Analysis"
    }
   }
  ]
 }
}
```



/intelligent_systems/sparql

JSON

Turtle

```
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>
SELECT DISTINCT ?topicName
    WHERE {

      ?course rdf:type vivo:Course ;
              acad:courseName "Programming and Problem Solving" ;
              acad:hasLecture ?lecture.
      ?lecture rdf:type acad:Lecture ;
             acad:lectureNumber ?lectureNumber .
             FILTER (?lectureNumber = 1)
      ?topic rdf:type acad:Topic ;
            acad:hasProvenanceInformation ?lecture ;
            acad:topicName ?topicName
}
```

1 result in 0.021 seconds

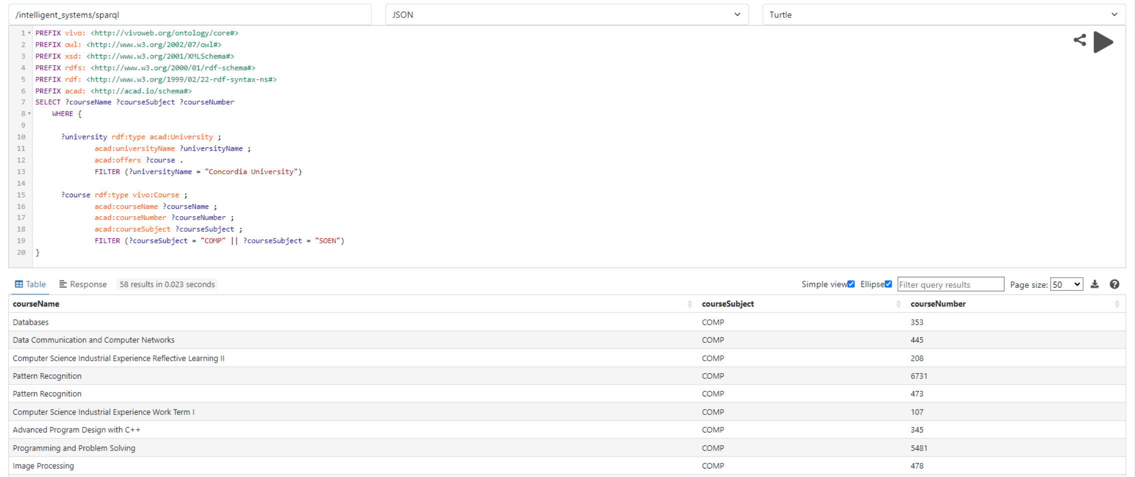
topicName
Algorithm Analysis

4. List all [courses] offered by [university] within the [subject] (e.g., \COMP", \SOEN").

In [23]:
```
# Define the SPARQL query
course_subject1 = "COMP"
course_subject2 = "SOEN"
university_name = "Concordia University"
lecture_number = 1

sparql_query = f"""
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>
SELECT ?courseName ?courseSubject ?courseNumber
    WHERE {{

          ?university rdf:type acad:University ;
                          acad:universityName ?universityName ;
                          acad:offers ?course .
              FILTER (?universityName = "{university_name}")

          ?course rdf:type vivo:Course ;
              acad:courseName ?courseName ;
                          acad:courseNumber ?courseNumber ;
              acad:courseSubject ?courseSubject ;
              FILTER (?courseSubject = "{course_subject1}" || ?courseSubject = "{co
}}
"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
```

```python
# Print the response
print(len(response.json()['results']['bindings']))
print(response.json()['results']['bindings'][:5])
```

```
58
[{'courseName': {'type': 'literal', 'value': 'Databases'}, 'courseSubject': {'type':
'literal', 'value': 'COMP'}, 'courseNumber': {'type': 'literal', 'value': '353'}},
{'courseName': {'type': 'literal', 'value': 'Data Communication and Computer Network
s'}, 'courseSubject': {'type': 'literal', 'value': 'COMP'}, 'courseNumber': {'type':
'literal', 'value': '445'}}, {'courseName': {'type': 'literal', 'value': 'Computer S
cience Industrial Experience Reflective Learning II'}, 'courseSubject': {'type': 'li
teral', 'value': 'COMP'}, 'courseNumber': {'type': 'literal', 'value': '208'}}, {'co
urseName': {'type': 'literal', 'value': 'Pattern Recognition'}, 'courseSubject': {'t
ype': 'literal', 'value': 'COMP'}, 'courseNumber': {'type': 'literal', 'value': '673
1'}}, {'courseName': {'type': 'literal', 'value': 'Pattern Recognition'}, 'courseSub
ject': {'type': 'literal', 'value': 'COMP'}, 'courseNumber': {'type': 'literal', 'va
lue': '473'}}]
```



/intelligent_systems/sparql

JSON

Turtle

```sparql
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>
SELECT ?courseName ?courseSubject ?courseNumber
    WHERE {

      ?university rdf:type acad:University ;
              acad:universityName ?universityName ;
              acad:offers ?course .
              FILTER (?universityName = "Concordia University")

      ?course rdf:type vivo:Course ;
              acad:courseName ?courseName ;
              acad:courseNumber ?courseNumber ;
              acad:courseSubject ?courseSubject ;
              FILTER (?courseSubject = "COMP" || ?courseSubject = "SOEN")
}
```

58 results in 0.023 seconds

Simple view ☑ Ellipse ☑ Filter query results Page size: 50

courseName	courseSubject	courseNumber
Databases	COMP	353
Data Communication and Computer Networks	COMP	445
Computer Science Industrial Experience Reflective Learning II	COMP	208
Pattern Recognition	COMP	6731
Pattern Recognition	COMP	473
Computer Science Industrial Experience Work Term I	COMP	107
Advanced Program Design with C++	COMP	345
Programming and Problem Solving	COMP	5481
Image Processing	COMP	478

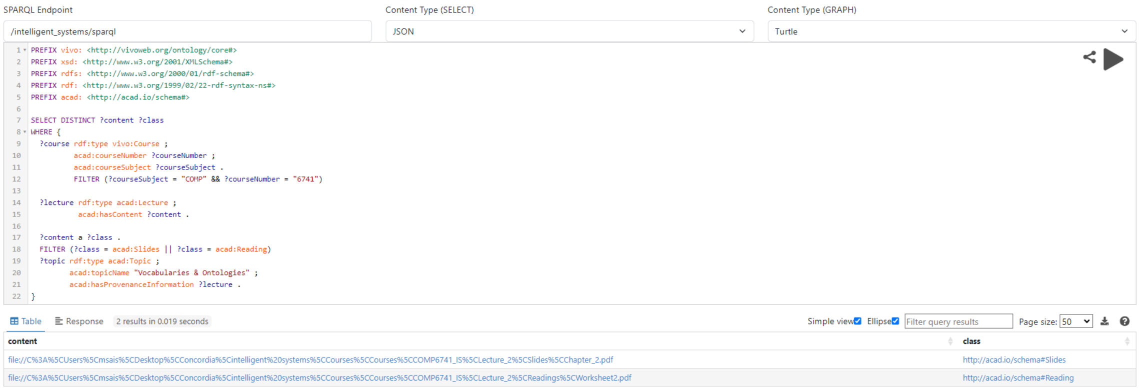
5. What [materials] (slides, readings) are recommended for [topic] in [course] [number]?

In [6]:

```python
# Define the SPARQL query
course_subject = "COMP"
course_number = "6741"
topic_name = "Vocabularies & Ontologies"

sparql_query = f"""
PREFIX ac: <http://umbel.org/umbel/ac/>
PREFIX prefix: <http://prefix.cc/>
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?content ?class
WHERE {{
  ?course rdf:type vivo:Course ;
```

```
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject .
          FILTER (?courseSubject = "{course_subject}" && ?courseNumber = "{course_n

  ?lecture rdf:type acad:Lecture ;
           acad:hasContent ?content .

  ?content a ?class .
  FILTER (?class = acad:Slides || ?class = acad:Reading)
  ?topic rdf:type acad:Topic ;
         acad:topicName "{topic_name}" ;
         acad:hasProvenanceInformation ?lecture .
}}
"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
print(json.dumps(response.json(), indent=1))
```

```
{
 "head": {
  "vars": [
   "content",
   "class"
  ]
 },
 "results": {
  "bindings": [
   {
    "content": {
     "type": "uri",
     "value": "file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_2%5CSlides%5CChapter_2.pdf"
    },
    "class": {
     "type": "uri",
     "value": "http://acad.io/schema#Slides"
    }
   },
   {
    "content": {
     "type": "uri",
     "value": "file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_2%5CReadings%5CWorksheet2.pdf"
    },
    "class": {
     "type": "uri",
     "value": "http://acad.io/schema#Reading"
    }
   }
  ]
 }
}
```



SPARQL Endpoint: /intelligent_systems/sparql

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

```
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?content ?class
WHERE {
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject .
          FILTER (?courseSubject = "COMP" && ?courseNumber = "6741")

  ?lecture rdf:type acad:Lecture ;
           acad:hasContent ?content .

  ?content a ?class .
  FILTER (?class = acad:Slides || ?class = acad:Reading)
  ?topic rdf:type acad:Topic ;
         acad:topicName "Vocabularies & Ontologies" ;
         acad:hasProvenanceInformation ?lecture .
}
```

2 results in 0.019 seconds

Simple view ☑ Ellipse ☑ Filter query results Page size: 50

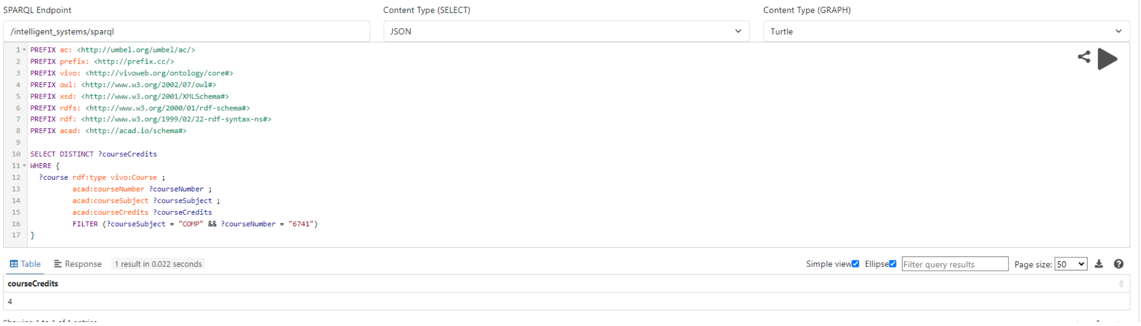
content	class
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_2%5CSlides%5CChapter_2.pdf	http://acad.io/schema#Slides
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_2%5CReadings%5CWorksheet2.pdf	http://acad.io/schema#Reading

6. How many credits is [course] [number] worth?

In [7]:
```
# Define the SPARQL query
course_subject = "COMP"
course_number = "6741"

sparql_query = f"""
```

```
PREFIX ac: <http://umbel.org/umbel/ac/>
PREFIX prefix: <http://prefix.cc/>
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?courseCredits
WHERE {{
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject ;
          acad:courseCredits ?courseCredits
          FILTER (?courseSubject = "{course_subject}" && ?courseNumber = "{course_n
}}
"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
print(json.dumps(response.json(), indent=1))
```

```
{
 "head": {
  "vars": [
   "courseCredits"
  ]
 },
 "results": {
  "bindings": [
   {
    "courseCredits": {
     "type": "literal",
     "value": "4"
    }
   }
  ]
 }
}
```



```
PREFIX ac: <http://umbel.org/umbel/ac/>
PREFIX prefix: <http://prefix.cc/>
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?courseCredits
WHERE {
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject ;
          acad:courseCredits ?courseCredits
          FILTER (?courseSubject = "COMP" && ?courseNumber = "6741")
}
```

courseCredits
4

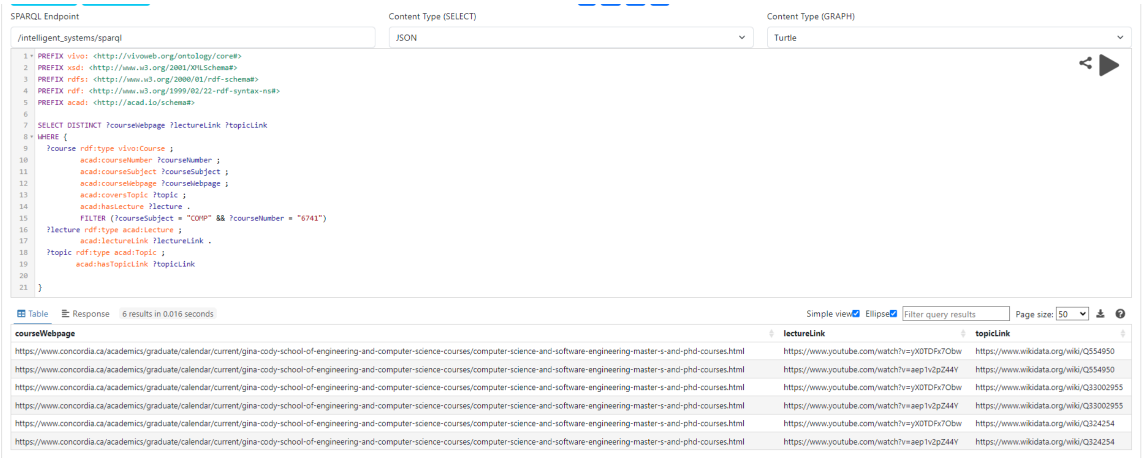
7. For [course] [number], what additional resources (links to web pages) are available

In [24]:

```python
# Define the SPARQL query
course_subject = "COMP"
course_number = "6741"

sparql_query = f"""
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?courseWebpage ?lectureLink ?topicLink
WHERE {{
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject ;
          acad:courseWebpage ?courseWebpage ;
          acad:coversTopic ?topic ;
          acad:hasLecture ?lecture .
          FILTER (?courseSubject = "{course_subject}" && ?courseNumber = "{course_n
  ?lecture rdf:type acad:Lecture ;
          acad:lectureLink ?lectureLink .
  ?topic rdf:type acad:Topic ;
         acad:hasTopicLink ?topicLink


}}
"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
# Print the response
```

```
print(len(response.json()['results']['bindings']))
print(response.json()['results']['bindings'][:5])
```

```
6
[{'courseWebpage': {'type': 'literal', 'value': 'https://www.concordia.ca/academics/
graduate/calendar/current/gina-cody-school-of-engineering-and-computer-science-cours
es/computer-science-and-software-engineering-master-s-and-phd-courses.html'}, 'lectu
reLink': {'type': 'literal', 'value': 'https://www.youtube.com/watch?v=yX0TDFx7Ob
w'}, 'topicLink': {'type': 'literal', 'value': 'https://www.wikidata.org/wiki/Q55495
0'}}, {'courseWebpage': {'type': 'literal', 'value': 'https://www.concordia.ca/acade
mics/graduate/calendar/current/gina-cody-school-of-engineering-and-computer-science-
courses/computer-science-and-software-engineering-master-s-and-phd-courses.html'},
'lectureLink': {'type': 'literal', 'value': 'https://www.youtube.com/watch?v=aep1v2p
Z44Y'}, 'topicLink': {'type': 'literal', 'value': 'https://www.wikidata.org/wiki/Q55
4950'}}, {'courseWebpage': {'type': 'literal', 'value': 'https://www.concordia.ca/ac
ademics/graduate/calendar/current/gina-cody-school-of-engineering-and-computer-scien
ce-courses/computer-science-and-software-engineering-master-s-and-phd-courses.htm
l'}, 'lectureLink': {'type': 'literal', 'value': 'https://www.youtube.com/watch?v=yX
0TDFx7Obw'}, 'topicLink': {'type': 'literal', 'value': 'https://www.wikidata.org/wik
i/Q33002955'}}, {'courseWebpage': {'type': 'literal', 'value': 'https://www.concordi
a.ca/academics/graduate/calendar/current/gina-cody-school-of-engineering-and-compute
r-science-courses/computer-science-and-software-engineering-master-s-and-phd-course
s.html'}, 'lectureLink': {'type': 'literal', 'value': 'https://www.youtube.com/watc
h?v=aep1v2pZ44Y'}, 'topicLink': {'type': 'literal', 'value': 'https://www.wikidata.o
rg/wiki/Q33002955'}}, {'courseWebpage': {'type': 'literal', 'value': 'https://www.co
ncordia.ca/academics/graduate/calendar/current/gina-cody-school-of-engineering-and-c
omputer-science-courses/computer-science-and-software-engineering-master-s-and-phd-c
ourses.html'}, 'lectureLink': {'type': 'literal', 'value': 'https://www.youtube.com/
watch?v=yX0TDFx7Obw'}, 'topicLink': {'type': 'literal', 'value': 'https://www.wikida
ta.org/wiki/Q324254'}}]
```



SPARQL Endpoint: /intelligent_systems/sparql

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

```
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?courseWebpage ?lectureLink ?topicLink
WHERE {
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject ;
          acad:courseWebpage ?courseWebpage ;
          acad:coversTopic ?topic ;
          acad:hasLecture ?lecture .
          FILTER (?courseSubject = "COMP" && ?courseNumber = "6741")
  ?lecture rdf:type acad:Lecture ;
          acad:lectureLink ?lectureLink .
  ?topic rdf:type acad:Topic ;
         acad:hasTopicLink ?topicLink

}
```

courseWebpage	lectureLink	topicLink
https://www.concordia.ca/academics/graduate/calendar/current/gina-cody-school-of-engineering-and-computer-science-courses/computer-science-and-software-engineering-master-s-and-phd-courses.html	https://www.youtube.com/watch?v=yX0TDFx7Obw	https://www.wikidata.org/wiki/Q554950
https://www.concordia.ca/academics/graduate/calendar/current/gina-cody-school-of-engineering-and-computer-science-courses/computer-science-and-software-engineering-master-s-and-phd-courses.html	https://www.youtube.com/watch?v=aep1v2pZ44Y	https://www.wikidata.org/wiki/Q554950
https://www.concordia.ca/academics/graduate/calendar/current/gina-cody-school-of-engineering-and-computer-science-courses/computer-science-and-software-engineering-master-s-and-phd-courses.html	https://www.youtube.com/watch?v=yX0TDFx7Obw	https://www.wikidata.org/wiki/Q33002955
https://www.concordia.ca/academics/graduate/calendar/current/gina-cody-school-of-engineering-and-computer-science-courses/computer-science-and-software-engineering-master-s-and-phd-courses.html	https://www.youtube.com/watch?v=aep1v2pZ44Y	https://www.wikidata.org/wiki/Q33002955
https://www.concordia.ca/academics/graduate/calendar/current/gina-cody-school-of-engineering-and-computer-science-courses/computer-science-and-software-engineering-master-s-and-phd-courses.html	https://www.youtube.com/watch?v=yX0TDFx7Obw	https://www.wikidata.org/wiki/Q324254
https://www.concordia.ca/academics/graduate/calendar/current/gina-cody-school-of-engineering-and-computer-science-courses/computer-science-and-software-engineering-master-s-and-phd-courses.html	https://www.youtube.com/watch?v=aep1v2pZ44Y	https://www.wikidata.org/wiki/Q324254

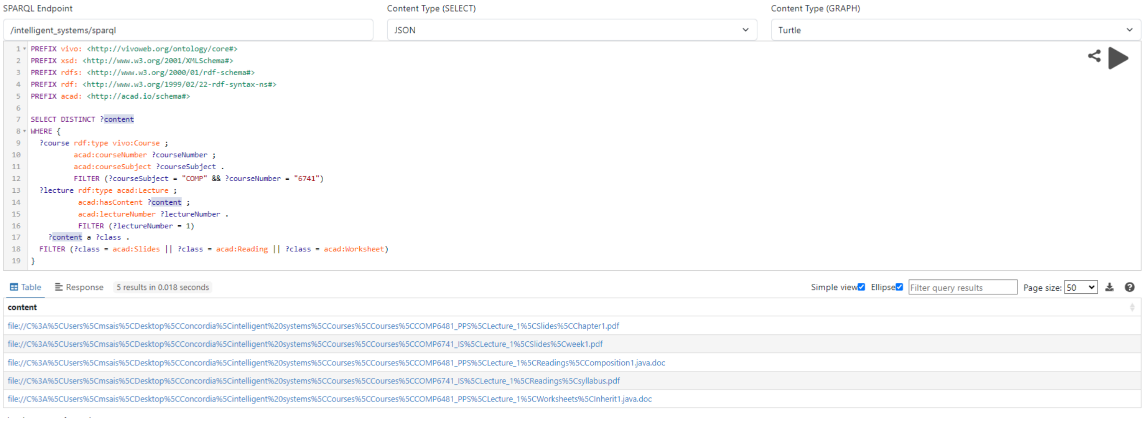
8. Detail the content (slides, worksheets, readings) available for [lecture number] in [course] [number].

In [9]:

```
# Define the SPARQL query
course_subject = "COMP"
course_number = "6741"
lecture_number = 1

sparql_query = f"""
```

```
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?content
WHERE {{
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject .
          FILTER (?courseSubject = "{course_subject}" && ?courseNumber = "{course_n
  ?lecture rdf:type acad:Lecture ;
           acad:hasContent ?content ;
                   acad:lectureNumber ?lectureNumber .
                   FILTER (?lectureNumber = {lecture_number})
    ?content a ?class .
  FILTER (?class = acad:Slides || ?class = acad:Reading || ?class = acad:Worksheet)
}}
"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
print(json.dumps(response.json(), indent=1))
```

```
{
 "head": {
  "vars": [
   "content"
  ]
 },
 "results": {
  "bindings": [
   {
    "content": {
     "type": "uri",
     "value": "file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20sys
tems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_1%5CSlides%5CChapter1.pdf"
    }
   },
   {
    "content": {
     "type": "uri",
     "value": "file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20sys
tems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_1%5CSlides%5Cweek1.pdf"
    }
   },
   {
    "content": {
     "type": "uri",
     "value": "file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20sys
tems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_1%5CReadings%5CComposition1.java.d
oc"
    }
   },
   {
    "content": {
     "type": "uri",
     "value": "file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20sys
tems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_1%5CReadings%5Csyllabus.pdf"
    }
   },
   {
    "content": {
     "type": "uri",
     "value": "file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20sys
tems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_1%5CWorksheets%5CInherit1.java.do
c"
    }
   }
  ]
 }
}
```



/intelligent_systems/sparql

JSON

Turtle

```
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?content
WHERE {
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject .
          FILTER (?courseSubject = "COMP" && ?courseNumber = "6741")
  ?lecture rdf:type acad:Lecture ;
           acad:hasContent ?content ;
           acad:lectureNumber ?lectureNumber .
           FILTER (?lectureNumber = 1)
    ?content a ?class .
  FILTER (?class = acad:Slides || ?class = acad:Reading || ?class = acad:Worksheet)
}
```

5 results in 0.018 seconds

Simple view☑ Ellipse☑ Filter query results Page size: 50

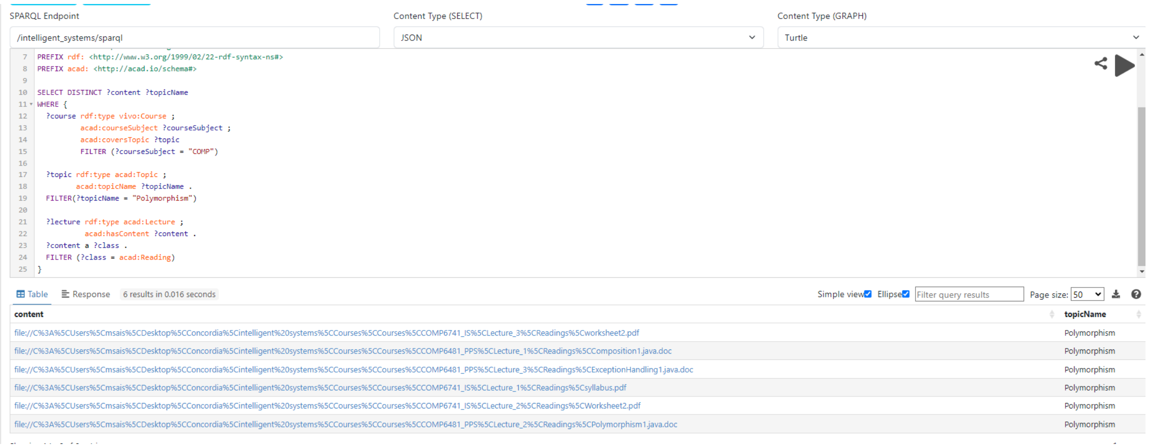
content
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_1%5CSlides%5CChapter1.pdf
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_1%5CSlides%5Cweek1.pdf
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_1%5CReadings%5CComposition1.java.doc
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_1%5CReadings%5Csyllabus.pdf
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_1%5CWorksheets%5CInherit1.java.doc

9. What reading materials are recommended for studying [topic] in [course]?

In [10]:
```
# Define the SPARQL query
course_subject = "COMP"
topic_name = "Polymorphism"

sparql_query = f"""
PREFIX ac: <http://umbel.org/umbel/ac/>
PREFIX prefix: <http://prefix.cc/>
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?content ?topicName
WHERE {{
  ?course rdf:type vivo:Course ;
          acad:courseSubject ?courseSubject ;
          acad:coversTopic ?topic
          FILTER (?courseSubject = "{course_subject}")

  ?topic rdf:type acad:Topic ;
         acad:topicName ?topicName .
  FILTER(?topicName = "{topic_name}")

  ?lecture rdf:type acad:Lecture ;
           acad:hasContent ?content .
  ?content a ?class .
  FILTER (?class = acad:Reading)
}}
"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}
```

```
# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
print(response.json())
```

```
{'head': {'vars': ['content', 'topicName']}, 'results': {'bindings': [{'content':
{'type': 'uri', 'value': 'file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintell
igent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_3%5CReadings%5Cworksheet
2.pdf'}, 'topicName': {'type': 'literal', 'value': 'Polymorphism'}}, {'content': {'t
ype': 'uri', 'value': 'file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintellige
nt%20systems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_1%5CReadings%5CComposition
1.java.doc'}, 'topicName': {'type': 'literal', 'value': 'Polymorphism'}}, {'conten
t': {'type': 'uri', 'value': 'file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cin
telligent%20systems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_3%5CReadings%5CExce
ptionHandling1.java.doc'}, 'topicName': {'type': 'literal', 'value': 'Polymorphis
m'}}, {'content': {'type': 'uri', 'value': 'file://C%3A%5CUsers%5Cmsais%5CDesktop%5C
Concordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_1%5CRe
adings%5Csyllabus.pdf'}, 'topicName': {'type': 'literal', 'value': 'Polymorphism'}},
{'content': {'type': 'uri', 'value': 'file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcor
dia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_2%5CReading
s%5CWorksheet2.pdf'}, 'topicName': {'type': 'literal', 'value': 'Polymorphism'}},
{'content': {'type': 'uri', 'value': 'file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcor
dia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_2%5CReading
s%5CPolymorphism1.java.doc'}, 'topicName': {'type': 'literal', 'value': 'Polymorphis
m'}}]}}
```



SPARQL Endpoint: /intelligent_systems/sparql

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?content ?topicName
WHERE {
  ?course rdf:type vivo:Course ;
          acad:courseSubject ?courseSubject ;
          acad:coversTopic ?topic
          FILTER (?courseSubject = "COMP")

  ?topic rdf:type acad:Topic ;
         acad:topicName ?topicName .
  FILTER(?topicName = "Polymorphism")

  ?lecture rdf:type acad:Lecture ;
           acad:hasContent ?content .
  ?content a ?class .
  FILTER (?class = acad:Reading)
}
```

6 results in 0.016 seconds

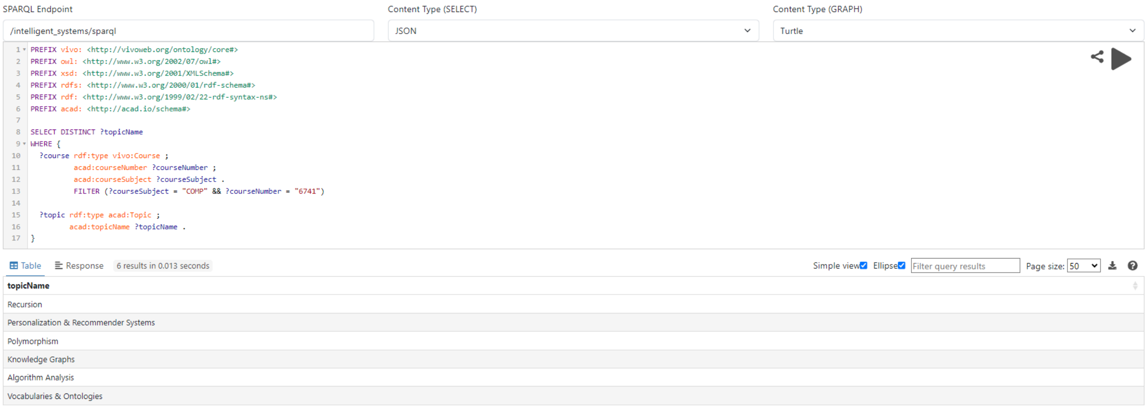
content	topicName
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_3%5CReadings%5Cworksheet2.pdf	Polymorphism
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_1%5CReadings%5CComposition1.java.doc	Polymorphism
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_3%5CReadings%5CExceptionHandling1.java.doc	Polymorphism
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_1%5CReadings%5Csyllabus.pdf	Polymorphism
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6741_IS%5CLecture_2%5CReadings%5CWorksheet2.pdf	Polymorphism
file://C%3A%5CUsers%5Cmsais%5CDesktop%5CConcordia%5Cintelligent%20systems%5CCourses%5CCourses%5CCOMP6481_PPS%5CLecture_2%5CReadings%5CPolymorphism1.java.doc	Polymorphism

10. What competencies [topics] does a student gain after completing [course] [number]?

In [11]:

```
# Define the SPARQL query
course_subject = "COMP"
course_number = "6741"

sparql_query = f"""
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?topicName
WHERE {{
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject .
          FILTER (?courseSubject = "{course_subject}" && ?courseNumber = "{course_n

  ?topic rdf:type acad:Topic ;
         acad:topicName ?topicName .
}}
"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
print(json.dumps(response.json(), indent=1))
```

```
{
 "head": {
  "vars": [
   "topicName"
  ]
 },
 "results": {
  "bindings": [
   {
    "topicName": {
     "type": "literal",
     "value": "Recursion"
    }
   },
   {
    "topicName": {
     "type": "literal",
     "value": "Personalization & Recommender Systems"
    }
   },
   {
    "topicName": {
     "type": "literal",
     "value": "Polymorphism"
    }
   },
   {
    "topicName": {
     "type": "literal",
     "value": "Knowledge Graphs"
    }
   },
   {
    "topicName": {
     "type": "literal",
     "value": "Algorithm Analysis"
    }
   },
   {
    "topicName": {
     "type": "literal",
     "value": "Vocabularies & Ontologies"
    }
   }
  ]
 }
}
```



SPARQL Endpoint: /intelligent_systems/sparql

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

```
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?topicName
WHERE {
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject .
          FILTER (?courseSubject = "COMP" && ?courseNumber = "6741")

  ?topic rdf:type acad:Topic ;
         acad:topicName ?topicName .
}
```

Table | Response | 6 results in 0.013 seconds

Simple view☑ Ellipse☑ Filter query results Page size: 50

topicName
Recursion
Personalization & Recommender Systems
Polymorphism
Knowledge Graphs
Algorithm Analysis
Vocabularies & Ontologies

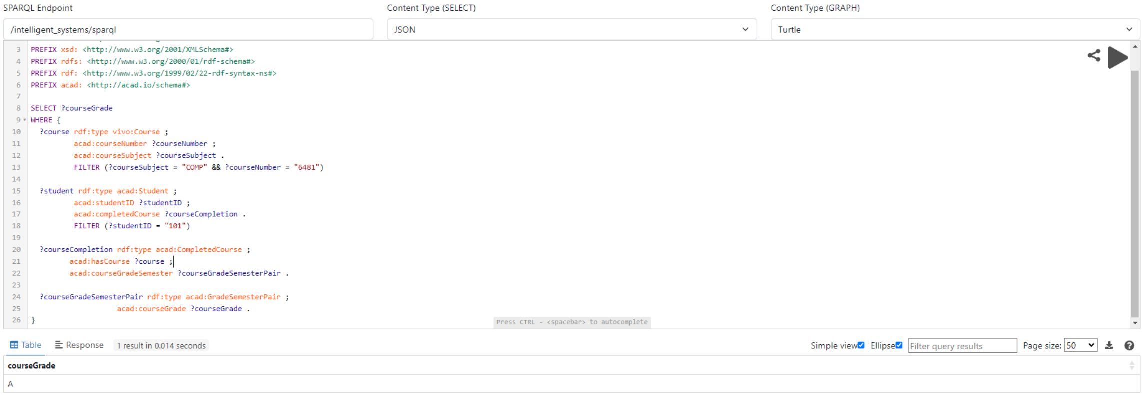
11. What grades did [student] achieve in [course] [number]?

In [12]:

```
# Define the SPARQL query
course_subject = "COMP"
course_number = "6741"
student_id = "101"

sparql_query = f"""
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT ?courseGrade
WHERE {{
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject .
          FILTER (?courseSubject = "{course_subject}" && ?courseNumber = "{course_n

  ?student rdf:type acad:Student ;
          acad:studentID ?studentID ;
                  acad:completedCourse ?courseCompletion .
                  FILTER (?studentID = "{student_id}")

  ?courseCompletion rdf:type acad:CompletedCourse ;
         acad:hasCourse ?course ;
                 acad:courseGradeSemester ?courseGradeSemesterPair .

  ?courseGradeSemesterPair rdf:type acad:GradeSemesterPair ;
                    acad:courseGrade ?courseGrade .
}}

"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'
```

```python
# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
print(json.dumps(response.json(), indent=1))
```

```
{
 "head": {
  "vars": [
   "courseGrade"
  ]
 },
 "results": {
  "bindings": [
   {
    "courseGrade": {
     "type": "literal",
     "value": "A+"
    }
   }
  ]
 }
}
```



SPARQL Endpoint: /intelligent_systems/sparql

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

```sparql
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT ?courseGrade
WHERE {
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject .
          FILTER (?courseSubject = "COMP" && ?courseNumber = "6481")

  ?student rdf:type acad:Student ;
          acad:studentID ?studentID ;
          acad:completedCourse ?courseCompletion .
          FILTER (?studentID = "101")

  ?courseCompletion rdf:type acad:CompletedCourse ;
         acad:hasCourse ?course ;
         acad:courseGradeSemester ?courseGradeSemesterPair .

  ?courseGradeSemesterPair rdf:type acad:GradeSemesterPair ;
                acad:courseGrade ?courseGrade .
}
```

1 result in 0.014 seconds

Simple view☑ Ellipse☑ Filter query results Page size: 50

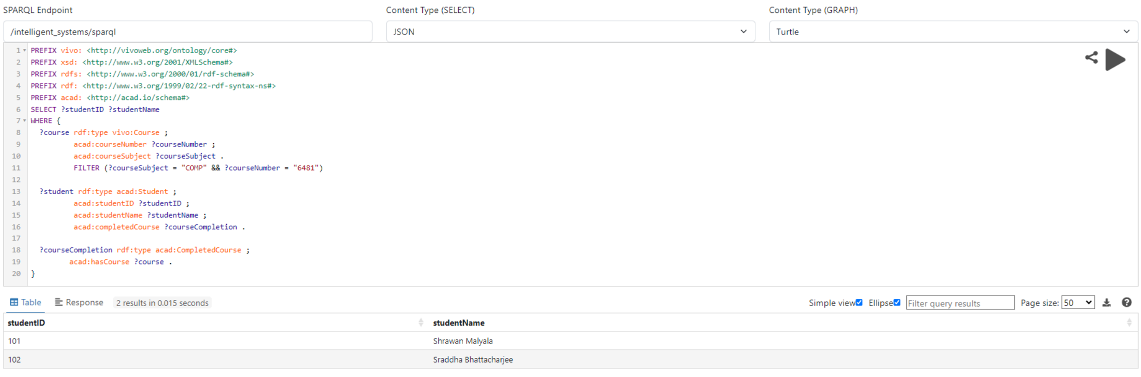
courseGrade
A

12. Which [students] have completed [course] [number]?

In [13]:
```python
# Define the SPARQL query
course_subject = "COMP"
course_number = "6741"

sparql_query = f"""
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>
```

```
SELECT ?studentID ?studentName
WHERE {{
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject .
          FILTER (?courseSubject = "{course_subject}" && ?courseNumber = "{course_n

  ?student rdf:type acad:Student ;
          acad:studentID ?studentID ;
          acad:studentName ?studentName ;
                  acad:completedCourse ?courseCompletion .

  ?courseCompletion rdf:type acad:CompletedCourse ;
         acad:hasCourse ?course .
}}

"""

# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
print(json.dumps(response.json(), indent=1))
```

```
{
 "head": {
  "vars": [
   "studentID",
   "studentName"
  ]
 },
 "results": {
  "bindings": [
   {
    "studentID": {
     "type": "literal",
     "value": "101"
    },
    "studentName": {
     "type": "literal",
     "value": "Shrawan Malyala"
    }
   },
   {
    "studentID": {
     "type": "literal",
     "value": "102"
    },
    "studentName": {
     "type": "literal",
     "value": "Sraddha Bhattacharjee"
    }
   }
  ]
 }
}
```



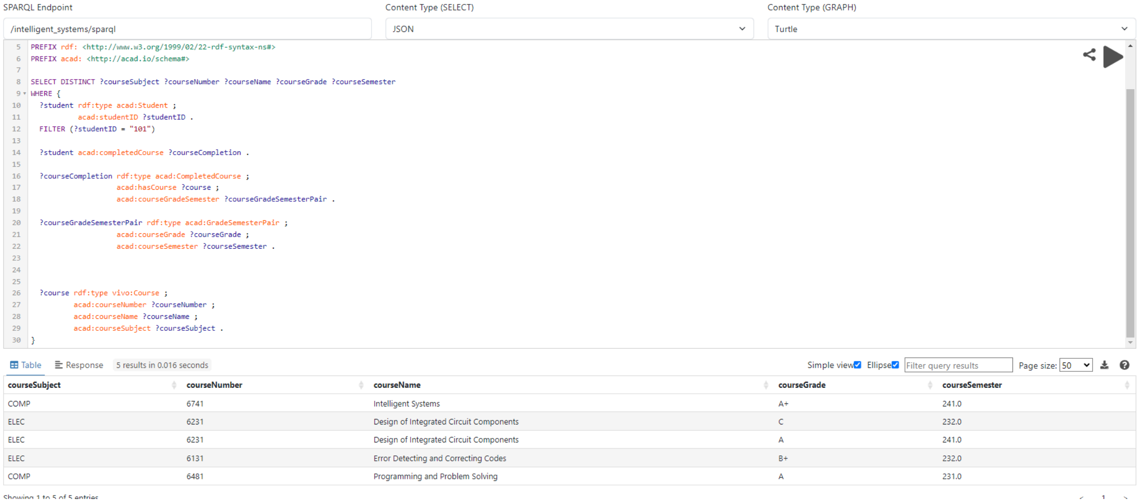
```
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>
SELECT ?studentID ?studentName
WHERE {
  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseSubject ?courseSubject .
          FILTER (?courseSubject = "COMP" && ?courseNumber = "6481")

  ?student rdf:type acad:Student ;
          acad:studentID ?studentID ;
          acad:studentName ?studentName ;
          acad:completedCourse ?courseCompletion .

  ?courseCompletion rdf:type acad:CompletedCourse ;
         acad:hasCourse ?course .
}
```

studentID	studentName
101	Shrawan Malyala
102	Sraddha Bhattacharjee

13. Print a transcript for a [student], listing all the course taken with their grades

In [14]:
```
# Define the SPARQL query
student_id = "101"

sparql_query = f"""
PREFIX vivo: <http://vivoweb.org/ontology/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?courseSubject ?courseNumber ?courseName ?courseGrade ?courseSemest
WHERE {{
  ?student rdf:type acad:Student ;
           acad:studentID ?studentID .
  FILTER (?studentID = "{student_id}")

  ?student acad:completedCourse ?courseCompletion .

  ?courseCompletion rdf:type acad:CompletedCourse ;
                    acad:hasCourse ?course ;
                                        acad:courseGradeSemester ?courseGradeSemest

  ?courseGradeSemesterPair rdf:type acad:GradeSemesterPair ;
                    acad:courseGrade ?courseGrade ;
                    acad:courseSemester ?courseSemester .



  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseName ?courseName ;
          acad:courseSubject ?courseSubject .
}}

"""


# Define the endpoint URL
endpoint_url = 'http://localhost:3030/intelligent_systems/sparql'

# Define the payload
payload = {'query': sparql_query}

# Send the POST request
response = requests.post(endpoint_url, data=payload)

# Print the response
print(response.json())
```

```
{'head': {'vars': ['courseSubject', 'courseNumber', 'courseName', 'courseGrade', 'co
urseSemester']}, 'results': {'bindings': [{'courseSubject': {'type': 'literal', 'val
ue': 'COMP'}, 'courseNumber': {'type': 'literal', 'value': '6741'}, 'courseName':
{'type': 'literal', 'value': 'Intelligent Systems'}, 'courseGrade': {'type': 'litera
l', 'value': 'A+'}, 'courseSemester': {'type': 'literal', 'value': '241.0'}}, {'cour
seSubject': {'type': 'literal', 'value': 'ELEC'}, 'courseNumber': {'type': 'litera
l', 'value': '6231'}, 'courseName': {'type': 'literal', 'value': 'Design of Integrat
ed Circuit Components'}, 'courseGrade': {'type': 'literal', 'value': 'C'}, 'courseSe
mester': {'type': 'literal', 'value': '232.0'}}, {'courseSubject': {'type': 'litera
l', 'value': 'ELEC'}, 'courseNumber': {'type': 'literal', 'value': '6231'}, 'courseN
ame': {'type': 'literal', 'value': 'Design of Integrated Circuit Components'}, 'cour
seGrade': {'type': 'literal', 'value': 'A'}, 'courseSemester': {'type': 'literal',
'value': '241.0'}}, {'courseSubject': {'type': 'literal', 'value': 'ELEC'}, 'courseN
umber': {'type': 'literal', 'value': '6131'}, 'courseName': {'type': 'literal', 'val
ue': 'Error Detecting and Correcting Codes'}, 'courseGrade': {'type': 'literal', 'va
lue': 'B+'}, 'courseSemester': {'type': 'literal', 'value': '232.0'}}, {'courseSubje
ct': {'type': 'literal', 'value': 'COMP'}, 'courseNumber': {'type': 'literal', 'valu
e': '6481'}, 'courseName': {'type': 'literal', 'value': 'Programming and Problem Sol
ving'}, 'courseGrade': {'type': 'literal', 'value': 'A'}, 'courseSemester': {'type':
'literal', 'value': '231.0'}}]}}
```



SPARQL Endpoint: /intelligent_systems/sparql

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX acad: <http://acad.io/schema#>

SELECT DISTINCT ?courseSubject ?courseNumber ?courseName ?courseGrade ?courseSemester
WHERE {
  ?student rdf:type acad:Student ;
           acad:studentID ?studentID .
  FILTER (?studentID = "101")

  ?student acad:completedCourse ?courseCompletion .

  ?courseCompletion rdf:type acad:CompletedCourse ;
                    acad:hasCourse ?course ;
                    acad:courseGradeSemester ?courseGradeSemesterPair .

  ?courseGradeSemesterPair rdf:type acad:GradeSemesterPair ;
                    acad:courseGrade ?courseGrade ;
                    acad:courseSemester ?courseSemester .



  ?course rdf:type vivo:Course ;
          acad:courseNumber ?courseNumber ;
          acad:courseName ?courseName ;
          acad:courseSubject ?courseSubject .
}
```

courseSubject	courseNumber	courseName	courseGrade	courseSemester
COMP	6741	Intelligent Systems	A+	241.0
ELEC	6231	Design of Integrated Circuit Components	C	232.0
ELEC	6231	Design of Integrated Circuit Components	A	241.0
ELEC	6131	Error Detecting and Correcting Codes	B+	232.0
COMP	6481	Programming and Problem Solving	A	231.0

Showing 1 to 5 of 5 entries

In [ ]: