# Assessment 6: Ontology Design

**Step-by-Step Instructions with Comments in Python Code**

# Step 1: Install and import the required library for ontology management

# We'll use `rdflib` for building and querying ontologies

!pip install rdflib

from rdflib import Graph, Literal, RDF, URIRef

from rdflib.namespace import RDFS, OWL

# Step 2: Initialize a graph for the ontology

g = Graph()

# Define namespaces

ontology = URIRef("http://example.org/ontology#")

animal = URIRef(ontology + "Animal")

mammal = URIRef(ontology + "Mammal")

bird = URIRef(ontology + "Bird")

dog = URIRef(ontology + "Dog")

eagle = URIRef(ontology + "Eagle")

# Step 3: Add entities and relationships

# Adding classes

g.add((animal, RDF.type, OWL.Class))

g.add((mammal, RDF.type, OWL.Class))

g.add((bird, RDF.type, OWL.Class))

# Adding relationships

g.add((mammal, RDFS.subClassOf, animal))

g.add((bird, RDFS.subClassOf, animal))

g.add((dog, RDF.type, mammal))

g.add((eagle, RDF.type, bird))

# Step 4: Query the ontology

# Query for all entities of type Animal

query = """

SELECT ?entity WHERE {

?entity rdf:type ?type .

?type rdfs:subClassOf <http://example.org/ontology#Animal> .

}

"""

results = g.query(query)

# Print results

for result in results:

print(f"Entity: {result['entity']}")

# Step 5: Save the ontology to a file

g.serialize(destination="ontology.ttl", format="turtle")