UGANDA CHRISTIAN UNIVERSITY
A Centre of Excellence in the Heart of Africa

FACULTY OF ENGINEERING, DESIGN AND TECHNOLOGY
DEPARTMENT OF COMPUTING AND TECHNOLOGY
ADVENT 2025 SEMESTER **EXAMINATION**

PROGRAM: *BACHELOR OF SCIENCE IN DATA SCIENCE & ANALYTICS*

YEAR: 3 SEMESTER: 1

COURSE CODE: DSC3113

COURSE NAME: *KNOWLEDGE ENGINEERING*

EXAMINATION TYPE: *PROJECT-BASED EXAM*

PROJECT DURATION: DEC 2025

TIME ALLOWED: Two weeks

Examination Instructions

1. The general Uganda Christian University examination guidelines and academic & financial policies apply to this examination. Violating any of the policies by the student automatically makes this examination attempt void, even if you have completed and submitted the answer booklet.
2. This exam consists of a project to be executed in Three weeks.
    i. Assessment of the project shall be based on five milestones, evaluated during the duration of the project. Each milestone shall be evaluated out of 20 marks.
3. Every student has a responsibility to prove their contribution towards every milestone, and marks may be awarded to every student individually.

PART A:
PROJECT DESCRIPTION

PROJECT OVERVIEW

This examination assesses your ability to design and implement **hybrid intelligent systems** that integrate **Knowledge Engineering (KE)** and **Machine Learning (ML)** or **Deep Learning (DL)**. Each group will select **ONE (1)** project from the three provided options. You will be expected to demonstrate the ability to:

- Build **ontologies and rule bases**,
- Implement **ML/DL models**,
- Integrate both approaches into a working **hybrid AI system**, and
- Evaluate and communicate your findings clearly.

The project should be completed collaboratively using **Python** and **Jupyter Notebook**, supported by appropriate **ontology tools**.

PROJECT OPTIONS (CHOOSE ONE)

GROUP 1 PROJECT:

Hybrid Healthcare Symptom & Diagnosis Assistant

1. Background and Purpose

Healthcare diagnosis often depends on both expert medical knowledge and data-driven pattern recognition. This project challenges students to design a **hybrid AI system** that combines **Knowledge Engineering (ontology and rules)** with **Machine Learning (ML)** for medical symptom checking and preliminary disease diagnosis.
The system should reason using encoded medical knowledge and integrate ML predictions for enhanced diagnostic accuracy.

2. Dataset

1. **Symptom-Disease Dataset:** https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset
2. **Optional Image Dataset (extension):** Chest X-ray Pneumonia dataset: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

3. Tasks

Part A – Knowledge Engineering

- Build a **medical ontology** connecting diseases, symptoms, and recommended treatments.
- Encode **at least 20 diagnostic rules** (e.g., *If cough + fever + chest pain → pneumonia*).
- Implement a **reasoning engine** (forward or backward chaining) to infer likely diseases.

Part B – Machine Learning / Deep Learning

- Train a **machine learning classifier** to predict diseases from structured symptom datasets.
- (Optional) Implement a CNN to classify X-ray images (Normal vs Pneumonia).
- Integrate ML/DL predictions with the KE reasoning process for hybrid decision-making.

Part C – Integration & Evaluation

- Demonstrate how the rule engine and ML model interact to reach final diagnoses.
- Evaluate system accuracy, explain misclassifications, and provide **3 test case scenarios**.

4. Expected Output

- Medical ontology and rule base file.
- Trained ML model integrated into reasoning framework.
- Executable notebook with reasoning demonstration.
- Evaluation metrics and confusion matrix.
- Report and screenshots of system results.

GROUP 2 PROJECT:

Smart Agriculture Advisor for Crops & Pests

1. Background and Purpose

Farmers often struggle with identifying crop diseases and pests early enough to prevent major yield loss. This project requires students to develop a **hybrid intelligent system** that combines **Knowledge Engineering (rules, ontology)** with **Deep Learning (CNN)** to recognize crop diseases and recommend treatments.
The system integrates **symbolic reasoning** for recommendations with **image-based recognition** for disease detection.

2. Dataset

3. **PlantVillage Crop Disease Dataset:** https://www.kaggle.com/datasets/emmarex/plantdisease
4. **Alternative Dataset (for specific crops):** Cassava Leaf Disease Dataset: https://www.kaggle.com/competitions/cassava-disease

3. Tasks

Part A – Knowledge Engineering

- Construct an **agriculture ontology** connecting crops, pests, diseases, symptoms, and treatments.
- Encode **at least 20 expert rules** (e.g., *If maize has yellow streaks + stunted growth → Maize Streak Virus*).
- Implement reasoning to provide treatment advice based on observed conditions.

Part B – Machine Learning / Deep Learning

- Train a **CNN model** to classify crop diseases from leaf images.
- Optionally extend with yield prediction using regression.
- Integrate CNN predictions with the rule base for final advice output.

Part C – Integration & Evaluation

- Demonstrate integration of the reasoning system with CNN outputs.
- Evaluate the CNN's accuracy and the reasoning system's correctness.
- Present **3 case studies** showing hybrid predictions and recommendations.

4. Expected Output

- Agriculture ontology and rules file.
- CNN model file and notebook demonstrating integration.
- Evaluation summary with visuals and metrics.
- One-page poster and short demo video.

GROUP 3 PROJECT:

University Chatbot with Knowledge Graph and Machine Learning

1. Background and Purpose

Universities handle a wide range of student queries—about admissions, programs, courses, and campus services—that can be automated using AI. This project challenges students to develop a **hybrid chatbot system** that combines a **Knowledge Graph (KE)** with **Machine Learning (ML)** to understand and respond intelligently to student queries.
The chatbot should reason over a university knowledge base while also using ML or NLP models to interpret natural language questions.

2. Dataset

5. **University FAQs or Student Query Dataset:** https://www.kaggle.com/datasets/keivalya/student-queries-dataset
6. **Alternative FAQ Dataset:** https://www.kaggle.com/datasets/narendrageek/faq-dataset

3. Tasks

Part A – Knowledge Engineering

- Develop a **University Knowledge Graph** (e.g., in Protégé or Neo4j) linking entities such as faculties, programs, courses, staff, locations, and services.
- Encode **at least 20 rules or SPARQL queries** for question-answering (e.g., *If user asks "admissions deadline" → retrieve from Admission entity*).
- Implement **rule-based reasoning** or ontology querying (using Owlready2, RDFLib, or Neo4j).

Part B – Machine Learning / NLP

- Train an **intent classification model** using a student-query dataset (e.g., logistic regression, SVM, or fine-tuned BERT).
- Implement **text preprocessing, tokenization, and vectorization** (TF-IDF or embeddings).
- Integrate ML model predictions with the Knowledge Graph to produce hybrid responses.

Part C – Integration & Evaluation

- Show how the ML intent recognizer triggers the correct ontology or rule queries.
- Evaluate chatbot accuracy (precision, recall, F1) using sample queries.
- Provide **3 dialogue examples** demonstrating correct and incorrect responses.

4. Expected Output

- Knowledge Graph (OWL/RDF/Neo4j format).
- Trained NLP model integrated into chatbot logic.
- Jupyter Notebook showcasing hybrid question answering.
- Evaluation metrics and error discussion.
- Report and poster summarizing chatbot architecture and findings.

NOTEBOOK STRUCTURE (FOR ALL PROJECTS)

7. **Introduction** – Problem statement, objectives, and relevance.
8. **Knowledge Engineering Setup** – Ontology/rule base construction and reasoning.
9. **Machine Learning / Deep Learning Setup** – Model design, training, and testing.
10. **Hybrid Integration** – KE + ML system pipeline and workflow explanation.
11. **Evaluation** – Accuracy metrics and reasoning validation.
12. **Results and Discussion** – Key findings, strengths, and limitations.
13. **Conclusion and Recommendations** – Summary and future work.

TOOLS & TECHNOLOGIES

- **Python 3.x**, **Jupyter Notebook**
- **Protégé**, **Owlready2**, **RDFLib**, or **Neo4j**
- **scikit-learn**, **TensorFlow/Keras**, **PyTorch**
- **spaCy**, **NLTK**, or **HuggingFace Transformers**
- **Pandas**, **NumPy**, **Matplotlib**, **Seaborn**
- **Gradio** or **Streamlit** for simple demos

LEARNING OUTCOMES (BLOOM'S TAXONOMY)

Level	Outcome
Apply	Build hybrid systems integrating knowledge-based reasoning and ML models.
Analyze	Examine and compare the reasoning flow and model outputs.
Evaluate	Assess hybrid system performance and justify architectural decisions.
Create	Develop and present a fully functional hybrid AI prototype solving a real-world problem.

SUBMISSION FORMAT

Submit one ZIP file named: GroupX_HybridIntelligentSystem_Project.zip
Contents:
ontology.owl / knowledge_graph.rdf
project_notebook.ipynb
trained_model.pt / .h5
report.pdf
poster.pdf
demo_video_link.txt

PART B: Project-based assessment guidelines

S/N	Milestone Description	Maximum Marks
1	**MILESTONE ONE: Project Definition and Design** Define the hybrid AI problem, objectives, and datasets; design the system architecture showing how Knowledge Engineering (ontology and rules) will integrate with ML/DL components. Assessed on creativity, clarity, and feasibility.	**20 %**
2	**MILESTONE TWO: Knowledge Base and Model Development:** Build the ontology or knowledge graph, encode at least 20 reasoning rules, preprocess data, and develop the ML/DL model. Evaluates technical competence and system-building ability.	**20 %**
3	**MILESTONE THREE: Hybrid Integration and Analysis** Integrate the KE and ML components into one pipeline, test their interoperability, and evaluate performance using appropriate metrics and reasoning accuracy.	**20 %**
4	**MILESTONE FOUR: Presentation, Demonstration, and Defense** Present and defend the system through a poster, oral demo, or video showing workflow, use cases, and real-world applications. Assessed on clarity and teamwork.	**20 %**
5	**MILESTONE FIVE: Report Writing and Documentation** Submit a structured report and clean Jupyter Notebook detailing design, methods, results, and references with clear explanations and citations.	**20 %**
	TOTAL MARKS	100 %

~END OF EXAM GUIDELINES~