COMP 474/6741
Intelligent Systems
Department of Computer Science and Software Engineering
Winter 2024

Course Instructor: Dr. René Witte, Associate Professor
Email: rene.witte@concordia.ca

Office Hours:
Every Thursday, 16:00-17:00, in ER 933 (starting Week 2).
Online meetings (via Zoom) can also be scheduled on-demand via Moodle.
You can also reach me on Moodle chat (works best for short, quick questions).

Labs: Please see your class schedule for details
    Lab Demonstrators:
    Ms. Alejandra Zambrano
    Mr. Amin Karimi
    Mr. Farzad Salajegheh

Course Calendar Description:

(COMP 474): Rule-based expert systems, blackboard architecture, and agent-based. Knowledge acquisition and representation. Uncertainty and conflict resolution. Reasoning and explanation. Design of intelligent systems. Project.

(COMP 6741): Knowledge representation and reasoning. Uncertainty and conflict resolution. Design of intelligent systems. Grammar-based, rule-based, and blackboard architectures. A project is required.

Component(s): Lecture 3 hours per week; Laboratory 2 hours per week

Prerequisites: COMP 352 or COEN 352 *(applies to COMP 474 only)*
Co-requisites: n/a

Specific Knowledge and Skills Needed for this Course:
To succeed in this course, you will need specific skills and knowledge in the following areas:

Programming Skills:

- **Python**: Proficiency in general programming using Python is essential for lab sessions and the course project. A brief introduction to Python will be provided during the first lab session, but pre-existing familiarity is strongly recommended.

- **APIs and Data Handling**: Basic understanding of working with APIs and data formats (like JSON, XML) is important, especially for labs and the project involving knowledge graphs, NLP applications, and interfacing with Deep Learning frameworks.

Mathematical Foundations:

- **Linear Algebra**: Concepts from linear algebra will be utilized frequently, especially in machine learning and neural network modules.

If you find yourself lacking in any of these areas, you are strongly encouraged to review these topics before the DNE (Did Not Enter) deadline.

Course materials
There is no single textbook that will be used for this course. For each lecture topic, required and recommended readings will be posted as part of the lecture information on Moodle. Generally, these readings will be available online or as an electronic resource through the Concordia Library.

Grading Scheme
Your grade will be based on exams and a team project (split into multiple submissions). The grading distribution of these deliverables is as follows:

35%	Course Project (team work, split into 2-3 submissions)
15%	Midterm Exam
50%	Final Exam

Letter grades for this course will be assigned based on a curve, which means they will be determined in relation to the overall class performance. There is no predetermined scale for converting percentages to letter grades, but the curve ensures that grading is fair and accounts for the performance of all students. To **pass the course**, you must achieve at least 50% of the total possible marks.

If your final exam grade (expressed as a percentage) is higher than your midterm exam grade, the midterm exam will be excluded from the calculation of your overall course grade. In this case, the final exam will make up 65% of your overall grade. However, this only applies if you have completed the midterm exam.

Should you fail to write the midterm exam **and** you have valid justification (e.g., doctor's note) then the weight of the missed midterm will be added to the final exam.

<u>Note</u>: It is your responsibility to adhere to the ***university's code of conduct*** as detailed in the calendar. All students must read and sign the <u>Expectations of Originality</u> form and submit the signed copy with their project deliverables.

Tentative Course Schedule
The primary goal of this course is to provide a comprehensive understanding of modern "intelligent" software systems, underpinned by Artificial Intelligence (AI) principles and methodologies. Compared to other courses introducing these concepts, this IS course places a stronger emphasis on the overall design, implementation and evaluation of a complete AI application. In particular, we will examine modern intelligent assistants, including chatbots, and study their foundations in Natural Language Processing, Knowledge Graphs, Personalization, and Deep Learning. The planned weekly schedule (subject to changes) is:

(Week 1) Introduction to Intelligent Systems
(Week 2) Introduction to Knowledge Graphs (RDF)
(Week 3) Knowledge Graphs: Vocabularies & Ontologies (RDFS, OWL)
(Week 4) Knowledge Base Queries (SPARQL) and Linked Open Data (LOD)
(Week 5) Personalization & Recommender Systems
(Week 6) Introduction to Machine Learning (ML) for Intelligent Systems
          Mid-term break: February 25-March 2
(Week 7) *Midterm exam* (tentative!)
(Week 8) Intelligent Agents & Introduction to Natural Language Processing (NLP)
(Week 9) Text Mining Systems
(Week 10) Artificial Neural Networks (ANNs) & Word Embeddings
(Week 11) Introduction to Deep Learning
(Week 12) Deep Learning for Intelligent Systems
          Examination period: April 18-May 1 (date of final exam tbd)

A more detailed week-by-week breakdown, as well as additional information for each topic, will be available on the Moodle web site.

Lab Details
Beginning in Week 2, weekly lab sessions are an obligatory part of this course and will take place in-person and on campus. Make sure to attend the lab section for which you are registered.

Before each lab, it is essential that you review the previous week's lecture material, complete any assigned readings, and review the corresponding lecture worksheet. While the lectures focus on theoretical concepts and algorithms, the labs introduce new programming content not covered in the lectures. These sessions offer hands-on experience with practical applications through various AI libraries and toolkits, such as Python's *scikit-learn* and *PyTorch*.

In summary, the lectures and labs are designed to work in tandem, offering a comprehensive understanding of AI by merging theoretical foundations with practical coding skills.

Active Learning
Starting with the second lecture on *Knowledge Graphs*, we will use *Active Learning* concepts in this course. Active Learning involves engaging with the material, classmates, and instructor in a more interactive way than simply listening to a lecture. In particular, worksheets will be distributed containing mini-exercises that you will solve in teams of two during practise sessions that last a few minutes each. The reasons for using active learning include:

- **Enhanced Retention**: Research shows you remember more when you actively engage with material.

- **Immediate Feedback**: Worksheets and pair discussions allow you to instantly clarify doubts and reinforce understanding.

- **Peer Learning**: Working with classmates exposes you to different perspectives and problem-solving approaches.

If you have never taken a class using active learning, you might have concerns like the following:
- **Feeling Shy?**: No worries! Active learning tasks are designed to be non-judgmental spaces for exploration. You won't be put "on the spot" in front of the class.

- **Don't Want to Interact?**: While collaboration is encouraged, the primary focus is on deepening your own understanding. Active learning can be a personal process too.

Course Learning Outcomes (CLOs):
By the end of this semester, students are expected to master the following skills and areas of knowledge:

Critical Analysis: Develop the ability to critically analyze and assess various AI algorithms and techniques, especially those related to intelligent assistants, natural language processing, and knowledge graphs, evaluating their suitability and efficiency in different application contexts.

Data Interpretation: Enhance skills in preprocessing, interpreting, and utilizing large datasets, with a focus on how these datasets can be effectively leveraged in AI applications, particularly in intelligent systems and natural language processing.

Algorithm Implementation: Gain practical experience in implementing, testing, and refining AI algorithms, with a focus on those relevant to intelligent systems, using a popular programming environment such as Python.

Model Evaluation: Acquire the expertise to critically evaluate the performance of AI models, including those based on deep learning, using a variety of metrics and considering factors specific to intelligent systems and NLP applications.

Integration of AI Components: Develop the ability to integrate various AI components such as Knowledge Graphs, NLP, and Deep Learning into a cohesive system. This outcome focuses on the architectural aspect of AI systems, ensuring students can design systems where these components work synergistically.

Collaboration Skills: Strengthen collaborative problem-solving and project development abilities, particularly in the context of building intelligent systems, through team projects and lab sessions.

Comparative Analysis: Learn to conduct a nuanced comparison of different AI and machine learning methodologies, with an emphasis on their application in the development of intelligent systems, including chatbots and other AI assistants.

Application to Real-World Problems: Gain practical skills in applying AI theories to real-world scenarios, with a particular focus on the challenges and constraints in implementing intelligent systems and the ethical considerations of AI applications.

Health and Safety Guidelines
General health and safety instructions and available health and safety trainings can be found at:
Safety Programs - Concordia University (https://www.concordia.ca/campus-life/safety/general-safety.html)

On Campus Resources
For the services available to Gina Cody School students, please visit Student services at Concordia University (https://www.concordia.ca/ginacody/students/services.html)