Computer Science 384
St. George Campus University of Toronto

Syllabus
Updated: May 6, 2022

Description: In this course, we will cover theories and algorithms that capture (or approximate) some of the core elements of computational intelligence. Assignments provide practical experience, in both theory and programming, of the core topics.

Prerequisites: Although we will cover all relevant background material, it is still expected that you have a basic understanding of algorithms design and probability theory. In particular, you should have taken;

- one of CSC263, CSC265, CSC263, CSCC63, ECE345, ECE358, MIE335
- one of STA237, STA247, STA255, STA257, STA237, STAB57, STAB52, STA286, CHE223, CME263, ECE302, ECE286, MIE231, MIE236, MSE238
- working knowledge of `Python`

If you do not have the appropriate prerequisites, you will need to obtain a prerequisite waiver from the CS undergraduate office (`cs.undergrad@utoronto.ca`). You do not need to email the teaching team. Even if your request is accepted, it is your responsibility to understand the prerequisite material.

Instructor:
Chandra Gummaluru
MASc Computer Engineering - Systems Control
`chandra.gummaluru@mail.utoronto.ca`

Teaching Assistants: Parsa Mirdehghan, Evi Micha, Zhewei Sun, Hongkai Chen

Textbook (Optional): This course has no required textbook. However, if you wish to have a reference, we recommend *Artifcial Intelligence: A Modern Approach (3rd edition)* by Stuart Russel and Peter Norvig.

Lectures: The lectures take place weekly on Mondays[1] from 18:00 - 21:00 in-person in BA1180. Note that although the Arts and Sciences calendar lists Mondays 20:00 - 21:00 for tutorials, we will use it as a lecture hour in the same room. Lectures are not recorded.

Office Hours: Weekly on Tuesdays from 19:00 - 20:00[2] online via Zoom (the link will be made available on the course page). Office hours are not recorded.

Software and Tools: We will use multiple tools for the delivery and collection of material in this course:

- **Quercus**: The syllabus, lecture slides, information about individual evaluations, and practice problems can all be found here.

[1] Starting on May 9th, 2022.
[2] Starting on May 10th, 2022.

- **Piazza**: A discussion form where you can ask questions about anything in the course (including personal concerns). We aim to respond to all inquiries within 24 hours.
- **Markus**: All programming assignments and problem sets should be submitted here. You will find your marks and can submit remark requests for all evaluations here as well.
- **Zoom**: We will use this to hold office hours and help sessions. You will need to use your University of Toronto account.

Links to all of the above will be posted on the course page.

Grading Scheme: This course consists of 3 programming assignments (24%), a mid-term exam (18%), two problem sets (10%) a final exam (36%), and a course presentation (12%). The tentative grade distribution is shown in the table below:

Evaluation	**Weight**	**Release Date**	**Due Date**
Programming Assignment 1: CSPs	8%	May 16	May 25
Programming Assignment 2: Search Problems	8%	May 30	Jun 8
Programming Assignment 3: Games	8%	Jun 13	Jun 20
Mid-Term Test	18%	-	-
Problem Set 1: Uncertainty	5%	Jul 11	Jul 20
Problem Set 2: Knowledge	5%	Jul 25	Aug 3
Course Presentation	12%	Jun 13	Aug 15
Final Exam	36%	-	-

- **All release dates and due dates are tentative**.
- **The deadlines for the programming assignments / problem sets are at 15:00 EST** on their respective due dates which all fall on a Monday.
- The mid-term test will be scheduled between Jun 22 - Jun 29.
- The final exam will be scheduled between Aug 17 - Aug 30.
- **You must obtain a grade of at least 40% on the final exam to pass the course.**
- You may discuss the programming assignments and problem sets with each other but must **submit your own individual work**.
- The course project *must* be done in groups of three students unless this is not possible.
- You will be provided with practice problems for each module. These will not be graded.

Late Policies: You will be given two weeks to complete each programming assignment, and one week to complete each problem set. **Late submissions will not be accepted unless you use grace days.**

- You will be given 5 grace days at the beginning of the term.
- If you use n grace days on an evaluation, you may submit the evaluation up to $24n$ hours beyond the due date without penalty. For example, if the deadline is Jun 13 at 15:00 EST, and you use 2 grace days, your new deadline is Jun 15 at 15:00 EST.

- You do NOT need to notify us when using grace days. This is automatically done by MarkUs.
- You cannot use grace days on the mid-term exam, the final exam, or the course presentation.

Missed Evaluations and Exceptional Circumstances: We understand that things do not always go as planned. In the event of an exceptional circumstance, such as an illness, please contact the instructor as soon as possible so that we are aware of your situation. If you are unable to submit an assignment on time, you should first use your grace days. If you have run out of grace days, we will use the following policy:

- If you miss *one* programming assignment (resp. problem set), its weight will be shifted to the either the mid-term test (resp. final exam).
- If you miss *multiple* programming assignments or problem sets, we will handle this case-by-case.

Remarking Policies: If you feel that you have been marked unfairly for any evaluation, you will have *one week* after the marks have been released to submit a remark request via MarkUs. After the deadline to submit a remark request has passed, we may not be able to consider your request. We aim to have your inquiry responded to within one week.

Academic Integrity: All suspected cases of academic dishonesty will be investigated following procedures outlined in the Code of Behaviour on Academic Matters. If you have questions or concerns about what constitutes appropriate academic behaviour or appropriate research and citation methods, please reach out to me or from other institutional resources (e.g., `www.academicintegrity.utoronto.ca`)

On Equity, Diversity and Inclusion: The University of Toronto is committed to equity, human rights and respect for diversity. All members of the learning environment in this course should strive to create an atmosphere of mutual respect where all members of our community can express themselves, engage with each other, and respect one another's differences. U of T does not condone discrimination or harassment against any persons or communities.

Week	Topics Covered
1 (May 09)	• Search Problems: Formalization – The Components of a Search Problem – The Generalized Search Algorithm – Properties of the Generalized Search Algorithm – Cycle Checking
2 (May 16)	• Constraint Satisfaction Problems – Formalizing a Constraint Satisfaction Problem – Backtracking Search – Constraint Propagation: Forward Checking – Constraint Propagation: Generalized Arc Consistency
3 (May 23)	• Search Problems: Uninformed Search Algorithms – Breadth-First Search and its properties – Depth-First Search and its properties – Uniform-cost Search and its properties – Iterative-Deepening Depth-First Search and its properties
4 (May 30)	• Search Problems: Informed Search Algorithms – Heuristic Functions – Admissible and Consistent Heuristic Functions – Greedy Best-First Search and its properties – A* Search and its properties – Designing Heuristics

5 (Jun 6)	• Games – Formalizing 2-Player Zero-Sum Deterministic Sequential Games – Minimax Algorithm – Expectimax Algorithm – Formalizing 1-Player Stochastic Game
6 (Jun 13)	• Review: – Search Problems – Constraint Satisfaction Problems – Games
7 (Jun 20)	• *No classes*
8 (Jun 27)	• *Mid-term Test*
9 (Jul 4)	• Uncertainty: Modelling – Review of Probability – Uncertain Situations as Bayesian Networks – Identifying Independence Relationships in Bayesian Networks
10 (Jul 11)	• Uncertainty: Inference – Variable Elimination Algorithm – Sampling Methods

11 (Jul 18)	• Knowledge: Introduction – Representation versus Reasoning – Formal Languages and Reasoning Mechanisms • Knowledge: Representation – First-Order Logic
12 (Jul 25)	• Knowledge: Reasoning – Resolution by Refutation – Representing Knowledge in Clausal Form – Unification of Clauses – Factoring Clauses – Answer Extraction
14 (Aug 01)	• Review: – Uncertainty – Knowledge
14 (Aug 07)	• *Extra Lecture*
15 (Aug 15)	• *Project Submissions*