# UofT – CS Course Overview

## CSC110 (Foundations of CS I) // Mario Badr

Intro to Python programming, storing data (lists, sets, dictionaries, etc.), control flow (for/while loops, if loops), designing functions, mutation, memory model, classes, intro to simple data structures (priority queue, stack). Formal logic, simple proofs, GCD, modular arithmetic, cryptography (RSA), asymptotics ($\mathcal{O}(n)$), basic running-time analysis. Final project based on learning and using Python libraries of choice.

## CSC111 (Foundations of CS II) // Mario Badr

Intro to more complex data structures (linked list, trees, graphs), recursion & induction, sorting algorithms (insertion, selection, merge, quick), intro to average-case running-time analysis, some object-oriented design.

## CSC236 (Intro to the Theory of Computation) // Francois Pitt

Induction & strong induction & well-ordering principle. Proving partial & total correctness of algorithms with loop invariants, recursion, etc. Recurrence relations and generalized running-time analysis. Proofs with regular expressions, recursively-defined structures, finite-state machine, and regular languages.

## CSC263 (Data Structures & Analysis) // Francois Pitt

Priority queues & heaps, dictionaries & balanced search trees & hashing, ordered sets, graph algorithms (DFS, BFS, strongly-connected components, minimum spanning tree), disjoint sets – implementations and running-times. Average-case running-time analysis & quicksort, amortized analysis (aggregating and accounting methods) & dynamic arrays, proving lower-bounds on running-times for certain problem types.

## CSC207 (Software Design) // Lindsey Shorser

Intro to object-oriented programming: classes, methods and fields, abstraction, inheritance, encapsulation, polymorphism. Design principles, clean architecture, code smells, accessibility, design patterns, version control and GIT, visualizing classes (UML diagrams, CRC cards). Intro to Java – strong typing, classes and interfaces. Also teaches Java regex for some reason. Very big group app design project.

## CSC258 (Computer Organization) // Mario Badr

MOSFET transistors, logic gates, bits & boolean algebra, negative numbers, overflow, floating values, Karnaugh maps, combinatorial circuits (multiplexers, unsigned/signed arithmetic, comparators), sequential circuits (clocks, flip flops, registers, finite state machines, counters, shifters), designing circuits, MIPS memory model, writing in MIPS assembly language and converting to machine code, single-cycle processors, memory hierarchy and cache.

## CSC209 (Software Tools & System Programming) // Michelle Craig

Intro to Linux and basic shell programming. Intro to C – pointers, arrays, strings, memory model and dynamic memory allocation (on the heap), structs, header files & makefiles & compiling, definitions and macros, streams and I/O, signals and processes, parallelism with forking, pipes and sockets. Involves using the teach.cs labs.

## CSC369 (Operating Systems) // Mario Badr

asda

## CSC336 (Numerical Methods) // Christina Christara

Floating-point numbers – overflow, precision, error bounds. Catastrophic cancellation, well/ill-conditioned functions/matrices, numerical algorithm stability, types of error, $p$-norms. Some running-time analysis in terms of flops and divisions, spatial analysis. Gaussian elimination algorithm, special cases (symmetric, banded, inverses), LU-decomposition, Choleski factorization, pivoting and scaling. Nonlinear solvers (bisection, Newton-Raphson, secant). Interpolation (polynomial, Newton, Lagrange, splines). Some linear algebra and math proofs. Some coding in MatLab.

## CSC320 (Intro to Visual Computing) // Kyros Kutulakos

asda

## CSC343 (Intro to Databases) // Daniel Heap

Relational algebra (selection, projection, union/intersection, cartesian products & natural join), keys & superkeys & foreign keys, constraints and functional dependencies, closure, minimal basis. Database design, Chase test, Boyce-Codd normal form (BCNF), 3rd Normal Form (3NF), entity-relation models. Database transactions – concurrency and isolation levels. Intro to PostgreSQL – queries, adding schemas/tables & data, enforcing constraints, embedded SQL.

## CSC311 (Intro to Machine Learning) // Rahul Krishnan

Supervised learning: nearest neighbours and $p$-norms, information theory & decision trees, linear regression & classification, neural networks, backpropagation, convolutional neural nets, likelihood functions and naïve bayes, Gaussian discriminant analysis. Unsupervised learning: PCA and linear algebra, alternating least squares, autoencoders, clustering (k-means, E-M algorithm), bias-variance decomposition, bagging/bootstrapping, regularization, training & validation & testing, regression vs. classification, generative vs. discriminative, decision boundaries, overfitting & underfitting, feature maps. Reinforcement learning: Q-learning. Implementations in PyTorch.

## CSC373 (Algorithm Design, Analysis, & Complexity) // Nisarg Shah

General divide & conquer algorithms (quicksort, quickselect, binary search). Greedy algorithms, dynamic programming, directed graphs and network flows, linear programming (primal, dual, converting to standard form, integer LP), complexity (P, NP, NP-complete, coNP) and proving reductions, approximation algorithms. Maybe randomized algorithms. Lots of examples, lots of designing algorithms, proofs of correctness, proofs of running-time.

## CSC485 (Computational Linguistics) // Gerald Penn

Grammars, syntax and semantics (voice, thematic roles, agreement, etc.), ambiguity, sentence structure, parts of speech (POS) tagging (HMM, inside-outside & statistical parsing), syntax trees, projectivity, parsing (top-down, bottom-up, graph-based, chart-based), word-sense disambiguation (LESK), vector-based semantics (word2vec, wordnet, RNNs, LSTMs, BERT), language models (GPT), supertagging, statistical and unsupervised parsing. Involves PyTorch & neural networks + TRALE (based on ALE, based on prolog), which is only accessible on teach.cs servers. Take CSC324, CSC311, CSC209 before this course if possible.

## CSC401 (Natural Language Processing) // Raeid Saqur

Asda

## CSC413 (Neural Networks & Deep Learning) // Amir-massoud Farahmand

asda

## CSC412 (Probabilistic Learning & Reasoning) // Murat Erdogdu

asda