

# Louis Hildebrand

[louis.hildebrand@mail.mcgill.ca](mailto:louis.hildebrand@mail.mcgill.ca)   
[github.com/louis-hildebrand](https://github.com/louis-hildebrand)   
[linkedin.com/in/louis-hildebrand](https://linkedin.com/in/louis-hildebrand) 

## Formation

---

**Université McGill** Maîtrise en génie électrique

*hiver 2024–hiver 2026*

- **GPA:** 4.0/4.0
- **Professeur:** Prof. Christophe Dubach (compilateurs et synthèse de haut niveau)
- **Mémoire:** “A Minimal Intermediate Language for Generating Streaming Accelerators”

**Université McGill** Baccalauréat en génie logiciel

*automne 2020–automne 2023*

- **GPA:** 4.0/4.0
- Liste du doyen: 2020/2021, 2021/2022, 2022/2023
- British Association Medal (notes les plus élevées aux examens finaux)

**Collège John Abbott** Science

*automne 2018–hiver 2020*

- Major de promotion
- Liste du doyen: automne 2018, hiver 2019, automne 2019

## Compétences

---

- **Langages formels:** C, langage assembleur (ARMv7, MIPS), Rust, Python, Scala, Java, C#, SQL (MS SQL Server, PostgreSQL), VHDL, OCaml, JavaScript, HTML, CSS
- **Langages naturels:** anglais, français, afrikaans
- **Frameworks:** Spring Boot, .NET (Framework, Core), Django, Vue.js
- **Autres outils:** Git, Bash, Valgrind, Gradle, JUnit, LATEX, etc.

## Auxiliaire d'enseignement

---

**Computer Organization** (ECSE 324)

*automne 2025*

- Animer des ateliers portant sur l'architecture d'ordinateurs (ex. l'accès aux périphériques mappés en mémoire)
- Encadrer les étudiants dans l'écriture de programmes en C ou langage assembleur ARM
- Répondre aux questions des étudiants dans le forum de discussion
- Corriger les travaux

**Model-Based Programming** (ECSE 223)

*hiver 2025*

- Animer des ateliers portant sur le « model-based design » (ex. les diagrammes UML de classes ou d'états-transitions) et d'autres outils (ex. Git, JUnit, Gradle, Cucumber)
- Répondre aux questions des étudiants dans le forum de discussion
- Participer à la rédaction de matériels pédagogiques (ex. résumés des ateliers, consignes pour projets)

**Intro. to Software Engineering** (ECSE 321)*automne 2022–hiver 2025*

- Animer des ateliers portant sur le développement d'une application web avec PostgreSQL, Spring Boot, et Vue.js
- Répondre aux questions des étudiants, aussi bien dans le forum de discussion que durant les « office hours »
- Participer à la rédaction de matériels pédagogiques (ex. examens, résumés des ateliers)
- Corriger les examens

**Ordinary Differential Equations for Engineers** (MATH 263)*automne 2021*

- Animer des ateliers portant sur les équations différentielles ordinaires
- Répondre aux questions des étudiants par courriel
- Corriger des travaux

## Expériences de travail

---

**MDA Space** Stagiare DevOps*été 2023*

- Développer de nouvelles fonctionnalités et résoudre des bugs dans des applications web en utilisant ASP.NET MVC, Razor Pages, Telerik, et Kendo UI
- Optimiser des requêtes et procédures stockées SQL

**123Loadboard** Stagiaire en développement back-end*étés 2021, 2022*

- Développer de nouveaux microservices C# et .NET Core
- Résoudre des bugs et ajouter de nouveaux endpoints à l'API existante

**Camps de Jour Pierrefonds** Moniteur*étés 2017–2019*

- Animer des activités tout en assurant la sécurité des enfants (5 à 12 ans)
- Travailler aussi bien en anglais qu'en français

## Projets

---

**Sirop** (application Scala; projet de maîtrise)

- Langage de programmation et compilateur
- Traduit du code source de haut niveau en VHDL

**Twisty Timer** (application Java pour Android)

- Ajouter de nouvelles fonctionnalités à une application existante de « Rubik's Cube » (ex. pour pratiquer la résolution à l'aveugle)

**SH Prediction** (application Python)

- Prédit les rôles des joueurs dans le jeu de société « Secret Hitler »

**Pocket Cube Solver** (projet Arduino)

- Robot qui peut résoudre un « Rubik's Cube »  $2 \times 2 \times 2$
- Présenté à l'expo-sciences de Montréal 2018
- **Prix:** Intel Excellence in Computer Science Award, McGill University School of Computer Science (Robotics) Award

# Cours

---

<b>Language-Based Security</b> (COMP 523)	<i>hiver 2025</i>
<ul style="list-style-type: none"><li>• Formally studied the syntax and semantics of programming languages and type systems</li><li>• <b>Project:</b> <code>chick</code>, a type checker for a dependently-typed language</li></ul>	
<b>Computer Graphics</b> (ECSE 532)	<i>automne 2024</i>
<ul style="list-style-type: none"><li>• Learned the fundamentals of computer graphics: 3D transformations, meshes, the graphics pipeline, lighting, textures, etc.</li><li>• <b>Project:</b> a raytracer with support for surfaces of different colour and smoothness, mirrors, depth of field blur, spherical environment maps, textures, etc. Implemented in Python using the taichi library for GPU acceleration.</li></ul>	
<b>Machine Learning for Engineers</b> (ECSE 551)	<i>automne 2024</i>
<ul style="list-style-type: none"><li>• Studied classical machine learning models (decision trees, naïve Bayes, etc.) as well as neural networks, CNNs, and RNNs</li><li>• <b>Project:</b> stacked classifier (with random forest, logistic regression, etc. as the base models) to categorize Reddit posts from four cities. Achieved the second-highest accuracy on the test dataset (out of 25 groups).</li></ul>	
<b>Compiler Design</b> (COMP 520)	<i>hiver 2024</i>
<ul style="list-style-type: none"><li>• Learned to implement a compiler, including parsing, semantic analysis, register allocation, and code generation</li><li>• <b>Project:</b> compiler targeting MIPS assembly from a subset of C</li></ul>	
<b>Microprocessors</b> (ECSE 444)	<i>automne 2023</i>
<ul style="list-style-type: none"><li>• Programmed an STM32 B-L4S5I-IOT01A board using C and ARMv7 assembly</li><li>• <b>Project:</b> memory game that plays a series of tones (high or low), detects user inputs via accelerometer (up or down), and provides feedback via a speaker</li></ul>	
<b>Parallel Computing</b> (ECSE 420)	<i>automne 2023</i>
<ul style="list-style-type: none"><li>• Learned GPU programming with CUDA</li><li>• <b>Project:</b> CUDA implementation of a general 2D cellular automaton simulator, achieving <math>590\times</math> higher throughput than an equivalent sequential implementation in C</li></ul>	
<b>Operating Systems</b> (ECSE 427)	<i>automne 2022</i>
<ul style="list-style-type: none"><li>• Learned fundamental OS concepts: processes, threads, memory management, etc.</li><li>• <b>Assignments:</b> a simple shell, threading library, and file system (all in C)</li></ul>	