

# Yousif Zito

Brampton, ON

(647) 385-7950 • YousifZito@gmail.com • [GitHub](#) • [LinkedIn](#) • [Portfolio](#)

## Education

### McMaster University | Hamilton, ON

*Expected May 2025*

Bachelor of Technology in Software Engineering Technology

Relevant Coursework: Data Structures & Algorithms, Operating Systems, Object-Oriented Programming

### Seneca College | Toronto, ON

*Sep 2019 – Aug 2022*

Advanced Diploma in Computer Engineering Technology with Honours

Cumulative GPA: 3.60/4.00 | President's Honour List Fall 2020, Summer 2021, Summer 2022

## Skills

**Programming:** Python, C/C++, C#, .NET, SQL

**Web:** HTML, CSS

**Tech:** GitHub, Git, SVN, Unix, Visual Studio, VSCode, PyCharm, MS SQL Server, MS Office Suite

**Platforms:** Windows, Mac, Linux (Ubuntu, Debian), Red Hat (Fedora), OpenVMS

**Hardware:** Raspberry Pi, ARM Mbed Microcontroller

**Communication:** Design proposals, technical reports, instruction manuals, presentations.

**Languages:** English, Chaldean, Arabic

## Projects

### Online Shop Using C#

*Fall 2021*

#### *Seneca College, Advanced Programming Concepts Using C# - [GitHub](#)*

This project was part of the coursework, and the goals of the project were to develop an online shopping solution that could handle multiple users and multiple requests at once using TCP socket from C# .NET framework while following single responsibility principle of the OOP.

- Developed thread-safe console-based server for the online shop application with capability of validating the user's credentials and handling exceptions.
- Implemented multiple protocol standards to handle user connections, authentication, products, current orders, incoming purchases, and sending appropriate messages to the client-side based on the requests.
- Designed fully functional thread-safe GUI for the client-side using Windows Forms. The implementation of the client-side GUI consisted of login form and shopping interface.
- Some of the concepts used in this project were concurrent collections, multithreading, interfaces, classes, and SRP principle.

### Lottery Checker Using Python CGI

*Summer 2021*

#### *Seneca College, Programming Python with the Raspberry Pi*

*This project was individual based which emulates the OLG lottery scanner for 6/49.*

- Created an HTML form (Client) to allow users to select a background color set of 6 numbers from Canada's Lotto 6/49 lottery for the user to input lottery numbers.
- Implemented Python CGI server, that retrieved the lottery results for 6/49 lotto from the internet to determine if the selected user numbers have won a prize.
- Then deployed the project on a [server](#)

### Conway's Game of Life

*Winter 2021*

#### *Seneca College, Event Driven Programming Using C# - [GitHub](#)*

For this project, I implemented my own Game of Life simulator in C# with a fully functional GUI based on Windows Forms from the .NET framework. Some of the features that distinguishes this Life simulator from other versions on the internet:

- Implemented functionality that allows the user to specify the color of each alive cell at the beginning of the simulation or when it is paused. The allowable colors are red, green, and blue.
- Implemented game engine that determines the color of the cell by averaging the red, green, and blue contributions of the neighboring "parent" cells. This allowed the simulator to explore an additional life dimension involving inheritance and leads to a dynamic simulation in which new cells are born with different colors than their individual parent cells.
- Implemented the capability to save the current state of the 2D grid as a 10-times scaled PPM color image.