Week 1 Day 1 Notes:

• Welcome

* Introduction
  + Connect our code to a database
  + Mr. Mathew Desjardins
    - Lecturer in the Department of Computing
    - B.S. and M.S. in Computer and Information Science
    - Ph.D. Information Systems, Analytics and Decision Support (In-progress)
* Why Study Web Programming
  + Full stack programming
    - Someone who can develop both client and server software.

(backend programming)

* + What you need to learn
    - Backend framework (.NET core)
    - Bootstrap (Pre-programmed CSS)
    - API development
* Why use C#
* Why .Net Core
  + Used in many places
  + C style language
  + Learning to stack
    - Collection of tools
  + Build/make
    - Websites
    - Full scale Software Application
    - Video games
    - Mobile development
    - AR
    - Front end client
    - API
    - Build anything

• Syllabus Review

* + Major grading items
    - Daily notes quiz
      * Will be assigned every class day except on lab days
      * Is used as a studying tool for the midterm and final
      * Due 15 minutes after class
  + By the end of this course
    - Write applications in one sever-side scripting language
    - Execute basic SQL
    - Work with modern development tools
    - Develop real-world server-side applications that interact with a database
  + Use Microsoft C# documentation (your best friend)
    - Gives an example of anything in C# and gives code examples
  + Zoom
    - Cameras on
  + Projects/Labs
    - Submit to GitHub
    - Submit when needing help on projects
  + Late Work
    - No late work is taken (Reasons are limited)
  + Grades
    - Quizzes – 20%
    - Midterm – 20%
    - Final Exam – 20%
    - Labs/Projects – 40%
    - DOES NOT ROUND UP
  + Email subject (CSCI 2120)
  + Do not cheat. Recite code if taken off the internet

• Understanding GitHub and GitHub Classroom

* Git
  + Created by Linus Torvalds in April 2005
  + 90% of companies use Git for version control
* Git DVC
  + No central server needed
  + Can work without internet connection
  + There is a difference between Git and GitHub
  + Developers can put any type of files in the Git and combine their work later
  + Every copy of Git repository can serve either as the server or as a client
    - Repository
      * Repos are used to organize a single project
      * Contains folders, files, images, videos, spreadsheets, and other data constructs
  + Branching
    - * When working on a project, there will be many ideas given. some good some not. Branching helps manage workflow
      * Ideas do not work? delete the branch
    - Main (master)
      * Anything in the main branch is always deployable
      * this branch has the latest working commit
    - Commits
      * Once main is created its time to start making changes
      * Break problems into smaller parts
      * Write clear concise committee messages
    - Pull Request
      * Go and pull my code
      * Start code review and conversations about proposed changes before merged into main
    - Merge and Deploy
      * After pull request is reviewed and branch passes the tests, you deploy changes to verify in production
  + GitHub
    - Host’s git code repos
    - Most popular Git host
    - Allows user to work on projects from anywhere