Dwain Anderson

516-591-7908 | dwainanderson05@gmail.com | linkedin.com/in/dwain | github.com/dwain

Education

Cornell University, College of Arts & Sciences Ithaca, New York
Bachelor of Arts in Computer Science *Aug. 2023 – May 2027*

Coursework: Functional Programming, Object Oriented Programming & Data Structures, Machine Learning Database Systems, Visual Data Analytics, Intermediate Web Programming, iOS Development, Backend Development, Probability Theory, Linear Algebra, Discrete Structures, Statistics, Calculus

Technical Skills

Languages: Python, Java, PHP, SQL, JavaScript, Swift, HTML5/CSS3
Frameworks/Library: React, SQLite3, jQuery,
Developer Tools: Postman, VSCode, IntelliJ, PyCharm, Pulsar, XCode, GitHub, Git

Projects

Spreadsheet Formula Evaluation | *Java, JUnit*

- Developed Java batch application to evaluate postfix notation formulas in CSV spreadsheets.
- Implemented an interactive calculator and CSV formula evaluator using subtype polymorphism.
- Utilized tree data structures for expression evaluation.
- Used lambda expressions for JUnit testing and exception handling.

Cornell AppDev Hackathon: BookKeeper | *Python, SQL, Flask*

- Designed a CRUD Flask app for saving and favoriting books.
- Designed the database using SQL and implemented user login/logout, password hashing, and sessions.
- Implemented a K-Nearest Neighbors book recommendation algorithm.
- Used JUnit test for both glass-box and black-box testing as well as end-to-end unit testing.

Merging Spreadsheets Evaluation | *Java, JUnit*

- Developed a Java batch application to perform a Left Inner Join on CSV files.
- Implemented the linked-list data structure and utilized its associated operations for efficiency.
- Utilized defensive programming for maintaining data integrity.
- Used end-to-end testing with JUnit.
- Implemented a binary search tree to store variables and query their values efficiently.

Connect-N Game | *Python, PyGame*

- Created a Connect-N game in Python with a graphic user interface.
- Implemented using object-oriented design principles and the model view controller design pattern
- Implemented AI player logic using iterative backtracking algorithms and memoization approaches.
- Utilized error handling defensive program to prevent bugs from occuring.