

Success in CS-460: Programming Languages

Robert Bruce

What is CS-460 Programming Languages all about?

In CS-460, we will be:

- ~ Defining and utilizing programming language grammar.
- ~ Parsing through tokenization and later, recursive descent to form syntax trees.
- ~ Utilizing lexical analysis for semantics (context) and syntax (conformity to grammatical rules).

On a more subtle level, we will be:

- ~ Examining differences in programming language design and features.

How will we implement this?

- You will write programs in C or C++ which implement:
 - 1) Procedural-based Deterministic Finite-state Automata (DFA).
 - 2) Binary trees using Left-child, Right-sibling (LCRS) to store parse trees.
 - 3) Linear linked lists to store symbol tables.
- The culminating final project in this class is developing an interpreter for the ChagaLite programming language.
- *Note: ChagaLite is a simplified version of the Chaga programming language. All ChagaLite documentation required for CS-460 is posted on Canvas.*

The Chaga Programming Language

- Chaga is an imperative programming language I created for numerical computation.
- I use the Chaga programming language to compute large numbers such as:
 - ~ [The ten-millionth term of the Fibonacci series](#)
 - ~ [The first thousand terms of the Fibonacci series](#)
 - ~ [Factorial tables up to one-hundred-thousand factorial](#)
- I wrote and maintain an open-source compiler as well as extensive documentation for the Chaga programming language at:
 - ~ https://www.robertjamesbruce.com/chaga_programming_language.html

Is this class really difficult?

- Yes, this class is difficult.
- I designed the in-class exercises to prepare you for success and overcome the difficulties.
- *Hint: a solid foundation built on tokenization will make lexical analysis and subsequent steps much easier when writing an interpreter.*

Emphasis on short lectures

- Lectures will generally be short (approximately 30 minutes).
- Class-time primarily focused on in-class exercises working with other students.
- I've designed the in-class exercises to help you understand and achieve course goals and objectives.
- *Note: the assigned readings in our textbook are not optional. Our textbook complements material I don't cover in class. I expect you to read the assigned readings each week and ask questions during or after class (or during student hours aka my office hours).*

Success in CS-460: Programming Languages

- Attend classes and contribute to the in-class exercises.
- Manage your time well: pace yourself!
- Utilize my weekly student hours (traditionally called “office hours”).
- Communicate! Communicate! Communicate!

Communicate! Communicate! Communicate!

- When issues arrive on a team and cannot be resolved, please let me know as-soon-as-possible.
- I have diplomatic ways to find common ground and get the team to work well together.
- I want your team to succeed!
- I have student hours (traditionally called “office hours”) to serve you! Student hours are a GREAT time to talk. I really enjoy hearing from students.

Individual versus Team work

- This class combines team work (25% of grade) and individual achievement (75% of grade).
- Everyone is responsible for developing their own implementation for the first four programming assignments:
 - ~ Programming assignment 1: Ignore comments
 - ~ Programming assignment 2: Tokenization
 - ~ Programming assignment 3: Recursive Descent Parser
 - ~ Programming assignment 4: Symbol Table
- In the last two assignments, students will work in a team of four:
 - ~ Team Programming assignment 5: Abstract Syntax Tree
 - ~ Team Programming assignment 6: Interpreter

The Takeaway

- I'm here to help you and guide you.
- We work as a team.
- Your commitment to this class is paramount to your success.
- Make me proud of your work!