Course Code	Course Title
COMP 3031	**Principles of Programming Languages (Fall, 23-24)**

Course Description

In this course, you will learn to design programs that are reliable, maintainable, efficient, and correct by construction. We achieve this through high-level programming techniques that focus on simplicity and expressiveness, notably using declarative and higher-order programming with algebraic data types. We also review important programming language paradigms and constructs, including imperative, object-oriented, logic, and concurrent programming techniques, with an overall emphasis on functional programming and the Scala programming language. The goal is to expand your mind and give you new tools enabling you to solve hard problems in easier and more powerful ways.

List of Topics

1. Functional programming
2. Recursion and tail-recursion
3. Evaluation strategies, lazy evaluation, substitution model
4. Modularity, data abstraction, representation independence
5. Subtyping, inheritance, type classes
6. Polymorphism, variance
7. Structural induction
8. Stateless parallelism, map-reduce, associative operations
9. Effects: state, exceptions
10. Interpreters and program semantics

Textbooks

N/A

Reference books

- Martin Odersky, Lex Spoon, Bill Venners. Programming in Scala, 5th edition. Artima, 2021.

Grading Scheme

Three programming assignments (10% each)	30%
Midterm exam	30%
Final exam	40%
Total	100%

Course Intended Learning Outcomes

1. Identify the general constructs and concepts used in implementing programming languages, particularly those in type systems and functional programming

2. Differentiate the alternative programming paradigms of functional, logic, and dependent programming.
3. Learn about declarative and algebraic ways of designing programs.

Assessment Rubrics
N/A