CS 345
Programming Languages
Final Review
Overview – What we have done

- The content described in the midterm review plus:
  
- Towards Functional Programming Languages
  
  - Unit Type, Derived Form, Sequencing, Ascription, Let Binding, Pairs, Tuples, Records, Sums, Uniqueness of Types, Variants, Option, Enumeration, Recursion, ML (fn, val, fun, basic syntax)
  
- Type Inference, Parametric Polymorphism
  
  - Type Inference, Constraint Generation, Unification, Free Type Variables, Most General Solution, Parametric Polymorphism, Let-Polymorphism, Type Schemas, Monotypes and Polytypes, Hindley-Milner
Overview

- **Scala**
  - Data Types and Operators, Variables, Control Flow, Functions, Recursion, Classes, Constructors, Functions / Methods, Objects, Traits, Case Classes, Pattern Matching, Generics, Type System, Exception Handling, Implicit Conversion, Internal and External DSLs

- **Subtyping, Subtype Polymorphism**
  - Subsumption, Subtype Relation, Nested Records, Subtyping for Functions, Contravariance, Covariance, Top Type, Lambda Calculus with Subtyping, Ascription and Casting, Coercion, Coherence, Algorithmic Subtyping, Algorithmic Typing, Joins and Meets
Overview

- **Object-Oriented Programming**
  - Objects, Encapsulation, Subtyping, Behavioral Subtyping, Liskov Substitution Principle, Inheritance, Subtyping vs. Inheritance, Dynamic Dispatch, Single Dispatch, VTables, Multiple Dispatch

- **Multiple Inheritance and Alternatives**
  - Subclassing, Inheritance without Subclassing, Multiple Inheritance, Ambiguity, Repeated Inheritance, The Diamond Problem, Virtual Inheritance in C++, Interfaces, Traits, Mixins, Mixin Traits in Scala
Overview

- Overview of Functional Programming
  - Pure vs. Impure, Mutability vs. Immutability of Data, Lazy Evaluation, Implicit vs. Explicit Typing, Static vs. Dynamic Typing, Abstract Data Type, Recursive Types, Existential Types, Ad-Hoc Polymorphism, Type Classes, Functors, Monads, Uniqueness Types, Closures, Continuations

- Trends in Modern Programming Languages