1. Untyped lambda calculus
   (a) Syntax
   (b) Evaluation: Beta, eta-reduction
   (c) Examples, church-encoding numbers, booleans, Y-combinator/loops
2. Simply typed lambda calculus
   (a) The type system
   (b) Examples: type checking some small programs
3. Curry-howard correspondence
   (a) Programs are proofs
   (b) Example: encoding modus-ponens and its proof
   (c) Limitations of simply typed calculus as proof language
4. Polymorphism
   (a) Forall types, let-polymorphism(ML), and first-class polymorphism
   (b) Type-checking polymorphism: the type rules
   (c) Example type-checking of a polymorphic function and application
5. Back to Curry-Howard isomorphism with polymorphic types
   (a) Polymorphic logic: "forall" theorems and proof terms - generalizing the modus ponens, why typechecking is proof-checking