# 7. Dolphin - Phase 4

## Task 1: Design and implement an AST and Typed AST for Phase 4

In ´Ast.ml´, we introduced support for function declarations and comma expressions by extending the AST with new nodes for ´function\_decl´ and ´CommaExpr´. The corresponding types were also added in ´typedAst.ml´ to ensure proper type handling for both functions and comma-separated expressions.

## Task 2: Extend your parser to support new language features

´parser.mly´ was updated to support the parsing of function declarations, including their return types, names, parameters, and bodies. Additionally, we added parsing rules for comma expressions, allowing expressions to be evaluated in sequence, which enhances the flexibility of code blocks and for-loops.

## Task 3: Extend your semantic analysis to support new language features

In ´semant.ml´, we implemented a two-pass system to collect function signatures first and then type-check function bodies, enabling support for mutual recursion. We also introduced type-checking rules for comma expressions to enforce their correct usage within the language, ensuring they are used only in appropriate contexts.

## Task 4: Design and implement code generation for Phase 4

In ´codegen.ml´, we added functionality to generate LLVM code for function declarations, including copying arguments to stack memory to allow modifications within the function body. We also implemented code generation for comma expressions, ensuring the left expression is evaluated and discarded while retaining the result of the right expression, in line with the intended semantics.