1. History of Haskell
   - Haskell was named after Haskell B. Curry, an American mathematician and logician.
   - This language is widely used in academia and the industry.
   - Haskell ranks at the 44th place on TIOBE Index for September 2020
   - Haskell ranks at 25th place on the Ranking Programming Languages by GitHub users chart.

2. Some Key Features of Haskell
   - Haskell is a purely functional language:
     - Remember: functional languages focus on the computation, not the “how to”!
     - Different between this and Java/C:
       - The program is a function!
       - Variables and Assignments aren’t useful as variables are assigned on the go!
   - Haskell is statically typed: Type determined at runtime instead of before runtime (Just like Java!)
     - Code compiled more quickly
     - Class errors are caught early.
   - Haskell uses type inference: The user does not have to explicitly state every type in a program. Instead, the type of variables or functions is automatically inferred.
   - Haskell uses lazy evaluation (also called call-by-need): Functions don’t evaluate their arguments. An expression will only be evaluated when its value is needed. This:
     - Reduces running time, memory usage becomes hard to predict
- Allows for processing infinite data and bypassing undefined values (e.g. results of infinite loops)

- Haskell is immutable and concurrent:
  - Immutability: Once a variable is assigned in Haskell, it’s immutable. This helps “reduce[ing] coupling between components, simplif[ing] concurrency and parallelism, and decreas[ing] the total number of moving pieces in a system, making it easier to maintain and develop over time.” (Snoyman, 2017)
  - Concurrency: Allow multiple threads to be run at the same time.

- Haskell has garbage collection, like Java: garbage collector collects unused objects, freeing up spaces.

Sources:


