

7. LISP began as a pure functional language but gradually acquired more and more imperative features. Why?

Maintaining the trait of a pure functional language became futile as developers added features that helped increase performance in certain tasks. In doing so LISP no longer remained as a pure functional language. This change of direction was due to LISP being a flexible and applicable language in various domains, which encouraged its developers to experiment with new features that will further extend its use.

12. Describe in your own words the concept of orthogonality in programming language design

A PL that follows the philosophy of orthogonality consists of small primitive instruction sets that are combined to generate more complex functions. Each primitive instructions has its unique functionality and the developed program is solely determined by the combination of these instructions which does not interfere with the others.

14. What are the arguments both for and against the idea of a typeless language

Advocates - A typeless language is more flexible as the user doesn't have to manually designate types nor convert them. This is a huge advantage when the user deals with complex data structures that involve various data types.

Disadvantage - Type flexibility means less readability thus the programmer can get more confused. The performance of a typeless language is worse because the constant type changing hinders performance.