2 language design principles:

1. Efficiency: efficiency is an important aspect for designing any language. It can be achieved in different ways:

- Ease of translation: the code should be easily translated by a reasonably sized compiler. Example: one-pass compilers

- Implementability: efficiency with which a translator can be written. This is usually dependent on the language's complexity. Example: Ada was hard to develop compilers for.

- Programming efficiency: ease of writing & development using the language. Example: python is very easy to use & learn

- Reliability & Maintainability: A language that is easy to read is easier to maintain. A language should also behave in the same way every time. Example: Python is an easy language to read and maintain.

2. Simplicity: A language must be not too simple that complex tasks require longer codes. It also must not be too complex & obscure to understand