Names, Scopes, and Bindings

High-Level Language (HLL): The language syntax and semantics are significantly more abstract—farther from the hardware.

Name: a name is a mnemonic character string used to represent something else. Names in most languages are identifiers (alphanumeric tokens). Names allow us to refer to **variables, constants, operations,** and **types** using symbolic identifiers rather than low-level concepts like addresses

Abstraction: a process by which the programmer associates a name with a potentially complicated program fragment, which can them be thought of in terms of its purpose or function, rather than in terms of how that function is achieved.

Subroutines are control abstractions: they allow the programmer to hide arbitrarily complicated code behind a simple interface.

Classes are data abstractions: they allow a programmer to hide data representation details behind a simple set of operations.

Referencing Environment: The complete set of bindings in effect at a given point in a program.

Binding time refers not only to the binding of a name to the thing it represents, but also in general to the notion of resolbing any design decision in a language implementation.