Binding to Names and Storage Addresses
Variables: The moment you declare a variable, the identifier is bound to a memory location where its value will be stored.
Pointers: When using pointers, the identifier points to the memory address of another variable.

Lifetime
The lifetime of an object ends when:

- if it is of a non-class type, the object is destroyed
- if it is of a class type, the destructor call starts
- the storage which the object occupies is released, or is reused by an object that is not nested within it.

Local Variables: These variables are created when a function is called and destroyed when it exits.
Static Variables: Have a lifetime for the entire duration of the program, but their scope can be limited to the block in which they're declared.
Dynamic Variables: Allocated using new and persist until explicitly deallocated with delete
https://en.cppreference.com/w/cpp/language/lifetime

Scope

1. Local Variables: Anything between "{" and "}" are local variables
2. Global Variables: Can be accessed throughout any part of the program
   https://www.geeksforgeeks.org/scope-of-variables-in-c/