



Unit I

CHAPTER 3

SCOPING



Learning Objectives

After the completion of this chapter, the student will be able to

- Understand what is Scoping
- Able to implement the LEGB rule
- Understand what is module
- Understand the implementation of access control in programming language

3.1 Introduction

Scope refers to the accessibility of a variable within one part of a program to another part of the same program.

3.2 Variable Scope

To understand the scope of variables in a programming language, it is important to learn about what variables really are. Essentially, they're addresses to an object in memory. When you assign a variable with `:=` to an instance (object), you're binding (or mapping) the variable to that instance. Multiple variables can be mapped to the same instance.



Note

The process of binding a variable name with an object is called mapping. `=` (equal to sign) is used in programming languages to map the variable and object.

Programming languages keep track of all these mappings with namespaces. ***Namespaces are containers for mapping names of variables to objects.*** You can think of them as dictionaries, containing a list of words and their meanings. The words are mapped with their meaning in dictionaries, whereas names are mapped with objects (name = object) in programming language. This allows access to objects by names you choose to assign to them.

In the following example, ***a*** is first mapped to the integer ***5***. In this case, ***a*** is the variable name, while the integer value ***5*** is the object.

Then, ***b*** is set equal to ***a***. This actually means that ***b*** is now bound to the same integer value as ***a***, which is ***5***.

1. `a:=5`
2. `b:=a`