

**Introduction to Lua Programming**

**Course Number:** LUA-100  
**Duration:** 2 days

**Overview**

[Lua](https://www.lua.org/) is a free, open-source programming language that is popular as an embedded plugin/extension language in different software products, In addition, Lua can also be used as a standalone language.

Lua is both powerful and easy to use with a reputation for being high performance, extremely portable, lightweight, and embeddable.

This Lua Programming training course teaches attendees how to write middle-sized standalone Lua programs and script software products where Lua is the embedded language. Students learn fundamental programming constructs in Lua, as well as concepts that are useful in other areas of software engineering.

**Headcount Cap:** Due to the intensive instructor interaction in this class, headcount is capped at 10 attendees per delivery.

**Prerequisites**

All students must have familiarity with at least one other programming language. Having familiarity with other scripting languages is helpful, but not required.

**Materials**

All Lua training students receive comprehensive courseware.

Courseware by NobleProg® Ltd, 2004 - 2021 All Rights Reserved.

**Software Needed on Each Student PC**

* Internet connection and a modern browser.

**Objectives**

* Program using Lua's basic syntax and semantics
* Work with data types and data structures
* Create tables
* Work with conditionals
* Incorporate error handling
* Work with functions
* Use modules and packages
* Understand the building blocks of OOP in Lua

**Outline**

* Introduction to Lua
  + A bit of history
  + Lua's goals, features and non-goals
  + Resources of Lua documentation and tutorials
  + Installing the Lua interpreter
  + Setting up and using LuaRocks
* Basic Syntax and Semantics
  + Identifiers
  + Comments, block comments
  + Global variables and enforcing of strictness
  + Local variables
  + Standalone programs, program arguments
  + Compilation units, chunks, expressions, semicolons
* Data Types and Data Structures
  + Basic types: nil, boolean, number, string
  + Object types: function, userdata, thread, table
  + References/objects vs. basic values
  + The importance of tables in Lua
* Introduction to Tables and Their Versatility
  + Tables as an associative array
  + Tables as numeric arrays, sequences
* Basic Control Structures
  + The if then elseif else end
  + The while loop
  + The repeat loop
  + The simple for loop
* Error Handling
  + Return values vs exceptions
  + Converting a return value to an exception
  + Converting an exception to a return value
  + Error levels
* Example Programs
  + Polynomial evaluation
  + Breadth first search
  + Additional exercises
* More about Functions
  + Named arguments
  + Object-oriented calls
  + Closures
  + Currying
  + Tail calls
  + Multiple assignment and return
  + Varargs
* Iterators and Co-Routines
  + The generic for loop
  + Stateless vs stateful iterators
  + Differences between iterators and co-routines
* Metatables and Metamethods
  + The set example
  + The \_\_tostring metamethod
  + Arithmetic metamethods
  + The \_\_index, \_\_newindex metamethods
  + The \_\_len metamethod
* Modules and Packages
  + Using modules
  + Creating modules
  + Organizing modules into packages
* Object-oriented Programming
  + The building blocks of OOP in Lua
  + Examples
* Discussion on more advanced language features
  + Weak tables
  + Finalizers
  + Compilation, eval, loading
* Conclusion