Wanjing Li

CS 330

September 24 2020

CS330: Programming Language Project (PLP) Assignment 3:

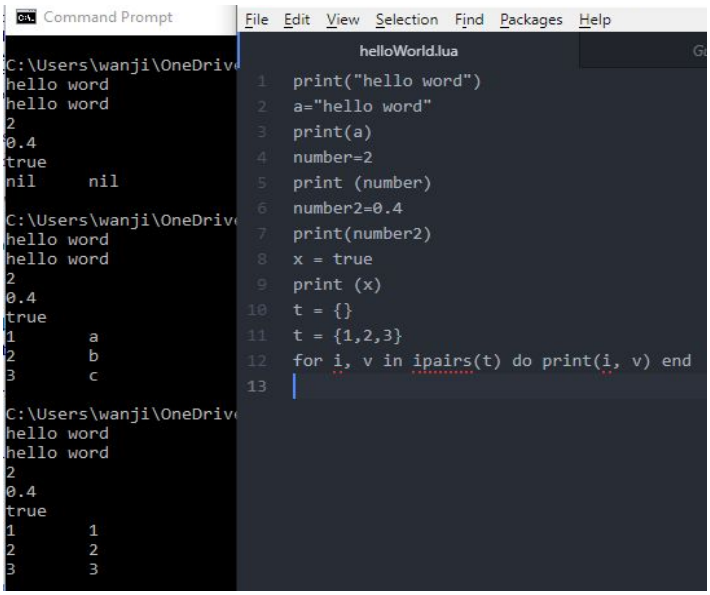
Data types and naming conventions

Understanding how different types of data are represented in your programming language is a key step in being able to use it for bigger projects. Research the naming conventions in your language for variables (i.e., do they have to start with lowercase letters? Can they start with numbers? Symbols? do programmers use underscores, as in "last_name", or do they use camel case (lastName))?

Variables can use uppercase letters but are typically lowercase. They can't start with numbers or symbols and require an underscore in between names, with the second word in uppercase. CamelCase is also used as a naming convention.

Write a piece of code that creates variable of each of these common data types and follows the naming conventions:

- **int**
- **string**
- **floating-point number**
- **boolean**
- **array/list /hash/dictionary(only have tables)**



Even if your programming language only has one data type, or if it doesn't require that types be declared, you should still be able to create variables that store these types of information (well, maybe not the hash table). If your language doesn't have variables or doesn't differentiate between data types, then find out how it stores information and do that.

In your code, experiment with doing different things with the data types: can you add ints and floats? If you do, is the resulting variable an int (narrowing conversion) or a float

(widening conversion)? Can you put different data types in the same array or list? Can one data type be converted to another (int to float, string to int, etc)?

There are no integer types so every number value is a float. Lua can put any data type in tables, and convert one data to another.

Discussion questions:

1. What are the naming requirements for variables in your language?

Lua is case sensitive so uppercase and lowercase variable names are distinct. Keywords also can't be used as variable names.

1. What about naming conventions? Are they enforced by the compiler/interpreter, or are they just standards in the community?

Lua's naming conventions are enforced by the compiler.

2. Is your language statically or dynamically typed?

It is dynamically typed, which checks for the error during compilation.[1]

3. Strongly typed or weakly typed?

Like python, Lua is also strongly typed, meaning that all type errors are detected.

4. If you put this line (or something similar) in a program and try to print x, what does it do? If it doesn't compile, why? Is there something you can do to make it compile?

x = "5" + 6

It printed out 11, which shows that Lua automatically converted the string 5 to a float.

5. Describe the limitations (or lack thereof) of your programming language as they relate to the coding portion of the assignment (adding ints and floats, storing different types in lists, etc). Are there other restrictions or pitfalls that the documentation mentions that you need to be aware of?

The only limitation that I have encountered was the lack of integer values.

6. How do type conversions work in your language? Are the conversions narrowing or widening, and do they work by default or do they have to be declared by the programmer?

Type conversions work automatically. Due to no int variables, the conversions are neither narrowing nor widening.

---

[1] http://lua-users.org/wiki/NumbersTutorial