CS330: Programming Language Project (PLP)

Assignment 6: Naming, scope, and bindings

You've already looked up the naming conventions for your variables, but what's the scope on your variables? Write code that tests the following questions, then write out your answers to the questions

1. Declare a variable (say, x) in the main body of your program. Then declare one inside of a for loop. Is there a conflict? Is the old variable overwritten, or do you now have two variables of the same name?

The old variable is overwritten.

x outside: 0

x: 2 at i loop: 0

x: 2 at i loop: 1

x: 2 at i loop: 2

x: 2 at i loop: 3

x: 2 at i loop: 4

x: 2 at i loop: 5

x after the loop: 2

2. What if the other x is inside of a function?

There is two x with different values:

x inside function: 3

2

Unless the variable is global in the function.

3. Can you have variables that are globally accessible? What are the rules for creating them?

y inside function: 5

Y print outside function: 5

Must be declared global before assigning any value. Must be inside of function. Must call for function first before accessing to the variable

4. Are some variables passed by value while others are passed by reference? Which ones are which?

Normally, it will be passed by value defaultly (x after the loop is different before the loop). But if the function is global, it will be by reference

5. If you run this code (or the equivalent) in your language, what is the output? What does that tell you about how the language handles assignments?

char [] a = {'c','a','t'}

char [] b = {'d','o','g'}

a=b

b[1] = 'u'

print a

print b

Item of A 0 d

Item of A 1 o

Item of A 2 g

Item of B 0 d

Item of B 1 u

Item of B 2 g

It is passed by value

As always, write out the answers to these questions as though you were writing a guide for a new learner. Make your answers as clear as possible, and don't just answer the questions in one or two words. Explain the reasoning behind the answers as much as possible. If there is no clear-cut answer to a question, explain why not. And cite your sources!

Turn in both your code and your written answers, and be prepared to run your code for me in lab.