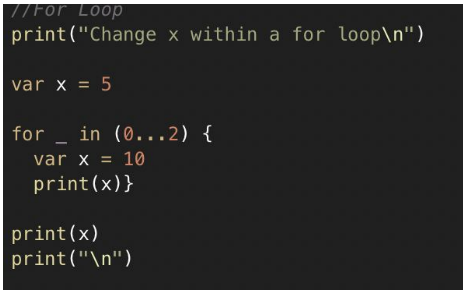
CS330: Programming Language Project (PLP)

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?

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?**

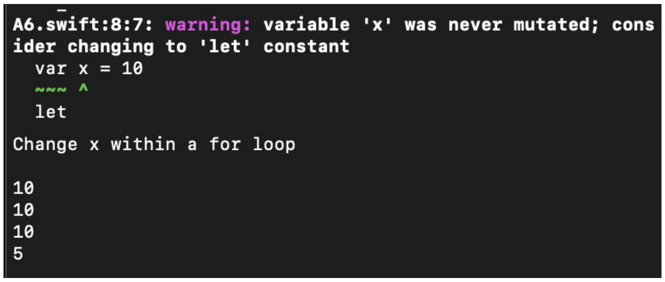


```
//For Loop
print("Change x within a for loop\n")

var x = 5

for _ in (0...2) {
  var x = 10
  print(x)}

print(x)
print("\n")
```

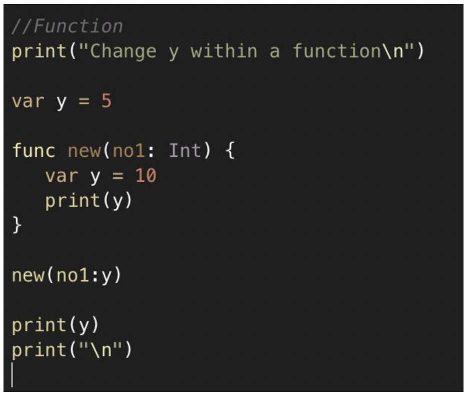


```
A6.swift:8:7: warning: variable 'x' was never mutated; cons
ider changing to 'let' constant
  var x = 10
  ~~~ ^
  let
Change x within a for loop

10
10
10
5
```

As you can see in the code above, there wasn’t a conflict. The original value of x remained unchanged, and an error popped up to show that it was never mutated, but within the for loop, print(x) printed out x as 10. This means that the code now has two variables of the same name.

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



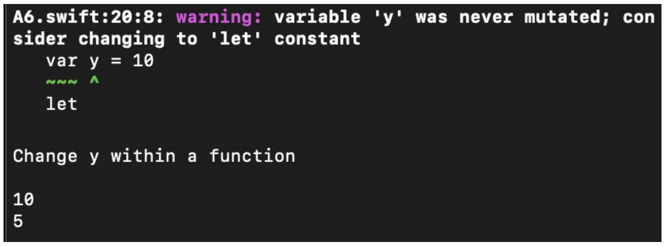
```
//Function
print("Change y within a function\n")

var y = 5

func new(no1: Int) {
   var y = 10
   print(y)
}

new(no1:y)

print(y)
print("\n")
```



```
A6.swift:20:8: warning: variable 'y' was never mutated; con
sider changing to 'let' constant
   var y = 10
   ~~~ ^
   let

Change y within a function

10
5
```

Since the function did not return anything, y retained its original value when printed outside of the function.

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

   Global variables are variables that are defined outside of any function, method, closure, or type context.[1] There are two ways to declare global variables:

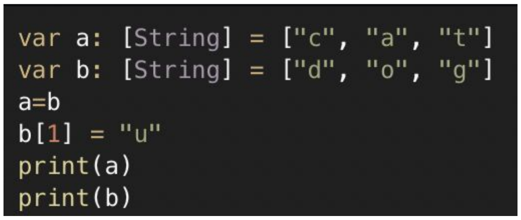
   1. Declare the variable within the class but outside any functions so that any functions within the class can access it.
   2. Declare all the global variables using a struct:

      ```
      struct Student {
         static let name: String="Aman"
         static let age: Int = 22
      }
      class Employee {
         func getData() {
            print(Student.age)
            print(Student.name)
         }}
      ```
      [2]

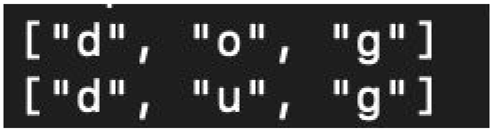
4. **Are some variables passed by value while others are passed by reference? Which ones are which?**

   Class instances and functions are passed by reference but arrays and strings are passed by value.

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**



```
var a: [String] = ["c", "a", "t"]
var b: [String] = ["d", "o", "g"]
a=b
b[1] = "u"
print(a)
print(b)
```



```
["d", "o", "g"]
["d", "u", "g"]
```

   It shows us that arrays are passed by value because we changed ‘b’, but ‘a’ remained the same. a=b didn’t assign the location/address of b’s contents to a, it assigned the actual value.

[1] https://www.tutorialspoint.com/how-to-create-and-use-global-variable-in-swift

[2] https://www.tutorialspoint.com/how-to-create-and-use-global-variable-in-swift