### 12.1 Variable name scope

#### **Scope of names**

A declared name is only valid within a region of code known as the name's **scope**. Ex: A variable userNum declared in main() is only valid within main(), from the declaration to main()'s end.

Most of this material declares variables at the top of main() (and if the reader has studied functions, at the top of other functions). However, a variable may be declared within other blocks too. A **block** is a brace-enclosed {...} sequence of statements, such as found with an if-else, for loop, or while loop. A variable name's scope extends from the declaration to the closing brace }.

PARTICIPATION ACTIVITY	12.1.1: Variable name scope extend to the end of the declaration's block.	
Animation	captions:	
declarat 2. A declar the decl	m's scope is from the declaration to main's closing brace. Using userNum before the tion would yield an "Undeclared name" compiler error. ration can appear within any block (statements in braces {}). valSquared is valid fror laration to the closing brace. ared is not valid outside that block.	n
PARTICIPATION	12.1.2: Variable name scope.	$\overline{}$

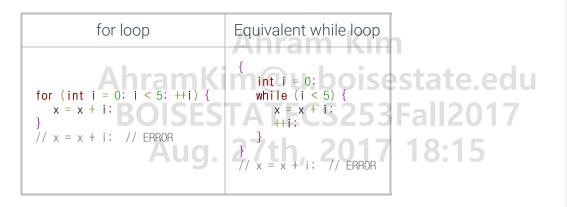
PARTICIPATION ACTIVITY	12.1.2: Variable name scope.		
Refer to the animation above.			
userNum c declaration			
O True	AhramKim@u.boisestate.ed	u	
O False	BOISESTATECS253Fall2017		
2) userNum c	can be used in val1's <b>Aug. 27th, 2017 18:15</b>		
O True			
O False	e		
,	can be used within the for k of statements.		

O True	
O False	
4) valSquared can be used within the for loop's block.	
O True	
O False Ahram Kim	
5) valSquared can be used in the for loop's state equal loop variable update, such as replacing ++i by i = i + valSquared.	
O True g. 27th, 2017 18:15	
O False	
6) valSquared can be used just before main's return statement.	
O True	
O False	

#### For loop index

Programmers commonly declare a for loop's index variable in the for loop's initialization statement. That index variable's scope covers the other parts of the for loop, up to the for loop's closing brace. The reason is clear from the for loop's equivalent while loop code shown below, noting the braces around the equivalent code.

Table 12.1.1: Index variable declared in a for loop's initialization statement.



The approach of declaring a for loop's index variable in the for loop's initialization statement makes clear that the variable's sole purpose is to serve as that loop's index.

12.1.3: For loop index declared in loop's initialization statement.	
Given the following for loop, use the above equivalent while-loop to determine whether index i's scope includes the indicated region.	
(a) for (int i = 0; (b); (c) ) {    (d)	
(e) Ahram Kim	
1) (a) hramKim@u.boisestate.edu	
OYeSESTATECS253Fall2017	
O Rug. 27th, 2017 18:15	
2) (b)	
O Yes	
O No	
3) (c)	
O Yes	
O No	
4) (d)	
O Yes O No	
5) (e) O Yes	
O No	
6) Suppose the above for loop is followed by a second for loop also with int i = 0 in the initialization statement. Will the ramkin@u.boisestate.ed compiler generate an error due to two declarations of i?  Ahram Kim @u.boisestate.ed	
O Yes Aug. 27th, 2017 18:15	
O No	

#### **Common error**

A <u>common error</u> is to declare a variable inside a loop whose value should persist across iterations. Below, the programmer expects the output to be 0, 1 (0+1), 3 (0+1+2), 6 (0+1+2+3), and 10 (0+1+2+3+4), but instead the output is just 0, 1, 2, 3, 4.

```
Figure 12.1.1: Common error: A variable declared within a loop block is (unexpectedly) re-initialized every iteration.

#include <stdio.h>
```

```
#include <stdio.h>

Ahram int main(void) {

BOISES

while (i < 5) {
    int tmpSum = 0;
    tmpSum = tmpSum + i; // Logic error: Sum is always just i
    printf("tmpSum: %dWn", tmpSum);
    i = i + 1;
}

return 0;
}
```

PARTICIPATION ACTIVITY

12.1.4: Common error of a variable declared within a loop block being reinitialized every iteration.

Given the following code, indicate i's value at the specified point.

```
for (int i = 0; i < 5; ++i) {
  int j = 0;

  j = j * i;
}</pre>
```

1) At the end of iteration i = 0.

**Check** Show answer

2) At the end of iteration i = 1.

Check Show answer

Ahram Kim

AhramKim@u.boisestate.edu\_BOISESTATECS253Fall2017

Aug. 27th, 2017 18:15

3) At the end of iteration i = 2.

Check Sho

Show answer

4) After the loo output? Type	op terminates, can j be e yes or no.	
Check	Show answer	

# Ahram Kim AhramKim@u.boisestate.edu

## 12.2 zyBooks built-in programming window

Aug. 27th, 2017 18:15

<b>*</b> •	ag. <i>Li</i> an, L	.017 10.15	
PARTICIPATION ACTIVITY	12.2.1: Programmin	g window.	
1 /* Enter 2	your program here */	Load default template	Pre-enter any input for program, the
2			Run
		Ahrai	n Kim
4		AhramKim@u	.boisestate.edu '
		POICECTATE	COEDE ALIONAT

BOISESTATECS253Fall2017 Aug. 27th, 2017 18:15