Variable Scope

The term SCOPE refers to the life cycle of variables. Specifically, the scope of a variable refers to where in a program that variable can be used. The scope of a variable is (or the variable is said to exist):

- From the line of code in which it is declared (using its type and name).
- To the end of the code block in which is declared.

Recall that a code block usually begins after the open curly brace seen in code such as a:

- Class declaration such as `public class MyClass {`
- Method declaration such as `public static int myMethod() {`
- Conditional statement such as `if (`*`expression`*`) {`
- Conditional clause such as `else {`
- While loop statement such as `while (`*`expression`*`) {`
- For loop statement such as `for (int i=0; i<10; i++) {`

And the code block ends with the matching close curly brace.

Look at the scope of the variables a and b in this simple example:

```
01 public static void main(String[] args) {
02    int a=0;
03    while (a<10) {
04       int b=0;
05       if (a%2==0) {
06          b+=a;
07          System.out.println(b);
08       }
09    }
10 }
```

- The scope of variable `a` begins on line `02` when it is declared.
- The scope of variable `a` ends at the end of the `main()` method code block on line `10`.
- The scope of variable `b` begins with each iteration of the "while" loop on line `04` and ends at the end of the "while" loop's code block on line `09`.

Special Note: The scope actually begins with the keyword, expression, or name in each of these cases!

Recall that an "if," "else," "while," and "for" statement always includes one statement that may be executed and that the use of the curly braces to create a code block simply extends that statement. Scope is usually considered to be inside the code block because the scope of,