 

**Java Lesson: Variable Scope
Last Updated: 5/11/2017**[**mr Hanley**](http://hanley.co.nr)

**Objective:** Student will understand that there are global and local variables. Global variables are in scope throughout the class while local variables are local to the codeblock there are declared in. Parameters, which are inputs to methods, are in scope inside the method only. Local variables in one method cannot be seen in another method.

In java, variables are used to store information which changes throughout the execution of a program. A well designed program utilizes global variables to store the important information that endures through the program. These usually have carefully chosen\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. Variables that act as temporary “scratch paper” if you will are known as Local Variables\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Parameters allow methods to be \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_adjusting for different options.

|  |  |  |
| --- | --- | --- |
| Type of Var Scope | How to identify | Where they can be seen |
| Global |  |  |
| Local |  |  |
| Parameter |  |  |

Example,

public class LemonAidStand {

 int numBrownies, numCookies, ouncesLemonaid;

 double dollars;

 public LemonaidStand() {

 numBrownies = 50; numCookies = 60; ouncesLemonaid = 500;

 dollars = 0;

 displayMenu();

 }

 public void displayMenu() {

 System.out.println(“Brownies .50 Cookies .25 Lemonaid 1.50”);

 double cost;

 int b,c,lem;

 b = input.nextInt();

 c = input.nextInt();

 lem = input.nextInt();

 cost = .50\*b + ; //you finish

 numBrownies -= b; numCookies -= c; ouncesLemonaid -= 8;

 printInventory(”both”); //print both money earned and food left

}

public void printInventory(String preference) {

 String display;

if(numBrownies > 0) {

 display = “Brownies “ + numBrownies;

}

 if(numCookies > 0) {

 display += “Cookies “ + numCookies;

}

 if(ouncesLemonaid > 0) {

 display += “Lemonaid “ + ouncesLemonaid;

 }

 if(preference.equals(“both”) {

 display += “$$$ “ + dollars;

 }

 System.out.println(display);

}

}//end of class

//Can the printInventory method use b, c, l and cost? Why or why not?

Note: Sometimes local variables are declared inside the { } of a if or loop

This means their scope is only inside that loop after they are declared

Circle any variables out of scope in this example

public void methodWithBugs() {

//line1 int x;

//line2 x=input.nextInt();

//line3 if (x >= 14) {

//line4 double y = 8;

//line5 x++;

//line6 y--;

//line7 }

//line8 int i=0;

//line9 while (i<10) {

//line10 if (y > 4) {

//line11 x++;

//line12 double m;

//line13 m = 9;

//line14 System.out.println(i);

//line15 }

//line16 System.out.println(m);

//line17 i++;

//line18 }

 }

NOTE: Two variables CANNOT have the same name in the same scope

However, variables in different scopes can have same name

The local variable **shadows** the global variable

You can use this.sum to access a global variable called sum if there is a local variable called sum