Lab 6: Scope

Solutions

Main 1: \( a = 1 \ b = 5 \ g = 4 \ x = 0 \)
All variables are their initial values.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Declared</th>
<th>Defined</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>( b )</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>( g )</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>A global variable... visible to all scope</td>
</tr>
<tr>
<td>( x )</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

Main 2: \( a = 7 \ b = 5 \ g = 10 \ x = 0 \)
Inside the code braces from line 13 to line 19, \( a \) and \( g \) are modified. Since \( g \) is a global, this change persists outside of the braces whereas the change to \( a \) does not.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Declared</th>
<th>Defined</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>7</td>
<td>14</td>
<td>14</td>
<td>This variable is live(in scope) from lines 13-19</td>
</tr>
<tr>
<td>( b )</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>( g )</td>
<td>10</td>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>( x )</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

Function 1: \( a = -4 \ x = 1 \ g = 6 \)
Since we’ve left the previous code block, \( a \) now equals 1 again.
Function 1 takes two arguments internally called \( x \) and \( y \). The variable \( y \) is passed into the function by reference, so when \( y \) is modified in Function 1, the corresponding variable passed into Function 1 in the \( y \) position will be modified as well.

\[
\text{function1}(\text{int } x, \text{int } \&y); \quad -- \ a = 1 => x \text{ and } b = 5 => y
\]

Function 1 (int \( x \), int &\( y \)):
- Declares/defines a new \( a = x \ (1) - y \ (5) = -4 \)
- Sets \( g = x \ (1) + y \ (5) = 6 \)
- Sets \( y = a \ (-4) + g \ (6) = 2 \)
- Returns \( a \ (-4) \)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Declared</th>
<th>Defined</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>-4</td>
<td>31</td>
<td>32</td>
<td>This variable is in scope within function1</td>
</tr>
<tr>
<td>( g )</td>
<td>6</td>
<td>5</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>( x )</td>
<td>1</td>
<td>30</td>
<td>20</td>
<td>( x ) received its value when function1( (a, b) ) was called</td>
</tr>
<tr>
<td>( y )</td>
<td>2</td>
<td>30</td>
<td>20, 34</td>
<td>( y ) is similar to ( x ) but was assigned to ( a+g ) inside function1 at line 34</td>
</tr>
</tbody>
</table>

Main 3: \( a = 1 \ b = 2 \ g = 6 \ x = -4 \)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Declared</th>
<th>Defined</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>( b )</td>
<td>2</td>
<td>8</td>
<td>34</td>
<td>Modified by ( y ) inside function1</td>
</tr>
<tr>
<td>( g )</td>
<td>6</td>
<td>5</td>
<td>33</td>
<td>Inside function1 ( g ) was set to 6</td>
</tr>
<tr>
<td>( x )</td>
<td>-4</td>
<td>9</td>
<td>20</td>
<td>( x ) was set to the result of function1. Since function1 returns it’s ( a ), now ( x = -4 )</td>
</tr>
</tbody>
</table>
**Function1: a = 1 x = 2 g = 3**

This call to function1 was made inside function2 at line 46.
Function2 was called with the arguments (a,b) or (1,2).
Function2(a,b); -- a maps to b and b maps to x inside function2

Function2 (int b, int &x):
- Declares/defines temp = b (1)
- Sets b = x (2)
- Sets x = temp (1)
- Sets temp = function1(b (2),x (1)) (1)

Function 1 (int x, int &y):
- Declares/defines a new a = x (2) - y (1) = 1
- Sets g = x (2) + y (1) = 3
- Sets y = a (1) + g (3) = 4
- Returns a (1)

Because b was passed by reference into function2 as x and x was passed by reference into function1 as y, the variable b in scope of main now equals 4.
Sets g = temp (1) * x (4) = 4
Returns temp (1)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Declared</th>
<th>Defined</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>31</td>
<td>32</td>
<td>This variable is in scope within function1</td>
</tr>
<tr>
<td>g</td>
<td>3</td>
<td>5</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>1</td>
<td>30</td>
<td>46</td>
<td>x received its value when function1(a,b) was called</td>
</tr>
<tr>
<td>y</td>
<td>4</td>
<td>30</td>
<td>46, 34</td>
<td>y is similar to x but was assigned to a+g inside function1 at line 34</td>
</tr>
</tbody>
</table>

**Main 4: a = 1 b = 4 g = 4 x = 1**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Declared</th>
<th>Defined</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>4</td>
<td>8</td>
<td>34</td>
<td>Modified by y inside function1</td>
</tr>
<tr>
<td>g</td>
<td>4</td>
<td>5</td>
<td>47</td>
<td>Inside function2 g was set to 4</td>
</tr>
<tr>
<td>x</td>
<td>1</td>
<td>9</td>
<td>24</td>
<td>x was set to the result of function2. Since function2 returns temp, now x = -4</td>
</tr>
</tbody>
</table>
```c++
#include<iostream>
using namespace std;

int function1(int, int &);
int function2(int, int &);

int g = 4;

int main(){
    int a = 1;
    int b = 5;
    int x = 0;
    cout << "Main 1: a = " << a << " b = " << b;
    cout << " g = " << g << " x = " << x;
    cout << endl << endl << endl << endl << endl;
    {
        int a = 7;
        g = 10;
        cout << "Main 2: a = " << a << " b = " << b;
        cout << " g = " << g << " x = " << x;
        cout << endl << endl << endl << endl << endl;
    }
    x = function1(a,b);
    cout << "Main 3: a = " << a << " b = " << b;
    cout << " g = " << g << " x = " << x;
    cout << endl << endl << endl << endl << endl;
    x = function2(a,b);
    cout << "Main 4: a = " << a << " b = " << b;
    cout << " g = " << g << " x = " << x;
    cout << endl << endl << endl << endl << endl;
}

int function1(int x, int &y){
    int a;
    a = x-y;
    g = x+y;
    y = a+g;
    cout << "Function1: a = " << a << " x = " ;
    cout << x << " g = " << g;
    cout << endl << endl << endl << endl << endl;
    return a;
}

int function2(int b, int &x){
    int temp;
    temp = b;
    b = x;
    x = temp;
    temp = function1(b,x);
    g = temp*x;
    return temp;
}
```