

Static Variables

July 20, 2009

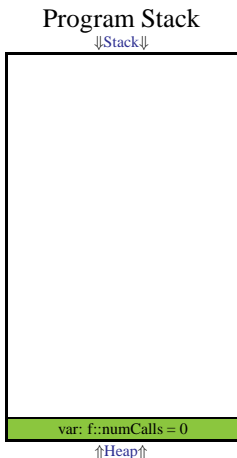
The following example demonstrates the use of static variables in C++ functions. static variables have the following interesting traits:

1. static variables must be initialized with values (no GARBAGE values!)
2. static variables retain their value across function calls.

```
1  /**
2  *  static variables in the Heap
3  */
4  #include <iostream>
5  using namespace std;
6  int f( int x )
7  {
8      static int numCalls(0);
9      return x*numCalls++;
10 }
11 int g( int y )
12 {
13     static int numCalls(0);
14     return y*numCalls++;
15 }
16 int main()
17 {
18     int Af( f(2) ), Ag( g(2) );
19     int Bf( f(2) ), Bg( g(2) );
20     int Cf( f(2) ), Cg( g(2) );
21     cout << "Af=" << Af << "\n" <<
22          "Ag=" << Ag << endl;
23     cout << "Bf=" << Bf << "\n" <<
24          "Bg=" << Bg << endl;
25     cout << "Cf=" << Cf << "\n" <<
26          "Cg=" << Cg << endl;
27     return 0;
28 }
```

The Operating System initializes `f::numCalls`

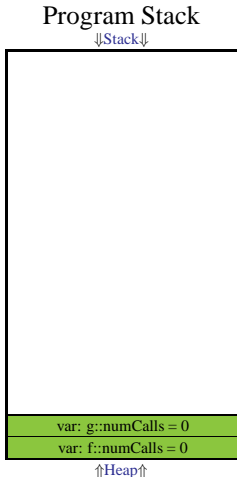
<OS>



```
6 int f( int x )
7 {
8     static int numCalls(0);
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     static int numCalls(0);
14     return y*numCalls++;
15 }
16 int main()
17 {
18     int Af( f(2) ), Ag( g(2) );
19     int Bf( f(2) ), Bg( g(2) );
20     int Cf( f(2) ), Cg( g(2) );
21     cout << "Af=" << Af << " " <<
22          "Ag=" << Ag << endl;
23     cout << "Bf=" << Bf << " " <<
24          "Bg=" << Bg << endl;
25     cout << "Cf=" << Cf << " " <<
26          "Cg=" << Cg << endl;
27     return 0;
28 }
```

The Operating System initializes `g::numCalls`

<OS>



```
6 int f( int x )
7 {
8     static int numCalls(0);
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     static int numCalls(0);
14     return y*numCalls++;
15 }
16 int main()
17 {
18     int Af( f(2) ), Ag( g(2) );
19     int Bf( f(2) ), Bg( g(2) );
20     int Cf( f(2) ), Cg( g(2) );
21     cout << "Af=" << Af << " " <<
22         "Ag=" << Ag << endl;
23     cout << "Bf=" << Bf << " " <<
24         "Bg=" << Bg << endl;
25     cout << "Cf=" << Cf << " " <<
26         "Cg=" << Cg << endl;
27     return 0;
28 }
```

The Operating System calls main ()

<OS>



```
6 int f( int x )
7 {
8     static int numCalls(0);
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     static int numCalls(0);
14     return y*numCalls++;
15 }
16 int main()
17 {
18     int Af( f(2) ), Ag( g(2) );
19     int Bf( f(2) ), Bg( g(2) );
20     int Cf( f(2) ), Cg( g(2) );
21     cout << "Af=" << Af << "\n" <<
22         "Ag=" << Ag << endl;
23     cout << "Bf=" << Bf << "\n" <<
24         "Bg=" << Bg << endl;
25     cout << "Cf=" << Cf << "\n" <<
26         "Cg=" << Cg << endl;
27     return 0;
28 }
```


Line 27: Return to OS

discard call frame, return to caller

<OS> \implies main



```
6 int f( int x )
7 {
8     static int numCalls(0);
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     static int numCalls(0);
14     return y*numCalls++;
15 }
16 int main()
17 {
18     int Af( f(2) ), Ag( g(2) );
19     int Bf( f(2) ), Bg( g(2) );
20     int Cf( f(2) ), Cg( g(2) );
21     cout << "Af=" << Af << " " <<
22         "Ag=" << Ag << endl;
23     cout << "Bf=" << Bf << " " <<
24         "Bg=" << Bg << endl;
25     cout << "Cf=" << Cf << " " <<
26         "Cg=" << Cg << endl;
27     return 0; // <==
28 }
```

Return from main

<OS>



```
6 int f( int x )
7 {
8     static int numCalls(0);
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     static int numCalls(0);
14     return y*numCalls++;
15 }
16 int main()
17 {
18     int Af( f(2) ), Ag( g(2) );
19     int Bf( f(2) ), Bg( g(2) );
20     int Cf( f(2) ), Cg( g(2) );
21     cout << "Af=" << Af << " " <<
22         "Ag=" << Ag << endl;
23     cout << "Bf=" << Bf << " " <<
24         "Bg=" << Bg << endl;
25     cout << "Cf=" << Cf << " " <<
26         "Cg=" << Cg << endl;
27     return 0;
28 }
```

finis