

Global Variables

July 20, 2009

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

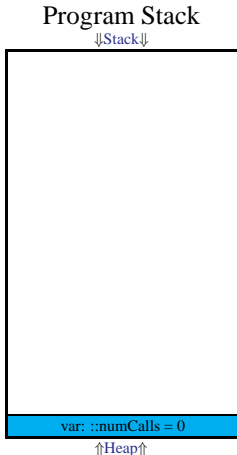
1. Global variables **should always** be initialized with values (no GARBAGE values!)
2. Global variables are visible to and shared by **all** the functions in a C++ source file.

Note that there is no global keyword in C++. Since global variables do not belong to any one particular function, they are written as `::varName` in C++.

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

<OS>



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

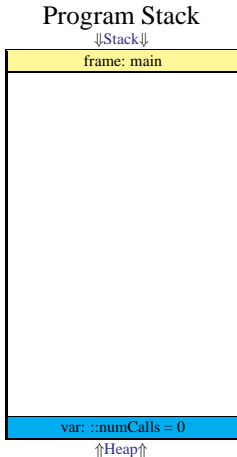


```
6 int numCalls (0);
7 int f( int x )
8 {
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     return y*numCalls++;
14 }
15 int main()
16 {
17     int Af( f(2) ), Ag( g(2) );
18     int Bf( f(2) ), Bg( g(2) );
19     int Cf( f(2) ), Cg( g(2) );
20     cout <<
21         "Af=" << Af << " "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

create call frame for main

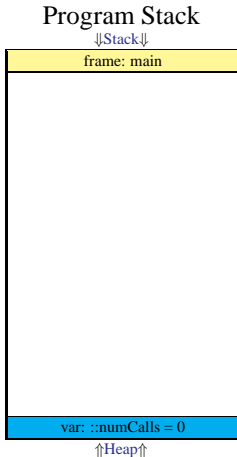
<OS>



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

Line 15: Function main entry

<OS> \implies main



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


Line 18: Return from f

<OS> \implies main

Program Stack

↓Stack↓

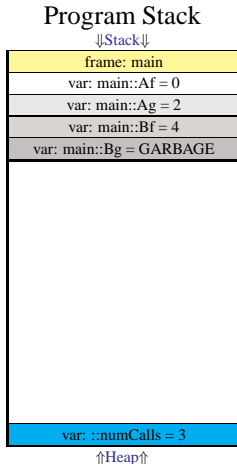
frame: main
var: main::Af = 0
var: main::Ag = 2
var: main::Bf = 4
var: ::numCalls = 3

↑Heap↑

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

Line 18: Declare Bg

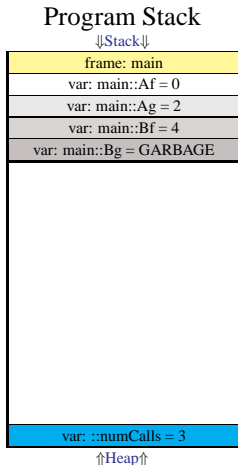
<OS> \implies main



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

Line 18: Call g(2) for Bg

<OS> \implies main

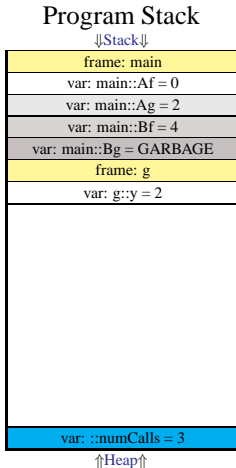


```
6 int numCalls(0);
7 int f( int x )
8 {
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     return y*numCalls++;
14 }
15 int main()
16 {
17     int Af( f(2) ), Ag( g(2) );
18     int Bf( f(2) ), Bg( g(2) ); // <==
19     int Cf( f(2) ), Cg( g(2) );
20     cout <<
21         "Af=" << Af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


Line 18: Call g(2) for Bg

create formal param y of g

<OS> \implies main



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


Line 13: Declare tempG for Return Value

<OS> \implies main \implies g

Program Stack

↓Stack↓

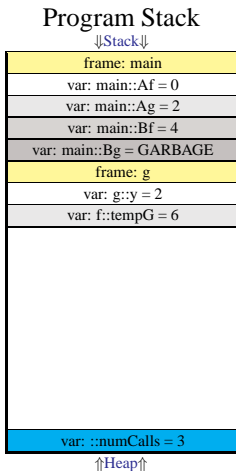
frame: main
var: main::Af = 0
var: main::Ag = 2
var: main::Bf = 4
var: main::Bg = GARBAGE
frame: g
var: g::y = 2
var: f::tempG = GARBAGE
var: ::numCalls = 3

↑Heap↑

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

Line 13: Calc $y \cdot \text{numCalls}$, store in tempG

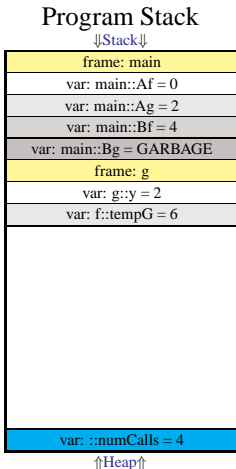
$\langle \text{OS} \rangle \implies \text{main} \implies \text{g}$



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

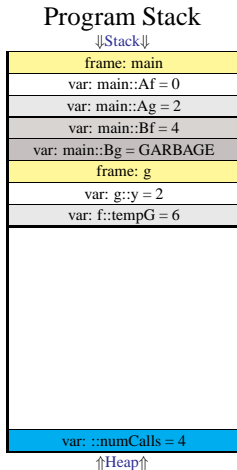
<OS> \implies main \implies g



```
6 int numCalls(0);
7 int f( int x )
8 {
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     return y*numCalls++;           // <==
14 }
15 int main()
16 {
17     int Af( f(2) ), Ag( g(2) );
18     int Bf( f(2) ), Bg( g(2) );
19     int Cf( f(2) ), Cg( g(2) );
20     cout <<
21         "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 13: Return value of tempG

<OS> \implies main \implies g

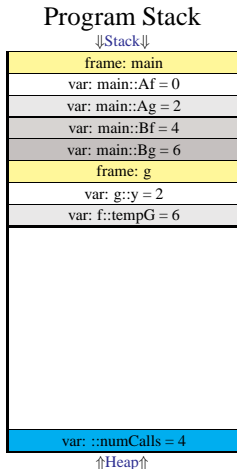


```
6 int numCalls (0);
7 int f( int x )
8 {
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     return y*numCalls++;           // <==
14 }
15 int main()
16 {
17     int Af( f(2) ), Ag( g(2) );
18     int Bf( f(2) ), Bg( g(2) );
19     int Cf( f(2) ), Cg( g(2) );
20     cout <<
21         "Af="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

Line 13: Return value of tempG

copy return value

<OS> \implies main \implies g

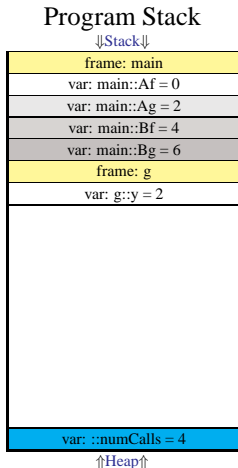


```
6 int numCalls (0);
7 int f( int x )
8 {
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     return y*numCalls++;           // <==
14 }
15 int main()
16 {
17     int Af( f(2) ), Ag( g(2) );
18     int Bf( f(2) ), Bg( g(2) );
19     int Cf( f(2) ), Cg( g(2) );
20     cout <<
21         "Af="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

Line 13: Return value of tempG

discard the auto (local) variable f :: tempG

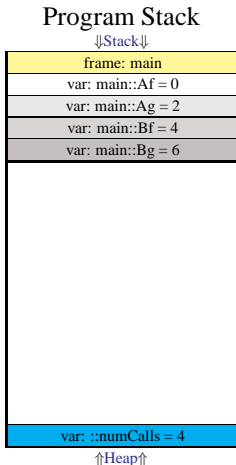
<OS> \implies main \implies g



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


Line 18: Return from g

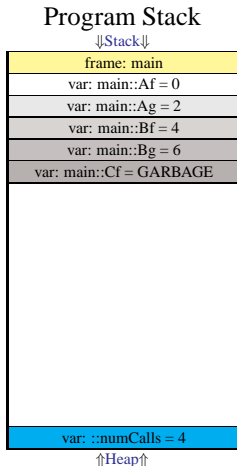
<OS> \implies main



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


Line 19: Call f(2) for Cf

<OS> \implies main



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


Line 19: Call f (2) for Cf

create formal param x of f

<OS> \implies main

Program Stack

↓Stack↓

frame: main
var: main::Af = 0
var: main::Ag = 2
var: main::Bf = 4
var: main::Bg = 6
var: main::Cf = GARBAGE
frame: f
var: f::x = 2
var: ::numCalls = 4

↑Heap↑

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


Line 9: Calc x*numCalls, store in tempF

<OS> \implies main \implies f

Program Stack

↓Stack↓

frame: main
var: main::Af = 0
var: main::Ag = 2
var: main::Bf = 4
var: main::Bg = 6
var: main::Cf = GARBAGE
frame: f
var: f::x = 2
var: f::tempF = 8
var: ::numCalls = 4

↑Heap↑

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


Line 9: Return value of tempF

<OS> \implies main \implies f

Program Stack

↓Stack↓

frame: main
var: main::Af = 0
var: main::Ag = 2
var: main::Bf = 4
var: main::Bg = 6
var: main::Cf = GARBAGE
frame: f
var: f::x = 2
var: f::tempF = 8
var: ::numCalls = 5

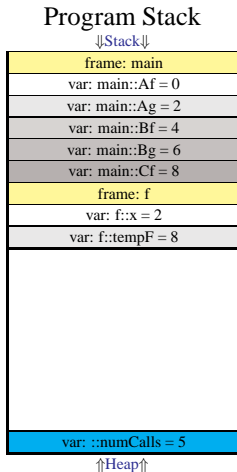
↑Heap↑

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

Line 9: Return value of tempF

copy return value

<OS> \implies main \implies f

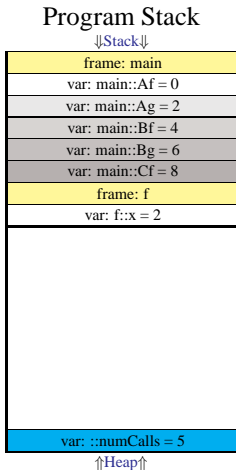


```
6 int numCalls (0);
7 int f( int x )
8 {
9     return x*numCalls++;           // <==
10 }
11 int g( int y )
12 {
13     return y*numCalls++;
14 }
15 int main()
16 {
17     int Af( f(2) ), Ag( g(2) );
18     int Bf( f(2) ), Bg( g(2) );
19     int Cf( f(2) ), Cg( g(2) );
20     cout <<
21         "Af="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

Line 9: Return value of tempF

discard the auto (local) variable f :: tempF

<OS> \implies main \implies f



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


Line 13: Declare tempG for Return Value

<OS> \implies main \implies g

Program Stack

↓Stack↓

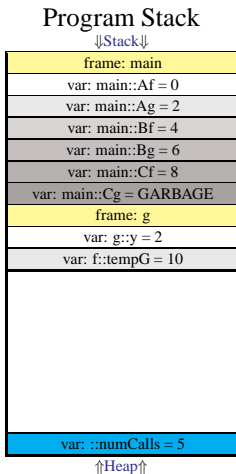
frame: main
var: main::Af = 0
var: main::Ag = 2
var: main::Bf = 4
var: main::Bg = 6
var: main::Cf = 8
var: main::Cg = GARBAGE
frame: g
var: g::y = 2
var: f::tempG = GARBAGE
var: ::numCalls = 5

↑Heap↑

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

Line 13: Calc $y * \text{numCalls}$, store in tempG

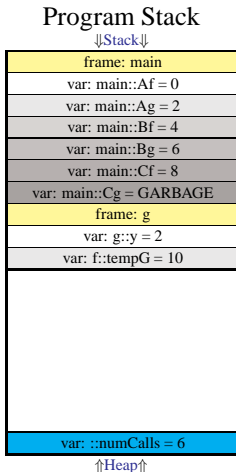
$\langle \text{OS} \rangle \implies \text{main} \implies \text{g}$



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

Line 13: Increment numCalls

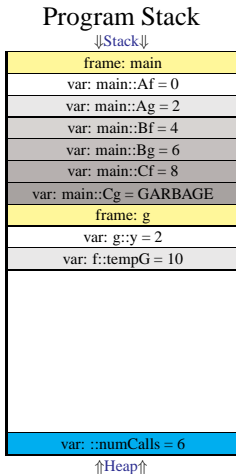
<OS> \implies main \implies g



```
6 int numCalls(0);
7 int f( int x )
8 {
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     return y*numCalls++;           // <==
14 }
15 int main()
16 {
17     int Af( f(2) ), Ag( g(2) );
18     int Bf( f(2) ), Bg( g(2) );
19     int Cf( f(2) ), Cg( g(2) );
20     cout <<
21         "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 13: Return value of tempG

<OS> \implies main \implies g



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


Line 13: Return value of tempG

discard call frame, return to caller

<OS> \implies main \implies g

Program Stack

↓Stack↓

frame: main
var: main::Af = 0
var: main::Ag = 2
var: main::Bf = 4
var: main::Bg = 6
var: main::Cf = 8
var: main::Cg = 10
var: ::numCalls = 6

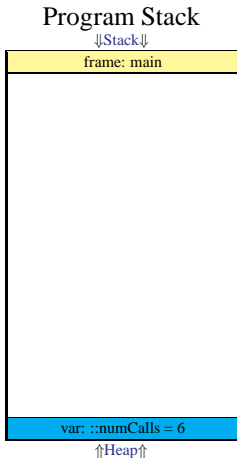
↑Heap↑

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


Line 27: Return to OS

discard the auto (local) variable `main::Af`

<OS> \implies main

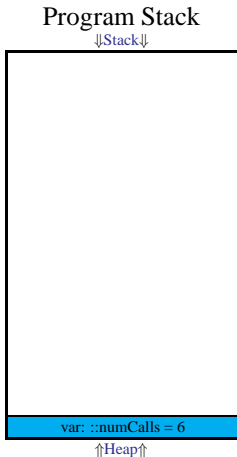


```
6 int numCalls (0);
7 int f( int x )
8 {
9     return x*numCalls++;
10 }
11 int g( int y )
12 {
13     return y*numCalls++;
14 }
15 int main()
16 {
17     int Af( f(2) ), Ag( g(2) );
18     int Bf( f(2) ), Bg( g(2) );
19     int Cf( f(2) ), Cg( g(2) );
20     cout <<
21         "Af=" << Af << " "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

Line 27: Return to OS

discard call frame, return to caller

<OS> \implies main



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