

```

// Program to demonstrate private and shared variables, including
// a parallel section that begins in a function.
//
#include <iostream>
#include <omp.h>
using namespace std ;

// -----
int f( int a, int b )
{
    int c, d, z ;
    int id ;

    c = 17 ; d = 19 ; z = 149 ;

    cout << "seq:  a = " << a << ", b = " << b
         << ", c = " << c << ", d = " << d << ", z = " << z  << endl ;

#pragma omp parallel shared(a,c) private(b,d,id)
{
    id = omp_get_thread_num() ;
#pragma omp critical
{
    cout << "(A:" << id << ")" << "    a = " << a << ", b = " << b
         << ", c = " << c << ", d = " << d << ", z = " << z  << endl ;
}

    a = 100 + id ; b = 200 + id ; c = 300 + id ; d = 300 + id ;
    z = 1000 + id ;

#pragma omp critical
{
    cout << "(B:" << id << ")" << "    a = " << a << ", b = " << b
         << ", c = " << c << ", d = " << d << ", z = " << z  << endl ;
}
} // End parallel section

    cout << "seq:  a = " << a << ", b = " << b
         << ", c = " << c << ", d = " << d << ", z = " << z  << endl ;

} // End function f()

// -----
int main()
{
    int x, y ;

    x = 11 ; y = 13 ;
    f(x,y) ;
}

```

===== Sample Session with Two Threads =====

```

cosmos%
cosmos% g++ -fopenmp pl.cc
cosmos% a.out
seq:  a = 11, b = 13, c = 17, d = 19, z = 149
(A:0)  a = 11, b = 0, c = 17, d = 0, z = 149
(B:0)  a = 100, b = 200, c = 300, d = 300, z = 1000
(A:1)  a = 100, b = -1, c = 300, d = -1, z = 1000
(B:1)  a = 101, b = 201, c = 301, d = 301, z = 1001
seq:  a = 101, b = 13, c = 301, d = 19, z = 1001

```