delegate bool D();

delegate bool D2(int i);

class Test

{

D del;

D2 del2;

public void TestMethod(int input)

{

int j = 0;

// Initialize the delegates with lambda expressions.

// Note access to 2 outer variables.

// del will be invoked within this method.

del = () => { j = 10; return j > input; };

// del2 will be invoked after TestMethod goes out of scope.

del2 = (x) => {return x == j; };

// Demonstrate value of j:

// Output: j = 0

// The delegate has not been invoked yet.

Console.WriteLine("j = {0}", j); // Invoke the delegate.

bool boolResult = del();

// Output: j = 10 b = True

Console.WriteLine("j = {0}. b = {1}", j, boolResult);

}

static void Main()

{

Test test = new Test();

test.TestMethod(5);

// Prove that del2 still has a copy of

// local variable j from TestMethod.

bool result = test.del2(10);

// Output: True

Console.WriteLine(result);

Console.ReadKey();

}

}