class Test

{

delegate void TestDelegate(string s);

static void M(string s)

{

Console.WriteLine(s);

}

static void Main(string[] args)

{

// Original delegate syntax required

// initialization with a named method.

TestDelegate testDelA = new TestDelegate(M);

// C# 2.0: A delegate can be initialized with

// inline code, called an "anonymous method." This

// method takes a string as an input parameter.

TestDelegate testDelB = delegate(string s) { Console.WriteLine(s); };

// C# 3.0. A delegate can be initialized with

// a lambda expression. The lambda also takes a string

// as an input parameter (x). The type of x is inferred by the compiler.

TestDelegate testDelC = (x) => { Console.WriteLine(x); };

// Invoke the delegates.

testDelA("Hello. My name is M and I write lines.");

testDelB("That's nothing. I'm anonymous and ");

testDelC("I'm a famous author.");

// Keep console window open in debug mode.

Console.WriteLine("Press any key to exit.");

Console.ReadKey();

}

}

/\* Output:

Hello. My name is M and I write lines.

That's nothing. I'm anonymous and

I'm a famous author.

Press any key to exit.

\*/