```csharp
1  //using is a directive
2  //System is a name space
3  //name space is a collection of features that our needs to run
4  using System;
5  //public means accessible anywhere
6  //partial means this class is split over multiple files
7  //class is a keyword and think of it as the outermost level of grouping
8  //:System.Web.UI.Page means our page inherits the features of a Page
9  public delegate bool Compare(double x, double y);
10 public delegate double Multiply(double x, double y);
11 public partial class _Default : System.Web.UI.Page
12 {
13     protected void Button1_Click(object sender, EventArgs e)
14     {
15         double x = 10, y = 25; //declare two variables
16         //the two variables are accessible inside the lambda expressions
17         Compare comp = (a, b) => (a == b);//define comparison lambda
18         //invoke the lambda in the line below
19         sampLabel.Text = $"{x} and {y} are equal is {comp(x, y).ToString().ToLower()}";
20         Multiply mult = (a, b) => (a * b);//line define a lambda for multiplication
21         sampLabel.Text += $"<br>{x}*{y} is {mult(x, y)}";//invoke the multiplication lambda
22         double[] dubsArray = new double[] { 1, 2, 3, 4, 5 };//make array of doubles
23         //actions encapsulate functions that do not return a value
24         //but actions can accept arguments to operate on
25         Action<double> showDouble = (a) => sampLabel.Text += "<br>" + (a * a);
26         //it's now possible to perform the action on each d repeatdely
27         foreach (var d in dubsArray)
28         {
29             showDouble(d);
30         }
31     }
32 }
33
34
35
```