

Step1: Object obj1 is created which is of value type. The datamember is x and its value is initialised to 0

```
Program.cs | ReferenceType.cs | RiskInputDataMerger.cs | RiskCalculator.cs | RDSDataModel.cs | TDSDataModel.cs | Program.cs
ReferenceType.cs
44 | obj3.Setx(400);
45 |
46 | Console.WriteLine(obj1.GetX());
47 | Console.WriteLine(obj2.GetX());
48 | Console.WriteLine(obj3.GetX());
49 |
50 | }
51 |
52 | 0 references
53 | static void Main()
54 | {
55 |     ValueType obj1 = new ValueType(); < 1ms elapsed
56 |     obj1.Setx(100);
57 |     Test(obj1);
58 |     ReferenceType obj2 = new ReferenceType();
59 |     NewTest(obj2);
60 | }
61 | 1 reference
62 | static void Test(ValueType arg)
63 | {
64 |     arg.Setx(200);
65 | }
```

Name	Value	Type
obj1	(demo.ValueType)	demo.ValueType
obj1.x	0	int

Name	Language
ReferenceType.cs:demo.EntryPoint.Main() Line 54	C#

Autos | Locals | Watch 1 | Call Stack | Breakpoints | Exception Settings | Command Window | Immediate Window | Output

Step2: The program control is still in Entrypoint.Main()

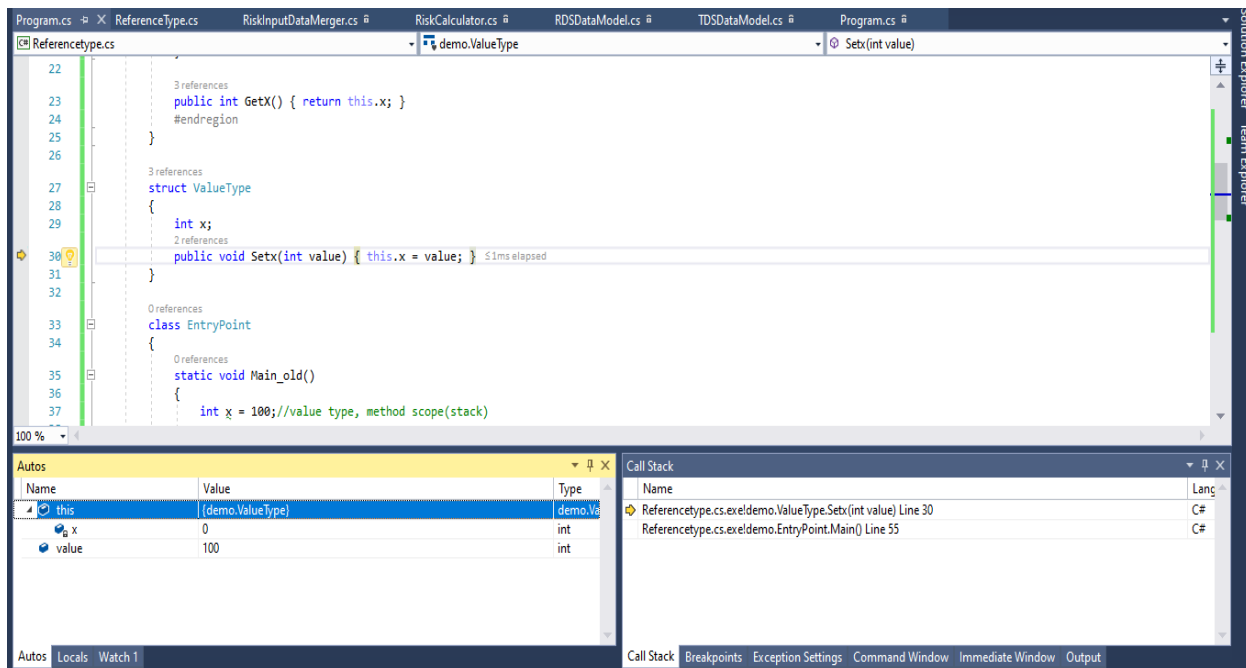
```
Program.cs | ReferenceType.cs | RiskInputDataMerger.cs | RiskCalculator.cs | RDSDataModel.cs | TDSDataModel.cs | Program.cs
ReferenceType.cs
44 | obj3.Setx(400);
45 |
46 | Console.WriteLine(obj1.GetX());
47 | Console.WriteLine(obj2.GetX());
48 | Console.WriteLine(obj3.GetX());
49 |
50 | }
51 |
52 | 0 references
53 | static void Main()
54 | {
55 |     ValueType obj1 = new ValueType();
56 |     obj1.Setx(100); < 1ms elapsed
57 |     Test(obj1);
58 |     ReferenceType obj2 = new ReferenceType();
59 |     NewTest(obj2);
60 | }
61 | 1 reference
62 | static void Test(ValueType arg)
63 | {
64 |     arg.Setx(200);
65 | }
```

Name	Value	Type
obj1	(demo.ValueType)	demo.ValueType
obj1.x	0	int

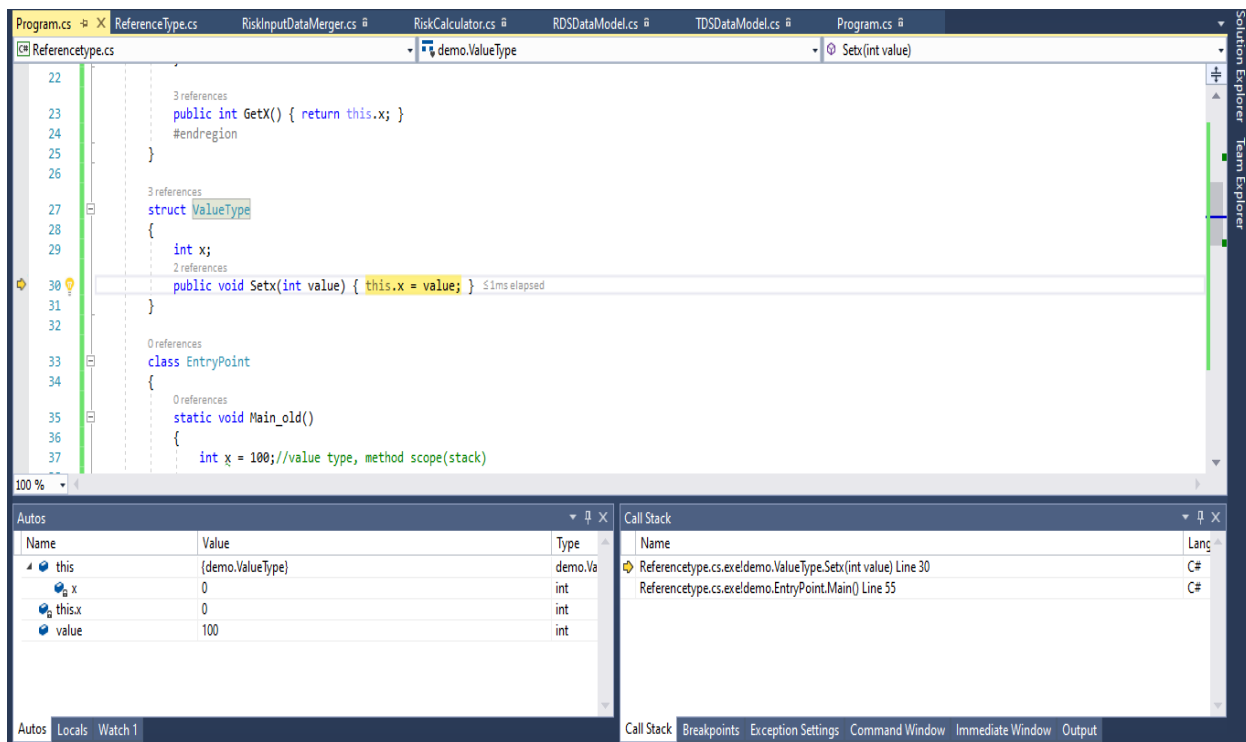
Name	Language
ReferenceType.cs:demo.EntryPoint.Main() Line 55	C#

Autos | Locals | Watch 1 | Call Stack | Breakpoints | Exception Settings | Command Window | Immediate Window | Output

Step3: The control is now transferred to ValueType.Setx and obj1 address and value 100 is sent as parameter.



Step 4: The value of x and this.x (both are same) is 0 and Value=100



Step 5: Value of x in obj1 is now set to 100.

The screenshot shows the Visual Studio IDE with the following components:

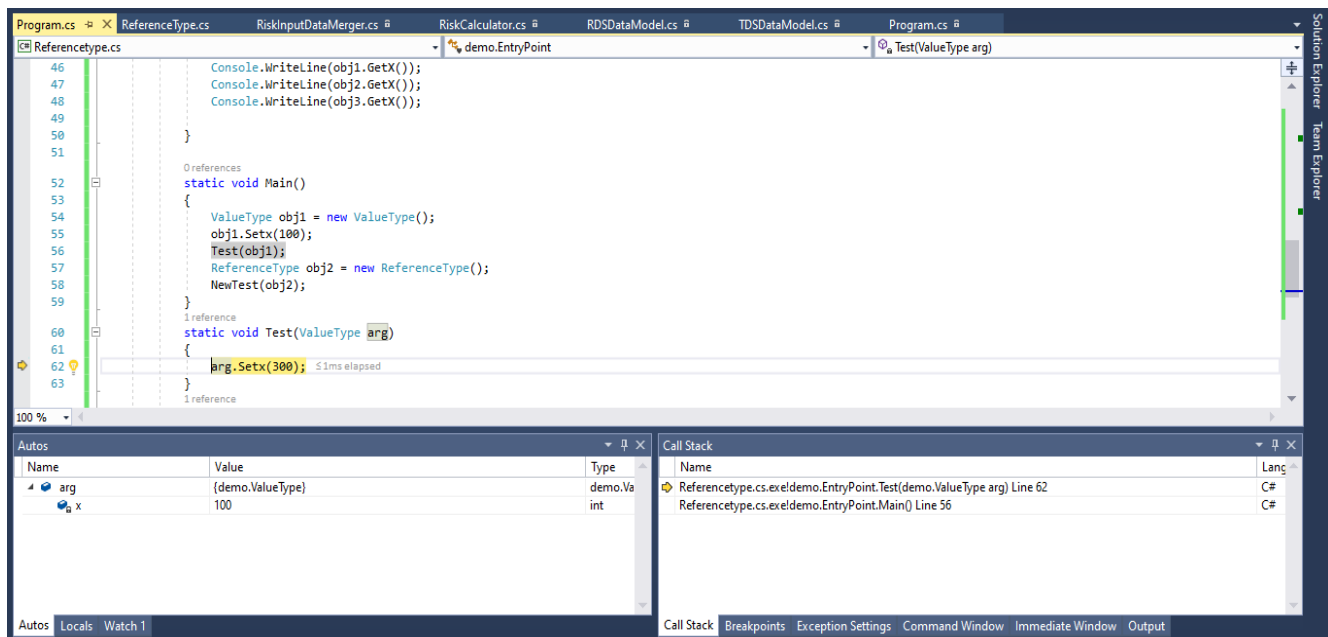
- Code Editor:** Displays `ReferenceType.cs`. The current line is 55: `obj1.Setx(100);`. The `Main()` method is visible, showing the creation of `obj1` and `obj2`, and the call to `Test(obj1)`. The `Test` method is also visible, showing `arg.Setx(300);`.
- Autos Window:** Shows the current state of variables. `obj1` is of type `demo.ValueType` and `x` is of type `int` with a value of 100.
- Call Stack:** Shows the current call stack with one entry: `ReferenceType.cs:exeldemo.EntryPoint.Main() Line 55`.

Step 6: The control is at `EntryPoint.Test()`. The value of x is sent as an argument along with `obj1` address.

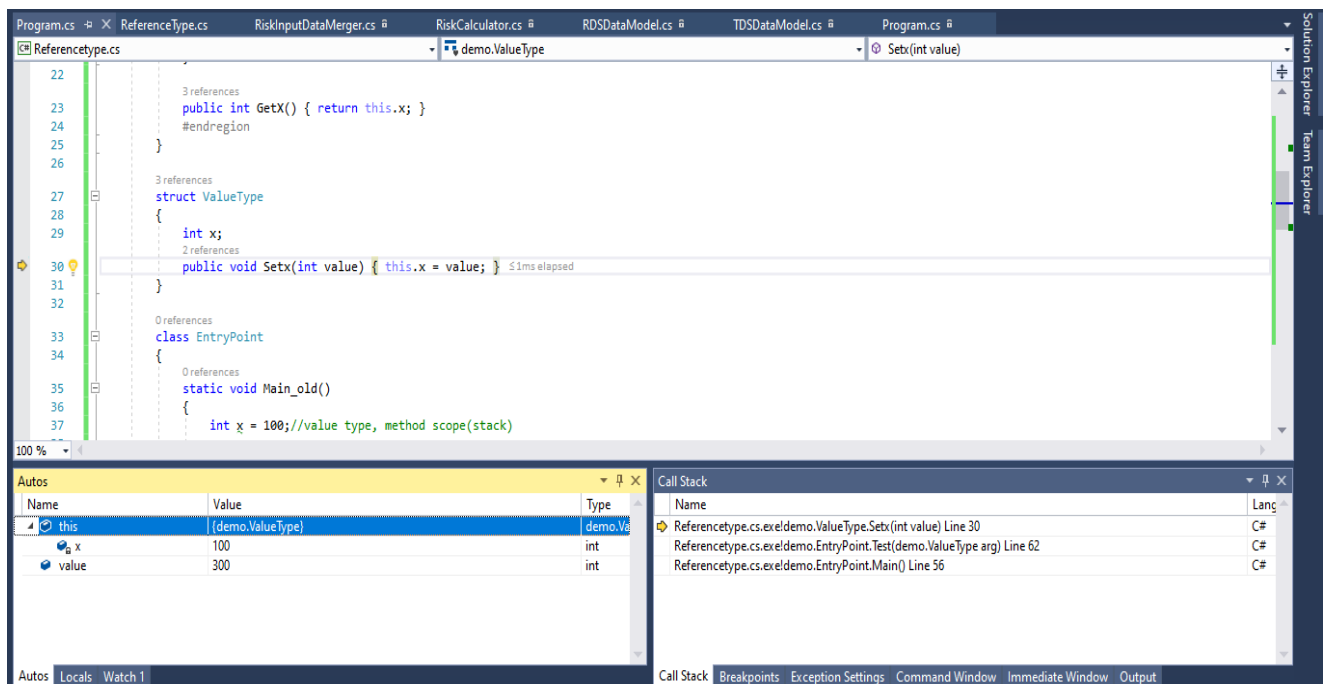
The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays `ReferenceType.cs`. The current line is 61: `arg.Setx(300);`. The `Main()` method is visible, showing the creation of `obj1` and `obj2`, and the call to `Test(obj1)`. The `Test` method is also visible, showing `arg.Setx(300);`.
- Autos Window:** Shows the current state of variables. `arg` is of type `demo.ValueType` and `x` is of type `int` with a value of 100.
- Call Stack:** Shows the current call stack with two entries: `ReferenceType.cs:exeldemo.EntryPoint.Test(demo.ValueType arg) Line 61` and `ReferenceType.cs:exeldemo.EntryPoint.Main() Line 56`.

Step 7: Calls Setx function and transfer its control to there.



Step 8: Now the control is in `ValueType.Setx(int value)`. Value is sent is 300 along with `obj1` address but value is `x` is 100



Step 9: Assigns the x value as 300. After this control comes back to EntryPoint.Test().

The screenshot shows the Visual Studio IDE with the `ReferenceType.cs` file open. The `Setx(int value)` method in the `demo.ValueType` struct is selected. The code in the editor is as follows:

```
22  
23 public int GetX() { return this.x; }  
24 #endregion  
25  
26  
27 3 references  
28 struct ValueType  
29 {  
30     2 references  
31     public void Setx(int value) { this.x = value; }  
32 }  
33  
34 0 references  
35 class EntryPoint  
36 {  
37     0 references  
38     static void Main_old()  
39     {  
40         int x = 100; //value type, method scope(stack)  
41     }  
42 }
```

The `Autos` window shows the following variables:

Name	Value	Type
this	(demo.ValueType)	demo.Va
x	300	int
this.x	300	int
value	300	int

The `Call Stack` window shows the following frames:

Name	Lang
ReferenceType.cs:demo.ValueType.Setx(int value) Line 30	C#
ReferenceType.cs:demo.EntryPoint.Test(demo.ValueType arg) Line 62	C#
ReferenceType.cs:demo.EntryPoint.Main() Line 56	C#

The screenshot shows the Visual Studio IDE with the `ReferenceType.cs` file open. The `Test(ValueType arg)` method in the `demo.EntryPoint` class is selected. The code in the editor is as follows:

```
53  
54 {  
55     ValueType obj1 = new ValueType();  
56     obj1.Setx(100);  
57     Test(obj1);  
58     ReferenceType obj2 = new ReferenceType();  
59     NewTest(obj2);  
60 }  
61 1 reference  
62 static void Test(ValueType arg)  
63 {  
64     arg.Setx(300);  
65 }  
66 1 reference  
67 static void NewTest(ReferenceType arg)  
68 {  
69     arg.Setx(500);  
70 }  
71 }
```

The `Autos` window shows the following variables:

Name	Value	Type
arg	(demo.ValueType)	demo.Va
x	300	int

The `Call Stack` window shows the following frames:

Name	Lang
ReferenceType.cs:demo.EntryPoint.Test(demo.ValueType arg) Line 62	C#
ReferenceType.cs:demo.EntryPoint.Main() Line 56	C#

Step 10: Now control comes to Entrypoint.Main().

The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays the `ReferenceType.cs` file. The execution is paused at line 57: `ReferenceType obj2 = new ReferenceType();`. The code includes:

```
53 {  
54     ValueType obj1 = new ValueType();  
55     obj1.Setx(100);  
56     Test(obj1);  
57     ReferenceType obj2 = new ReferenceType();  
58     NewTest(obj2);  
59 }  
60 static void Test(ValueType arg)  
61 {  
62     arg.Setx(300);  
63 }  
64 static void NewTest(ReferenceType arg)  
65 {  
66     arg.Setx(500);  
67 }  
68 }  
69 }  
70 }  
71 }
```


Autos Window:

Name	Value	Type
obj1	{demo.ValueType}	demo.ValueType
x	100	int
obj2	null	demo.ReferenceType

Call Stack:

Name	Lang
ReferenceType.cs.exe\demo.EntryPoint.Main() Line 57	C#

Step 11 : obj1 memory is cleared. New object obj2 is created and the value of x is initialised to 0 by default.

The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays the `ReferenceType.cs` file. The execution is paused at line 58: `NewTest(obj2);`. The code is the same as in Step 10.
- Autos Window:**

Name	Value	Type
obj2	{demo.ReferenceType}	demo.ReferenceType
x	0	int
Static members		
- Call Stack:**

Name	Lang
ReferenceType.cs.exe\demo.EntryPoint.Main() Line 58	C#

Step 12 : The control now comes to EntryPoint.NewTest().The arguments sent is obj2.

The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays the `ReferenceType.cs` file. The `NewTest` method is highlighted, showing it takes a `ReferenceType` argument. The code includes a `Test` method and a `NewTest` method that calls `Test` and then `NewTest` again.
- Autos Window:** Shows the current state of variables. The `arg` variable is of type `demo.ReferenceType` and has a value of `{demo.ReferenceType}`. The `x` variable is of type `int` and has a value of `0`.
- Call Stack:** Shows the sequence of calls. The top call is `ReferenceType.cs:exe\demo.EntryPoint.NewTest(demo.ReferenceType arg) Line 65`. Below it is `ReferenceType.cs:exe\demo.EntryPoint.Main() Line 58`.

Step 13: Calls Setx method and sends obj2 and value=500 as parameter.

The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays the `ReferenceType.cs` file. The `NewTest` method is highlighted, showing it calls `arg.Setx(500)`. The code includes a `Test` method and a `NewTest` method that calls `Test` and then `NewTest` again.
- Autos Window:** Shows the current state of variables. The `arg` variable is of type `demo.ReferenceType` and has a value of `{demo.ReferenceType}`. The `x` variable is of type `int` and has a value of `0`.
- Call Stack:** Shows the sequence of calls. The top call is `ReferenceType.cs:exe\demo.EntryPoint.NewTest(demo.ReferenceType arg) Line 66`. Below it is `ReferenceType.cs:exe\demo.EntryPoint.Main() Line 58`.

Step 14 : The value in “value” is assigned to x.

The screenshot shows the Visual Studio IDE with the `ReferenceType.cs` file open. The code defines a `demo.ReferenceType` struct with a `Setx(int value)` method. The execution is paused at line 21, where `this.x = value;` is being executed. The `Autos` window shows the following variables:

Name	Value	Type
this	{demo.ReferenceType}	demo.Re
x	500	int
Static members		
this.x	500	int
value	500	int

The `Call Stack` window shows the following calls:

- `ReferenceType.cs:demo.ReferenceType.Setx(int value) Line 21` (C#)
- `ReferenceType.cs:demo.EntryPoint.NewTest(demo.ReferenceType arg) Line 66` (C#)
- `ReferenceType.cs:demo.EntryPoint.Main() Line 58` (C#)

Step 15: Control comes back to Entrypoint.NewTest().

The screenshot shows the Visual Studio IDE with the `ReferenceType.cs` file open. The code defines a `demo.ReferenceType` struct with a `NewTest` method. The execution is paused at line 67, where `arg.Setx(500);` is being executed. The `Autos` window shows the following variables:

Name	Value	Type
arg	{demo.ReferenceType}	demo.Re
x	500	int
Static members		

The `Call Stack` window shows the following calls:

- `ReferenceType.cs:demo.EntryPoint.NewTest(demo.ReferenceType arg) Line 67` (C#)
- `ReferenceType.cs:demo.EntryPoint.Main() Line 58` (C#)

Step 16: Control jumps to EntryPoint.Main() completes the program execution.

The screenshot shows the Visual Studio IDE with the following components:

- Code Editor:** Displays the code in `ReferenceType.cs`. The execution flow is shown jumping to the `Main()` method in `demo.EntryPoint`. The code includes a `NewTest(obj2);` call, a `Test(ValueType arg)` method, and a `NewTest(ReferenceType arg)` method.
- Autos Window:** Shows the state of variables:

Name	Value	Type
obj2	{demo.ReferenceType}	demo.Re
x	500	int
Static members		
- Call Stack Window:** Shows the current call stack entry:

Name	Lang
Referencetype.cs.exe\demo.EntryPoint.Main() Line 59	C#