

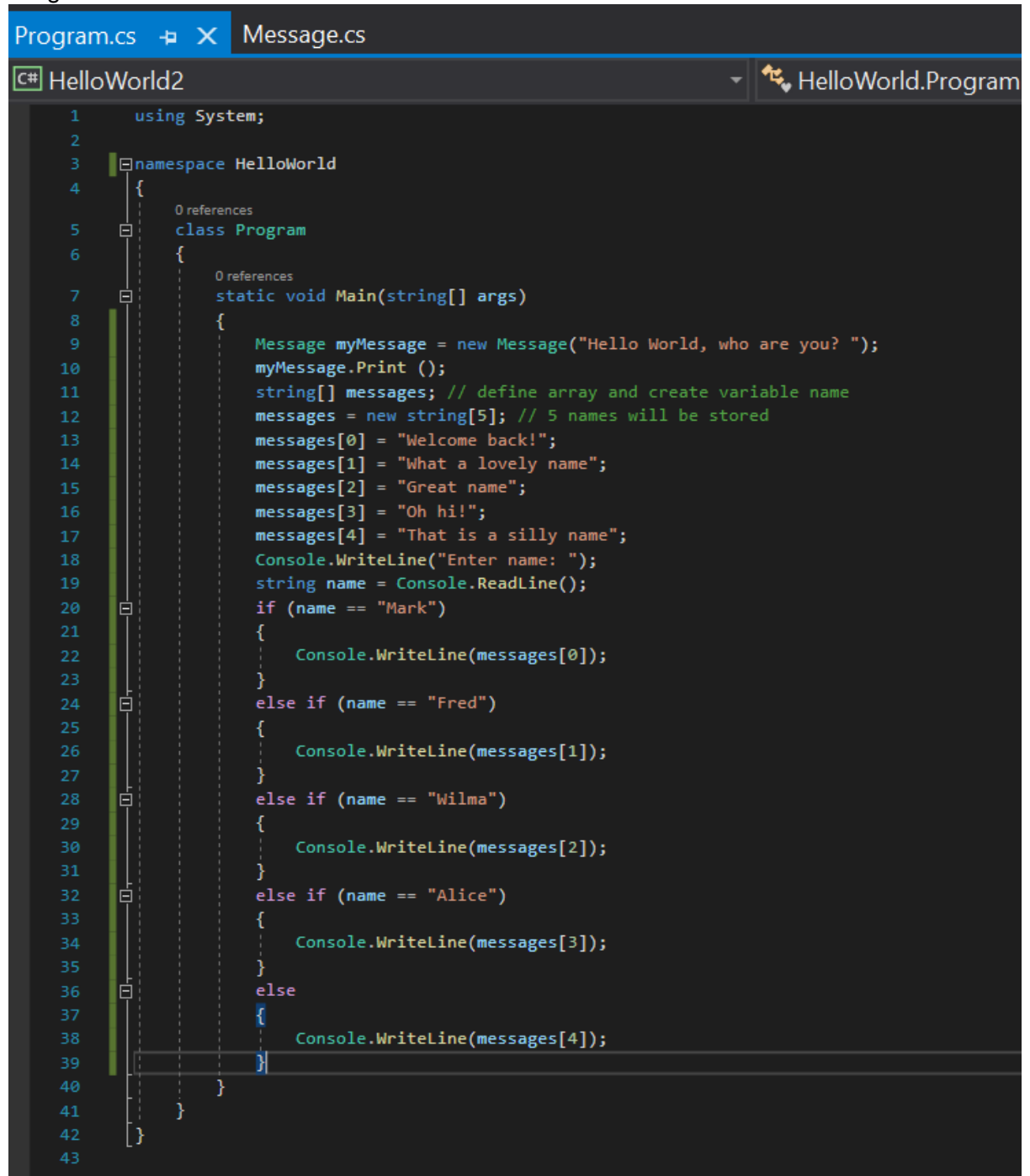
# 1.2

## Object Oriented Hello World

Thanh Minh

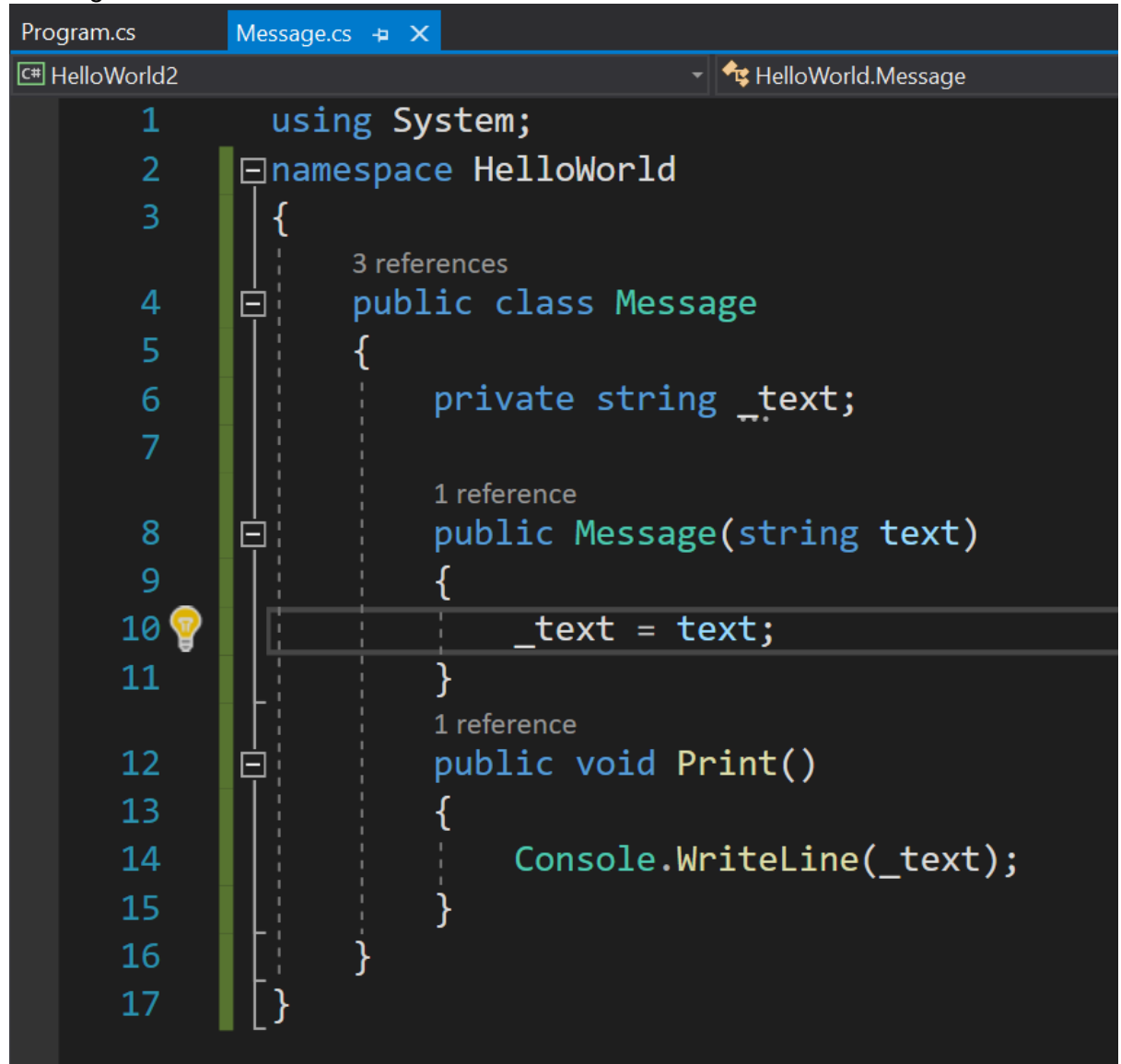
### 1. Evidence of the code

Program.cs



```
1  using System;
2
3  namespace HelloWorld
4  {
5      0 references
6      class Program
7      {
8          0 references
9          static void Main(string[] args)
10         {
11             Message myMessage = new Message("Hello World, who are you? ");
12             myMessage.Print ();
13             string[] messages; // define array and create variable name
14             messages = new string[5]; // 5 names will be stored
15             messages[0] = "Welcome back!";
16             messages[1] = "What a lovely name";
17             messages[2] = "Great name";
18             messages[3] = "Oh hi!";
19             messages[4] = "That is a silly name";
20             Console.WriteLine("Enter name: ");
21             string name = Console.ReadLine();
22             if (name == "Mark")
23             {
24                 Console.WriteLine(messages[0]);
25             }
26             else if (name == "Fred")
27             {
28                 Console.WriteLine(messages[1]);
29             }
30             else if (name == "Wilma")
31             {
32                 Console.WriteLine(messages[2]);
33             }
34             else if (name == "Alice")
35             {
36                 Console.WriteLine(messages[3]);
37             }
38             else
39             {
40                 Console.WriteLine(messages[4]);
41             }
42         }
43     }
```

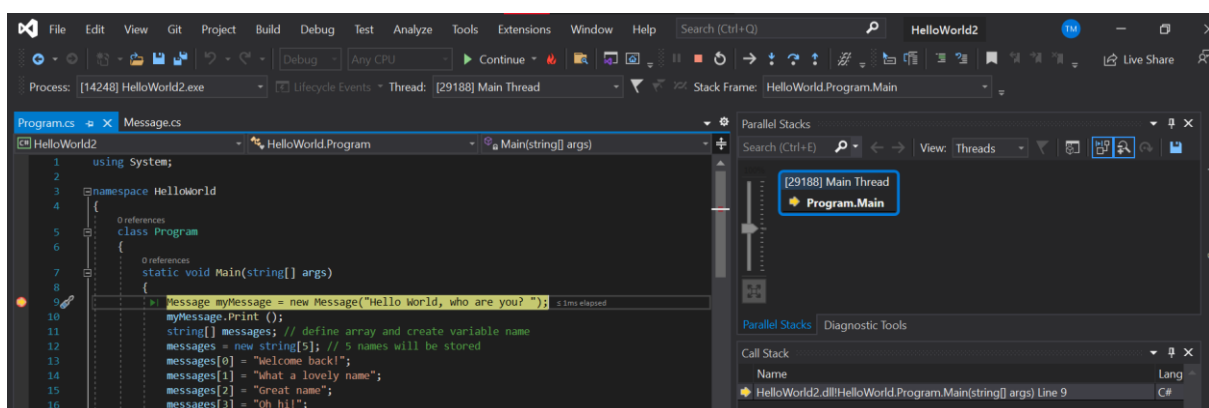
## Message.cs

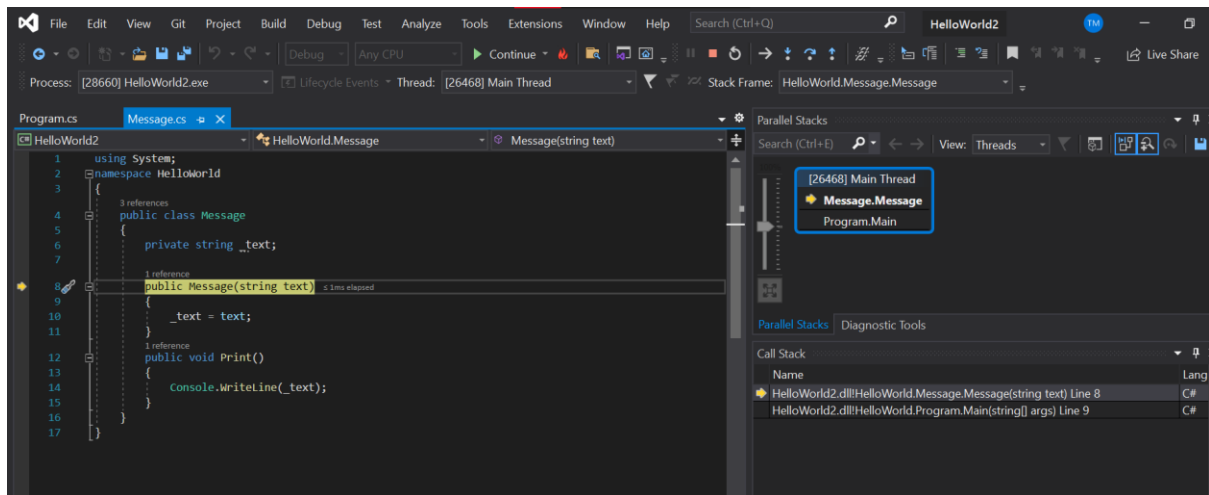


```
1  using System;
2  namespace HelloWorld
3  {
4      3 references
5      public class Message
6      {
7          private string _text;
8
9          1 reference
10         public Message(string text)
11         {
12             _text = text;
13         }
14
15         1 reference
16         public void Print()
17         {
18             Console.WriteLine(_text);
19         }
20     }
21 }
```

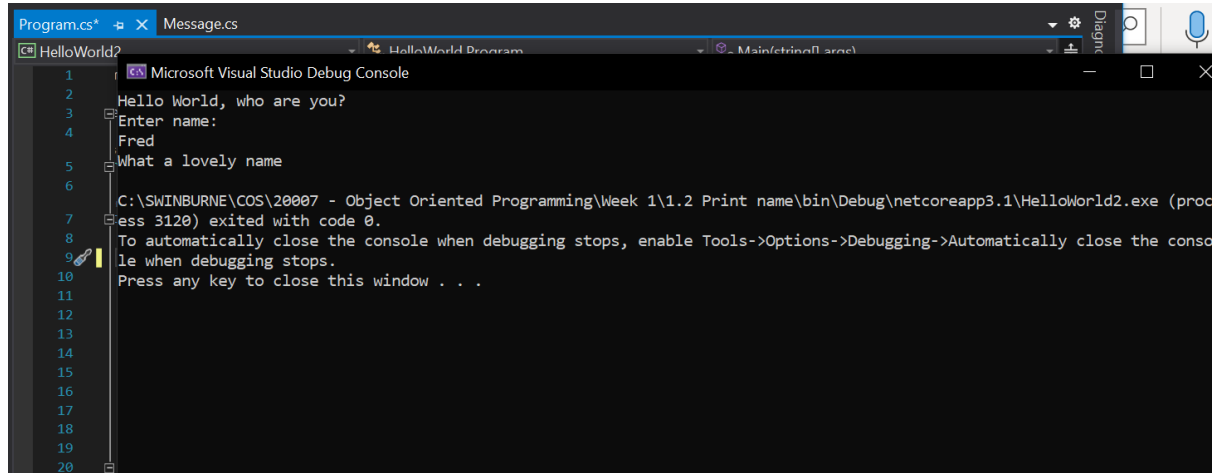
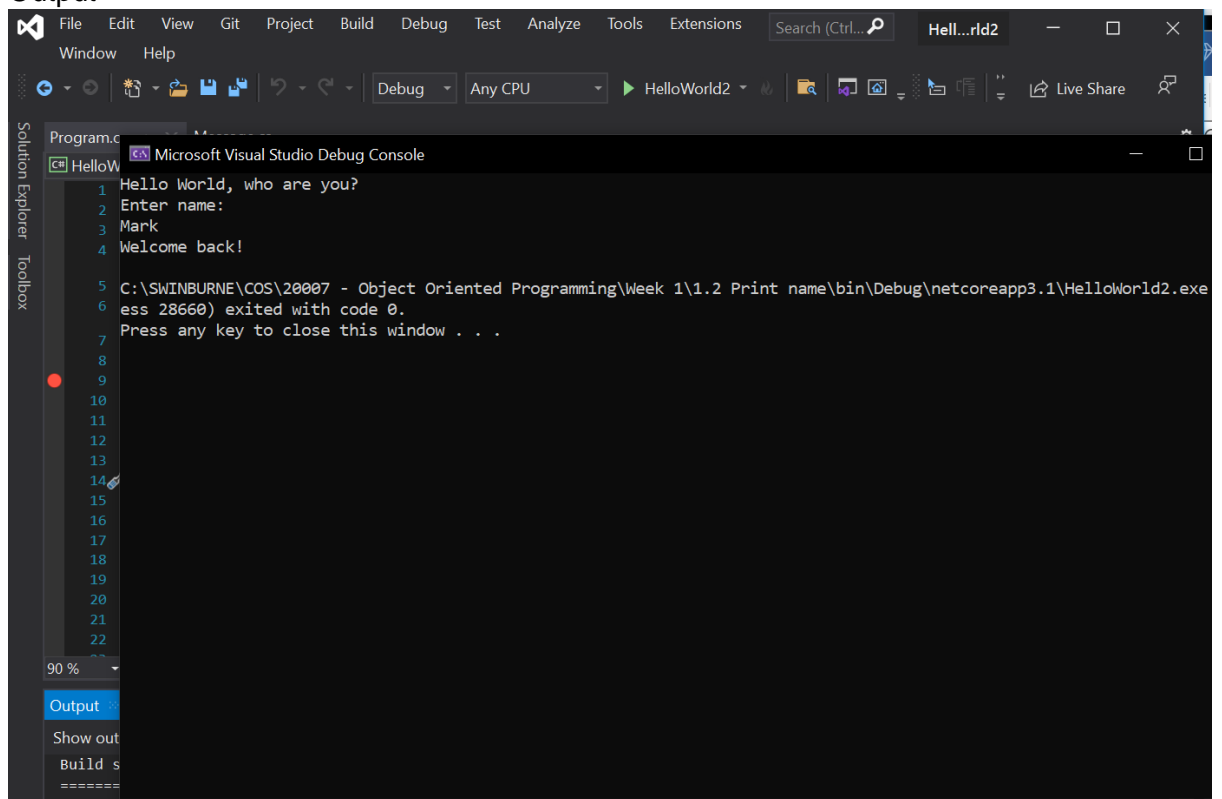
The screenshot shows the Message.cs file in Visual Studio. A call hierarchy is displayed for the Print method, showing references from the Message class and its constructor. A lightbulb icon is visible next to line 10.

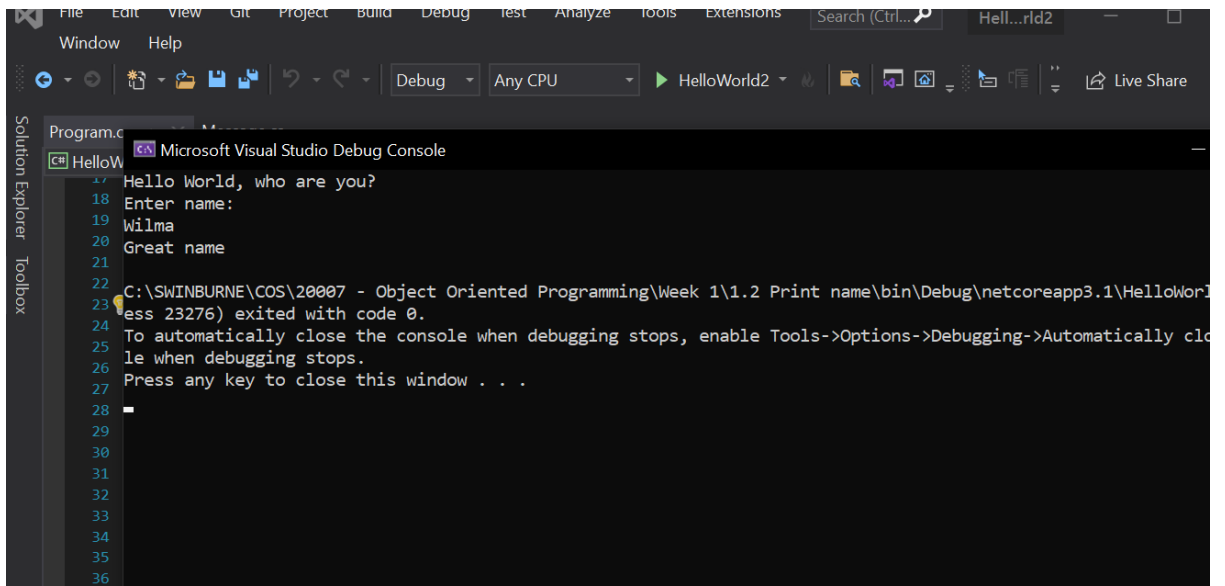
## 2. Evidence of break point within the program





### 3. Output





File Edit View Git Project Build Debug Test Analyze Tools Extensions Search (Ctrl...) Hell...rld2

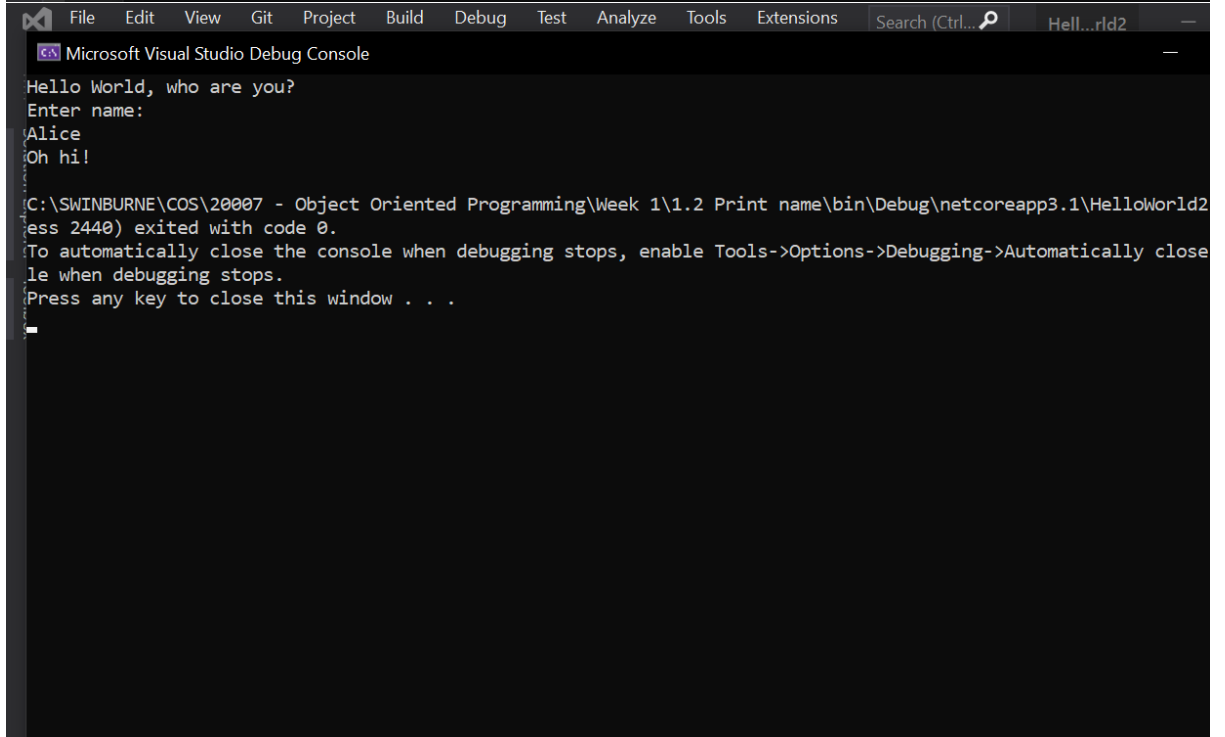
Window Help

Debug Any CPU HelloWorld2 Live Share

Solution Explorer Toolbox

Program.cs

```
17 Hello World, who are you?  
18 Enter name:  
19 Wilma  
20 Great name  
21  
22 C:\SWINBURNE\COS\20007 - Object Oriented Programming\Week 1\1.2 Print name\bin\Debug\netcoreapp3.1\HelloWorld2  
23 (Process 23276) exited with code 0.  
24 To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close  
25 console when debugging stops.  
26 Press any key to close this window . . .  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36
```



File Edit View Git Project Build Debug Test Analyze Tools Extensions Search (Ctrl...) Hell...rld2

Microsoft Visual Studio Debug Console

```
Hello World, who are you?  
Enter name:  
Alice  
Oh hi!  
  
C:\SWINBURNE\COS\20007 - Object Oriented Programming\Week 1\1.2 Print name\bin\Debug\netcoreapp3.1\HelloWorld2  
(Process 2440) exited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close  
console when debugging stops.  
Press any key to close this window . . .  
  

```

