

Universidad Rafael Landívar
Facultad de ingeniería
Ingeniería en informática y sistemas
Catedrático: Luis Enrique Aguilar Rojas
Santiago Marengo | 1273525

Instrucción selección if



```
Programs X
C# Program1
1 DisplayWeatherReport(15.0); // Output: Cold...
2 DisplayWeatherReport(34.0); // Output: Perfect!
3
4 void DisplayWeatherReport(double tempInCelsius)
5 {
6     if (tempInCelsius < 20.0)
7     {
8         Console.WriteLine("Cold.");
9     }
10    else
11    {
12        Console.WriteLine("Perfect!");
13    }
14 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Restaurando C:\Users\ADMIN\Desktop\trabajo\compu\semana9\clasesemana9\clasesemana9.csproj:
Restauración realizada correctamente.

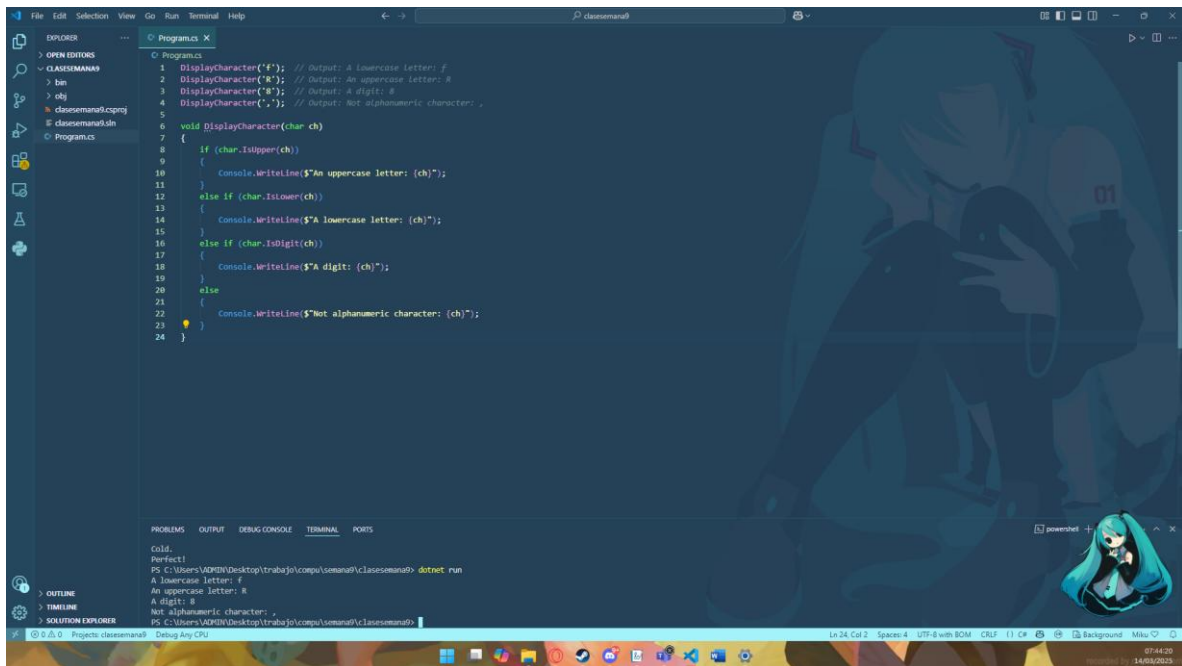
PS C:\Users\ADMIN\Desktop\trabajo\compu\semana9\clasesemana9> dotnet run
Cold.
Perfect!

PS C:\Users\ADMIN\Desktop\trabajo\compu\semana9\clasesemana9>

Debug Any CPU

Ln 14, Col 2 Spaces 4 UTF-8 with BOM CRLF (1) C# Background Miku

Else if



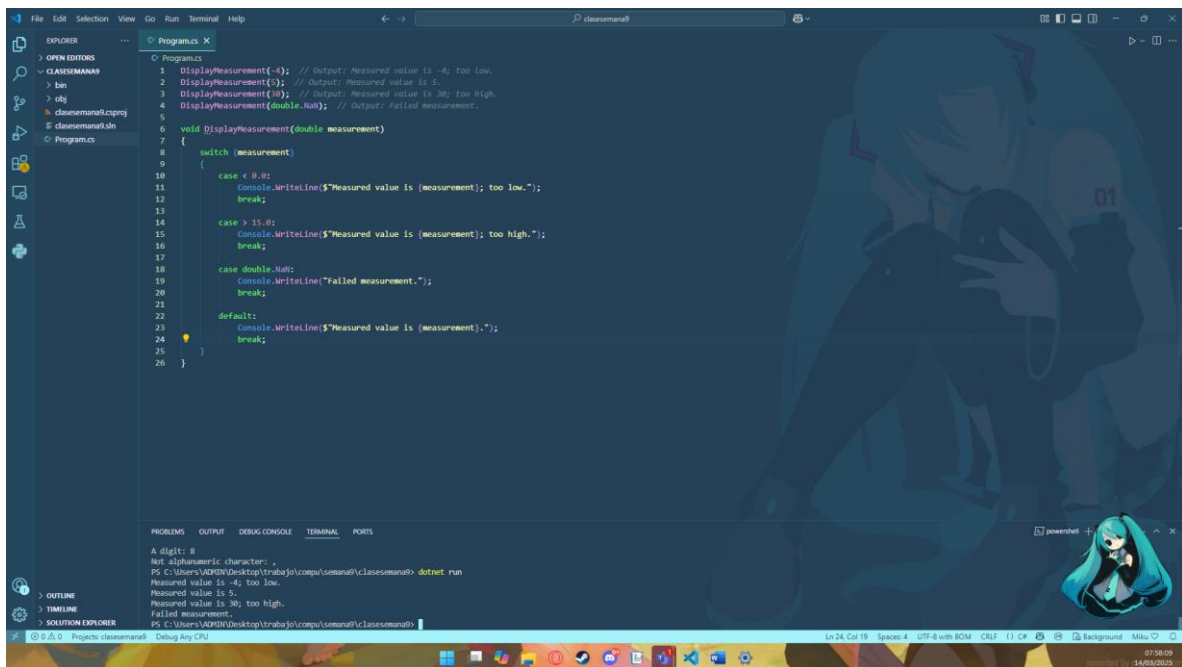
The screenshot shows the Visual Studio Code editor with a C# file named `Program.cs`. The code defines a `DisplayCharacter` method that takes a `char` and prints its classification based on whether it is an uppercase letter, lowercase letter, digit, or not alphanumeric. The `main` method calls this function with 'f', 'R', '8', and '.'. The output window shows the results: 'Cold. Perfect!', 'A lowercase letter: f', 'An uppercase letter: R', 'A digit: 8', and 'Not alphanumeric character: .'.

```
1 DisplayCharacter("f"); // Output: A lowercase letter: f
2 DisplayCharacter("R"); // Output: An uppercase letter: R
3 DisplayCharacter("8"); // Output: A digit: 8
4 DisplayCharacter("."); // Output: Not alphanumeric character: .
5
6 void DisplayCharacter(char ch)
7 {
8     if (char.IsUpper(ch))
9     {
10         Console.WriteLine($"An uppercase letter: {ch}");
11     }
12     else if (char.IsLower(ch))
13     {
14         Console.WriteLine($"A lowercase letter: {ch}");
15     }
16     else if (char.IsDigit(ch))
17     {
18         Console.WriteLine($"A digit: {ch}");
19     }
20     else
21     {
22         Console.WriteLine($"Not alphanumeric character: {ch}");
23     }
24 }
```

Output:

```
Cold.
Perfect!
PS C:\Users\ADMIN\Desktop\trabajo\comp\semanaf\clasesemanaf> dotnet run
A lowercase letter: f
An uppercase letter: R
A digit: 8
Not alphanumeric character: .
PS C:\Users\ADMIN\Desktop\trabajo\comp\semanaf\clasesemanaf>
```

Switch



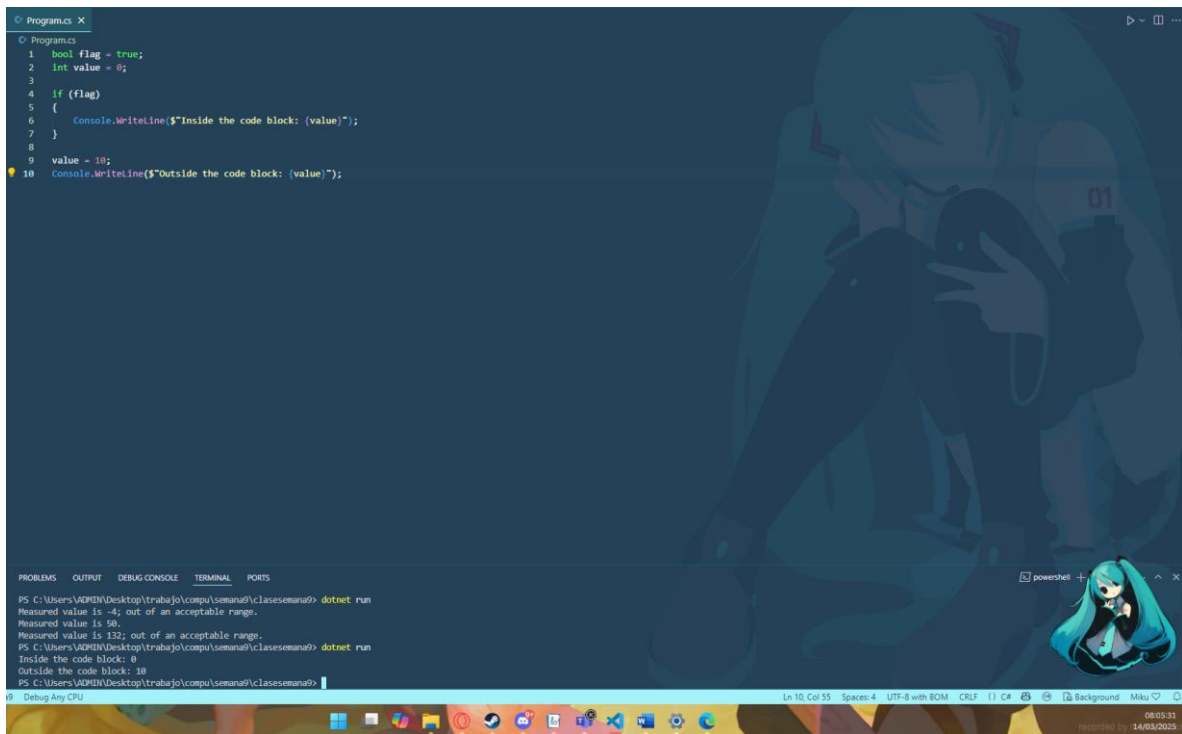
The screenshot shows the Visual Studio Code editor with a C# file named `Program.cs`. The code defines a `DisplayMeasurement` method that takes a `double` and prints a message based on the value of the measurement. The `main` method calls this function with -4, 5, 30, and `double.NaN`. The output window shows the results: 'A digit: 8', 'Not alphanumeric character: .', 'Measured value is -4; too low.', 'Measured value is 5.', 'Measured value is 30; too high.', and 'Failed measurement:'.

```
1 DisplayMeasurement(-4); // Output: Measured value is -4; too low.
2 DisplayMeasurement(5); // Output: Measured value is 5.
3 DisplayMeasurement(30); // Output: Measured value is 30; too high.
4 DisplayMeasurement(double.NaN); // Output: Failed measurement.
5
6 void DisplayMeasurement(double measurement)
7 {
8     switch (measurement)
9     {
10         case < 0.0:
11             Console.WriteLine($"Measured value is {measurement}; too low.");
12             break;
13         case > 15.0:
14             Console.WriteLine($"Measured value is {measurement}; too high.");
15             break;
16         case double.NaN:
17             Console.WriteLine("Failed measurement.");
18             break;
19         default:
20             Console.WriteLine($"Measured value is {measurement}.");
21             break;
22     }
23 }
24
25
26 }
```

Output:

```
A digit: 8
Not alphanumeric character: .
PS C:\Users\ADMIN\Desktop\trabajo\comp\semanaf\clasesemanaf> dotnet run
Measured value is -4; too low.
Measured value is 5.
Measured value is 30; too high.
Failed measurement:
PS C:\Users\ADMIN\Desktop\trabajo\comp\semanaf\clasesemanaf>
```

Ejemplo bool

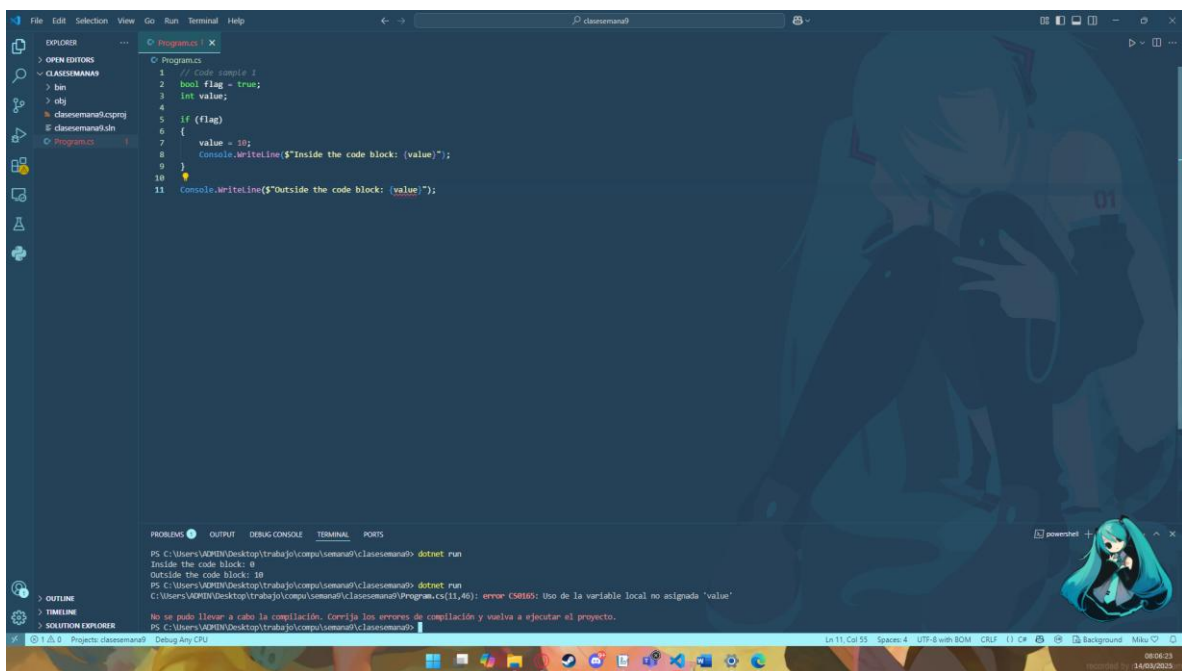


```
1 bool flag = true;
2 int value = 0;
3
4 if (flag)
5 {
6     Console.WriteLine($"Inside the code block: {value}");
7 }
8
9 value = 10;
10 Console.WriteLine($"Outside the code block: {value}");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\ADMIN\Desktop\trabajo\comp\semana9\clasesemana9> dotnet run
Measured value is -4; out of an acceptable range.
Measured value is 50.
Measured value is 132; out of an acceptable range.
PS C:\Users\ADMIN\Desktop\trabajo\comp\semana9\clasesemana9> dotnet run
Inside the code block: 0
Outside the code block: 10
PS C:\Users\ADMIN\Desktop\trabajo\comp\semana9\clasesemana9>

ejemplo de fallo

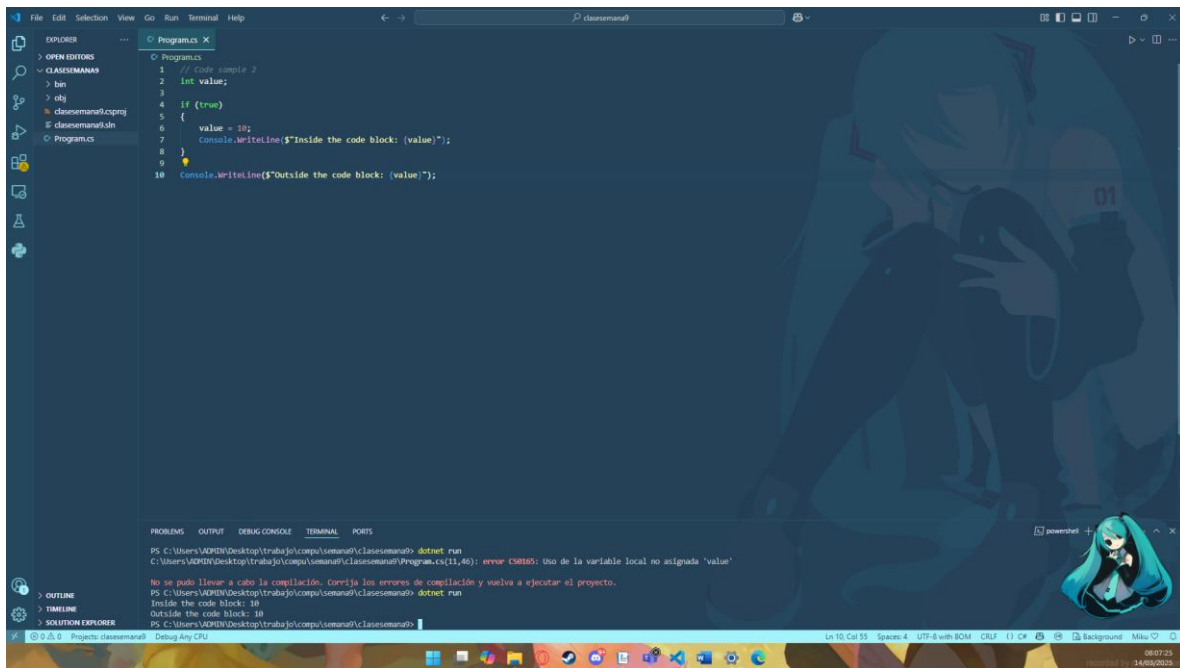


```
1 // Code sample 1
2 bool flag = true;
3 int value;
4
5 if (flag)
6 {
7     value = 10;
8     Console.WriteLine($"Inside the code block: {value}");
9 }
10
11 Console.WriteLine($"Outside the code block: {value}");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\ADMIN\Desktop\trabajo\comp\semana9\clasesemana9> dotnet run
Inside the code block: 0
Outside the code block: 10
PS C:\Users\ADMIN\Desktop\trabajo\comp\semana9\clasesemana9> dotnet run
C:\Users\ADMIN\Desktop\trabajo\comp\semana9\clasesemana9\Program.cs(11,40): error CS0165: Use of the locally declared variable 'value' without assignment.
No se pudo llevar a cabo la compilación. Corrija los errores de compilación y vuelva a ejecutar el proyecto.
PS C:\Users\ADMIN\Desktop\trabajo\comp\semana9\clasesemana9>

Ejemplo de corrección



Comprobación de conocimientos

1. Un desarrollador escribe código que incluye un bloque de código de la instrucción `if`. Inicializan una variable de entero en un valor de 5 por encima (fuera) del bloque de código. Inicializan una segunda variable de entero en un valor de 6 en la primera línea dentro del bloque de código. La expresión booleana del bloque de código de bloque se evalúa como `true` si la primera variable de entero tiene un valor superior a 0. En la segunda línea del bloque de código, asignan la suma de los dos valores a la primera variable. En la primera línea después del bloque de código, escriben código para mostrar el valor del primer entero. ¿Cuál es el resultado cuando se ejecuta la instrucción de código utilizada para mostrar el primer entero? *



No se genera ningún error y se muestra el valor entero. El valor que se muestra es la suma del primer y segundo enteros.

✓ Correcto. Puesto que el primer entero se inicializa encima del código de instrucción `if`, sigue en el ámbito después del bloque de código. Además, dado que ambos enteros están en el ámbito y se inicializan con valores dentro del bloque de código, la adición de los valores se ejecuta correctamente. Por último, aunque el segundo entero no exista fuera del bloque de código, el primer entero conserva los cambios de su valor que se produjeron dentro del bloque de código.



No se genera ningún error y se muestra el valor entero. El valor mostrado es el valor inicializado que figura encima del bloque de código.



Se genera un error porque la primera variable no se encuentra en el ámbito después del bloque de código.

Ejercicio: Eliminación de bloques de código en instrucciones "if"

✓ 100 XP

6 minutos