

- 1- The output shows a dollar sign because the :C format specifier converts the number into a currency format. The currency symbol depends on my computer's regional settings, so in my case it shows \$. Currency formatting also adds two decimal places by default, which is why the result becomes 30.00. I can also control the number of decimal places by writing something like C1 or C2.
- 2- The benefit of the C specifier is that it automatically formats numbers as currency. It also uses the correct currency symbol and decimal format based on the computer's regional settings.
- 3- I used the N specifier; it formats the number with thousand separators and decimal places. The B specifier; it is used to format integral values in binary.

The screenshot displays the Visual Studio IDE with a C# program named 'Task1'. The code defines two integers, X and Y, and uses Console.WriteLine to output the sum of X and Y using three different format specifiers: 'C' (currency), 'N' (number with thousand separators), and 'B' (binary). The output window shows the results: '\$30.00', '30.00', and '11110'. The Solution Explorer on the right shows the project structure, and the Output window at the bottom shows the build process.

```
31 int X = 10;
32 int Y = 20;
33 Console.WriteLine(value: $"Equation: {X} + {Y} = {X + Y:C}");
34 Console.WriteLine(value: $"Equation: {X} + {Y} = {X + Y:N}");
35 Console.WriteLine(value: $"Equation: {X} + {Y} = {X + Y:B}");
36
37
38
39
40
```

Microsoft Visual Studio Debug Console

```
Equation: 10 + 20 = $30.00
Equation: 10 + 20 = 30.00
Equation: 10 + 20 = 11110

C:\Users\Mat\source\repos\Task1\Task1\bin\Debug\net9.0\Task1.exe (process 13680) exited with code 0 (0x0).
Press any key to close this window . . .
```

Output

```
Show output from: Build
Build started at 11:1
1>----- Build starte
1>Skipping analyzers
1>Task1 -> C:\Users\W
----- Build: 1 s
----- Build comp
```

Build succeeded