

(Standard numeric format)

```
int X = 10;
```

```
int Y = 20;
```

```
Console.WriteLine($"Equation: {X} + {Y} = {X + Y:C}");
```

in one (1) pdf page:

1. why the output of this Equation = \$30.00?

2. what is its benefit?

3. try another example with a different specifier with a screenshot of the output.

١. لماذا خرجت النتيجة Equation = \$30.00 ؟:

```
Console.WriteLine($"Equation: {X} + {Y} = {X + Y:C}");
```

- المتغيران X و Y يحتويان على القيم 10 و 20.
- عملية الجمع $X + Y$ تعطي الناتج 30.
- الجزء $\{X + Y:C\}$ يستخدم محدد التنسيق C، والذي يعني عرض الرقم كعملة (Currency).
- بالتالي، يتم عرض الناتج بصيغة العملة الافتراضية حسب إعدادات النظام، وغالبًا ما تكون الدولار الأمريكي (en-US)، فيظهر الناتج على الشكل:

Equation: 10 + 20 = \$30.00

٢. ما فائدة استخدام هذا النوع من التنسيقات؟

استخدام محددات التنسيق (مثل N..., P, C: في لغة C# له فوائد عديدة:

- عرض الأرقام بطريقة احترافية وواضحة، سواء كعملة أو نسبة مئوية أو رقم عشري.
- يوفر الوقت والجهد، بدلًا من كتابة دوال خاصة لمعالجة التنسيق.
- يتوافق مع إعدادات اللغة والثقافة في النظام، مما يجعل البرنامج أكثر مرونة للمستخدمين حول العالم.
- يسهل على المستخدم قراءة وفهم القيم مباشرة دون الحاجة لتفسيرها يدويًا.

٣. مثال آخر باستخدام محدد تنسيق مختلف:

```
int A = 15;
```

```
int B = 5;
```

```
Console.WriteLine($"Equation: {A} / {B} = {(double)A / B:P}");
```

- العملية $A / B = 3.0$
- تم استخدام **P**: وهو محدد تنسيق النسبة المئوية. (Percentage)
- **C#** تقوم تلقائيًا بتحويل الرقم 3.0 إلى نسبة مئوية بضربه في 100، فتكون النتيجة:

Equation: 15 / 5 = 300.00%