





Academy .NET

# C# e Sintassi di base

# Sommario

- Introduzione
- Utilizzare Visual Studio
- Dichiarare le variabili
- Leggere e Scrivere dalla Console
- Convertire input dalla Console
- Stampare nella Console
- Usare i Placeholders

# Introduzione

C # è un linguaggio di programmazione moderno, flessibile e generico orientato agli oggetti per natura, tipizzato e compilato.

- Funziona su .NET Framework / .NET Core.

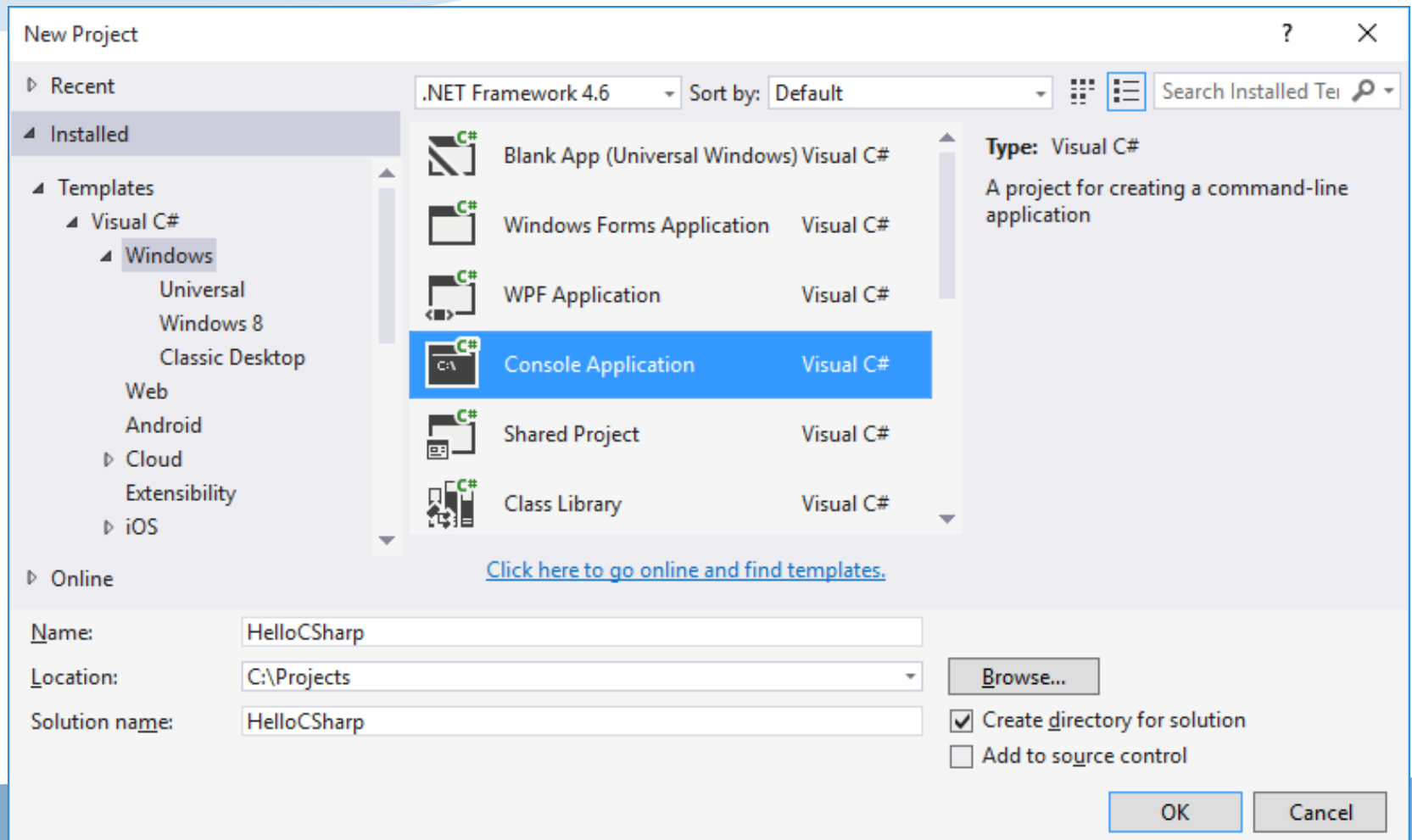
```
static void Main()
{
    Console.Write("Come ti chiami?");
    var nome = Console.ReadLine();
    Console.WriteLine($"Ciao {nome}!");
}
```

# Utilizzare Visual Studio

**Visual Studio** (VS) è un potente IDE per C # e altri linguaggi

- A seguire creazione di un'applicazione Console.

# Utilizzare Visual Studio

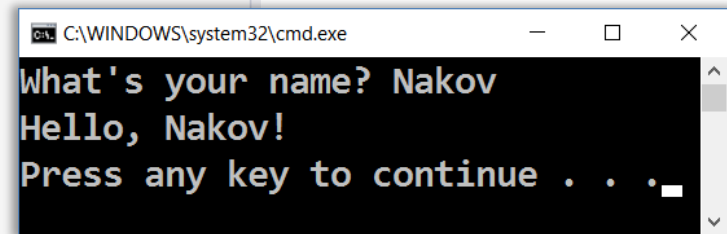
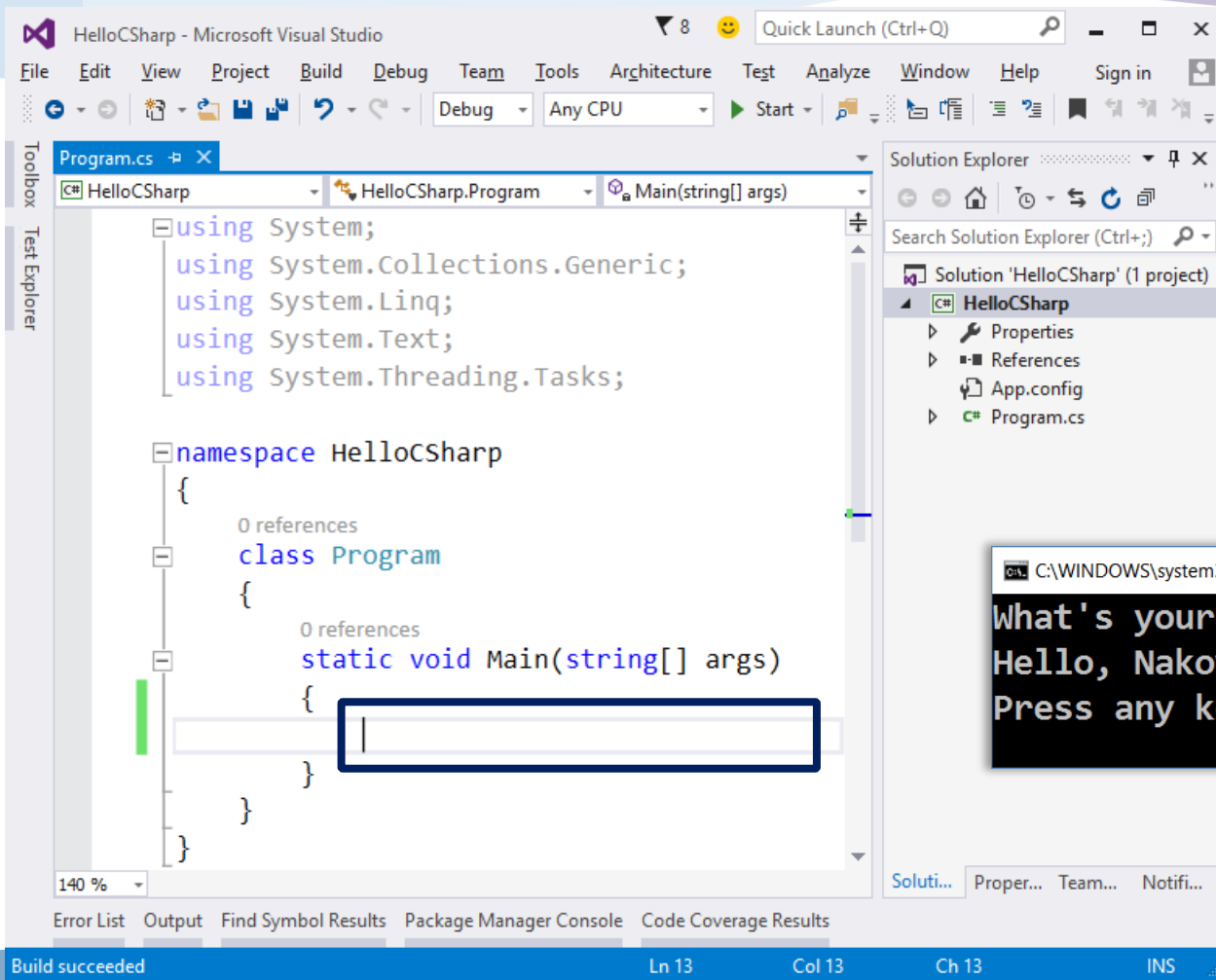


# Utilizzare Visual Studio

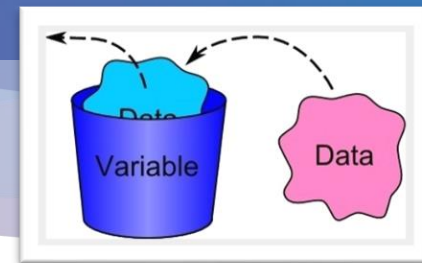
- Scrivere il codice ed eseguire il programma
- É possibile eseguire il programma con **CTRL + F5**



# Utilizzare Visual Studio



# Dichiarare le variabili



Per dichiarare le variabili in C # è necessario utilizzare il pattern:

```
{tipo di dato/ var} {nome variabile} = {valore};
```

- Esempi:

```
var name = "Pesho";  
var isPassed = false;  
var gender = 'F';  
var mathGrade = 5.49;
```

```
int firstNumber = 5;  
string name = "Pesho";  
bool isPassed = false;  
char gender = 'F';
```

Il tipo viene dedotto dal lato destro dell'espressione (usando var)

# Leggere e Scrivere dalla Console



É possibile leggere / scrivere sulla console utilizzando la classe Console:

- Utilizzare lo spazio dei nomi System per accedere alla classe System.Console

```
using System;
```

- Lettura dell'input dalla console utilizzando Console.ReadLine():

```
var nome = Console.ReadLine();
```

return **string**

# Convertire input dalla Console

- **Console.ReadLine()** restituisce una stringa
- Conversione di una stringa in numero con il **parsing**:

```
var nome = Console.ReadLine();  
var età = int.Parse(Console.ReadLine());  
var salario = double.Parse(Console.ReadLine());
```

# Stampare nella Console

É possibile stampare sulla console, usando la classe **Console**

- Utilizzare lo spazio dei nomi **System** per accedere a **System.Console**.
- Scrittura dell'output sulla console utilizzando `Console.WriteLine ()`:

```
var nome = "Gosho";  
Console.WriteLine(nome);
```

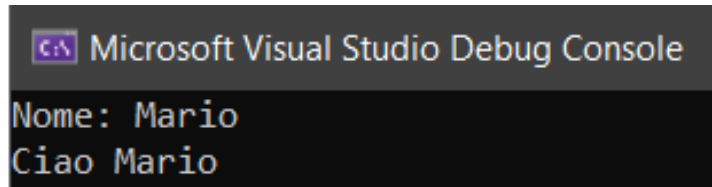
Stampa **Gosho**

# Stampare sulla stessa Riga

Può essere utile stampare testo sulla stessa linea:

- Usa **Console.Write()**:

```
Console.Write("Nome: ");  
var nome = Console.ReadLine();  
Console.WriteLine(" Ciao " + nome);
```



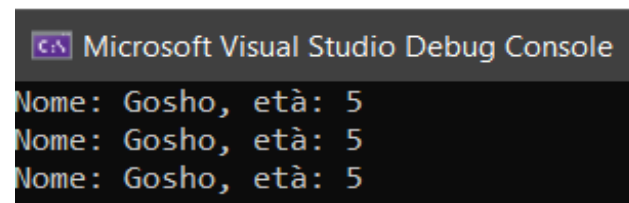
Microsoft Visual Studio Debug Console

```
Nome: Mario  
Ciao Mario
```

# Stampare sulla Console

Può essere utile stampare testo sulla stessa linea:

- Utilizza **Console.Write()**:
- Utilizza la concatenazione di stringhe per stampare testo con numeri
- Oppure i placeholders **{0}**
- Oppure la sintassi **\$ {variabile}**



Microsoft Visual Studio Debug Console

```
Nome: Gosho, età: 5  
Nome: Gosho, età: 5  
Nome: Gosho, età: 5
```

```
var nome = "Gosho";  
var età = 5;  
Console.WriteLine("Nome: " + nome + ", età: " + età);  
Console.WriteLine("Nome: {0}, età: {1}", nome, età);  
Console.WriteLine($"Nome: {nome}, età: {età}");
```

# Problema: Saluti

Scrivere un programma C #, che saluta l'utente per nome:

Mario



Ciao Mario!

Ivan



Ciao Ivan!

Rob



Ciao Rob!



# Problema: Saluti

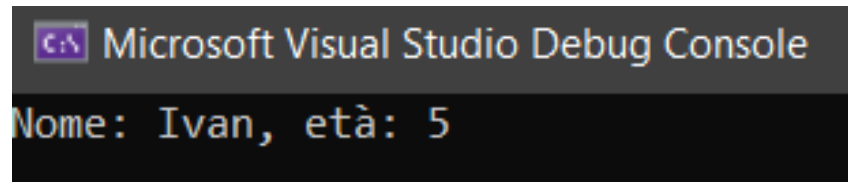
Leggere il nome dalla console e stampalo:

```
using System;

static void Main()
{
    var nome = Console.ReadLine();
    Console.WriteLine(" Ciao, " + nome + "!");
}
```

# Usare i Placeholders

Usare i placeholders per stampare sulla console:



Microsoft Visual Studio Debug Console  
Nome: Ivan, età: 5

```
var nome = "Ivan";  
var età = 5;  
Console.WriteLine("Nome: {0}, età: {1}", nome, età);
```

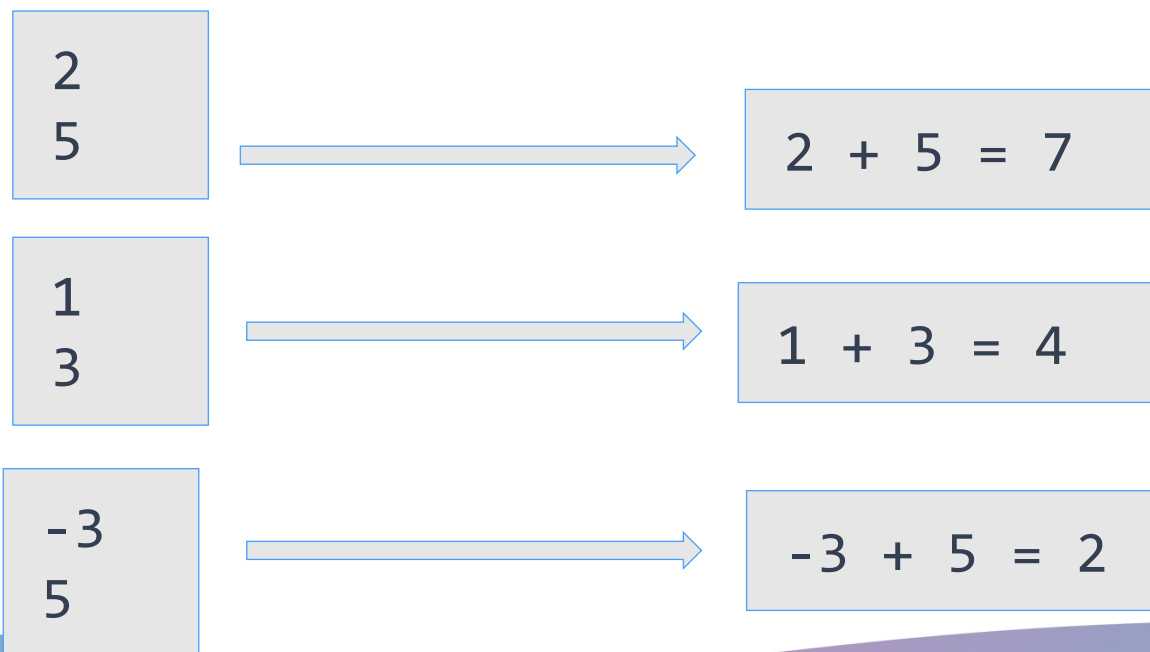
Placeholder {0} corrisponde a **name**

Placeholder {1} corrisponde a **age**

# Problema: Aggiungere due numeri

Scrivere un programma C # per leggere due numeri interi e sommarli.

Stampa la somma come mostrato negli esempi:



# Soluzione: Aggiungere due numeri

Leggere gli interi dalla console, sommarli e stamparli:

```
var a = int.Parse(Console.ReadLine());  
var b = int.Parse(Console.ReadLine());  
var somma = a + b;  
  
Console.WriteLine("{0} + {1} = {2}", a, b, somma);
```

# Interpolazione di stringhe

Usando l'interpolazione di stringhe:

```
var nome = "Ivan";  
var età = 5;  
Console.WriteLine($"Nome: {nome}, età: {età}");
```

Aggiungi \$ di fronte a " " per usare  
l'interpolazione di stringhe

# Formattazione dei numeri in placeholder

- F** - formatta il numero in virgola mobile con alcune cifre dopo il punto decimale
- D** – format number per determinate cifre con zeri iniziali

Esempi:

```
var grado = 5.5334;  
var percentuale = 55;  
Console.WriteLine("{0:F2}", grado); // 5.53  
Console.WriteLine("{0:D3}", percentuale); // 055
```

# Problema: Employee Data

Scrivere un programma C #, che legge le informazioni sui dipendenti e le stampa, formattato come mostrato di seguito:

Ivan  
24  
1192  
1500.353



Nome: Ivan  
età: 24  
Employee ID: 00001192  
Salario: 1500.35

# Soluzione: Employee Data

Leggere i dati dalla console e formattarli:

```
var nome = Console.ReadLine();  
var età = int.Parse(Console.ReadLine());  
var employeeId = int.Parse(Console.ReadLine());  
var salario = double.Parse(Console.ReadLine());  
Console.WriteLine($"Nome: {nome}");  
Console.WriteLine($"età: {età}");  
Console.WriteLine($"Employee ID: {employeeId:D8}");  
Console.WriteLine($"Salario: {salario:F2}");
```





# Domande & approfondimenti



Academy .NET