

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



نسبت سطح بال زنبور به بدنه او بسیار کم
است، با توجه به قوانین آیرودینامیک
پرواز برایش ممکن نیست....

اما زنبور این را نمی داند و پرواز می کند.



C#



آموزش زبان برنامه نویسی
سی شارپ

مدرس : مهندس رزق جو

Mohsen.rezghjoo@modares.ac.ir

فصل اول

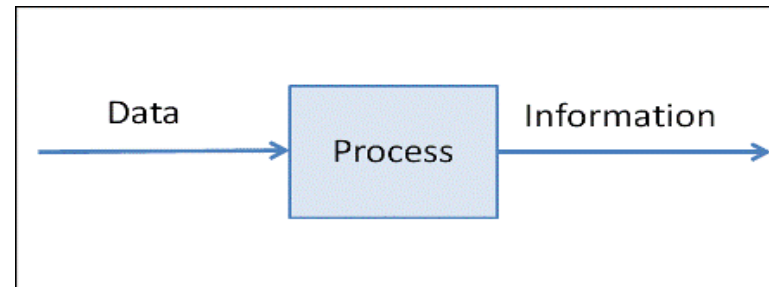
مقدمه



مقدمه



▶ داده ، پردازش و اطلاعات :

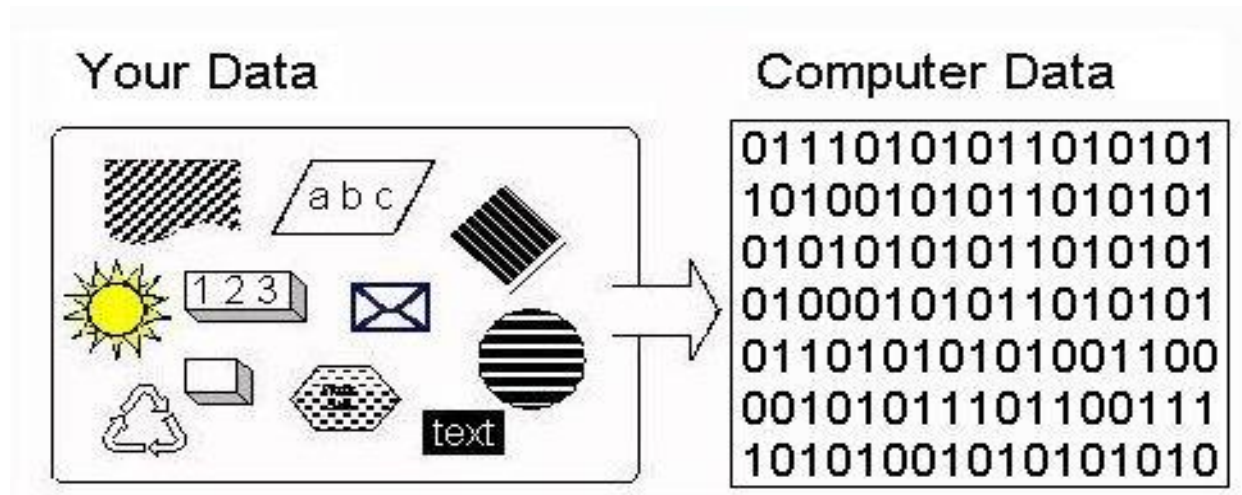


▶ تعریف برنامه (Program) :

به مجموعه دستوراتی که به کامپیوتر می فهماند که چه نوع پردازشی را بر روی داده ها انجام دهد و همچنین اطلاعات به دست آمده را چگونه نمایش دهد را برنامه می گویند.

مقدمه

▶ زبان قابل فهم سخت افزار کامپیوتر، زبان ماشین نام دارد که متشکل از دنباله ای از کد های 0 و 1 است. زبان قابل فهم برای انسان زبان محاوره ای است که در مکالمات روزمره از آن استفاده می کنیم.



▶ **برنامه نویسی :**

نوشتن دستورات لازم برای کنترل نحوه کار کامپیوتر، به طوری که کامپیوتر بتواند یک کار مشخص را انجام دهد را برنامه نویسی می گویند. کسی که به دستورات زبان برنامه نویسی آشنا است و می تواند برنامه نویسی کند را برنامه نویس می گویند.

انواع زبان های برنامه نویسی

▶ زبان سطح پایین :

زبان هایی که به زبان پردازشگر کامپیوتر (Cpu) نزدیک است.
زبان ماشین (Machine language) و زبان اسمبلی (Assembly language) نمونه ای از این زبان ها هستند.

▶ زبان سطح میانی :

این زبان ها به زبان عامیانه ما نزدیک بوده و در آن دستوراتی برای دسترسی راحت تر به سخت افزار پیش بینی شده است.
زبان C نمونه ای از این زبان است.

▶ زبان سطح بالا :

این زبان به زبان محاوره ای ما بسیار نزدیک تر می باشد.
زبان ویژوال بیسیک (VB) ، Java ، و سی شارپ (C#) نمونه ای از این زبان ها می باشد.

در زبان های سطح میانی و سطح بالا فهم دستورات برای ما راحت است ولی فهم آن برای کامپیوتر سخت است بنابراین نیاز به **ترجمه** دستورات هست تا کامپیوتر نیز دستورات را درک کند.

نرم افزار های مترجم

► مفسر (Interpreter) :

نرم افزارهایی که دستورات برنامه نویسی را خط به خط خوانده، تفسیر و اجرا می کنند مفسر نام دارند.

► کامپایلر یا مترجم (Compiler) :

نرم افزارهایی که در ابتدا کل دستورات برنامه را می خوانند و سپس کل آن ها را ترجمه و اجرا می کنند را کامپایلر می گویند.

مراحل ساخت نرم افزار : تحلیل نیازمندی ها ، طراحی ، پیاده سازی (کد نویسی) ، نصب و تست ، نگهداری و پشتیبانی

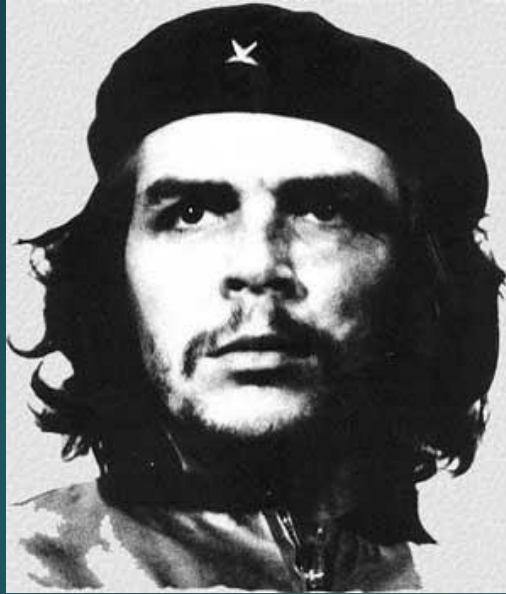


سوال

► موارد ستون سمت چپ را به ستون سمت راست وصل کنید. (یک مورد اضافی است)

- | | |
|--|------------------|
| الف - برنامه ای برای تبدیل کد سطح بالا به کد قابل فهم برای سخت افزار | ۱- برنامه |
| ب - دنباله ای از کدهای صفر و یک | ۲- برنامه نویسی |
| ج - مجموعه ای از دستورات عملی ها برای پردازش داده ها | ۳- مترجم |
| د - نوشتن دستورات لازم برای حل یک مسئله | ۴- زبان ماشین |
| ه - زبان برنامه نویسی نزدیک به زبان محاوره ای | ۵- زبان سطح بالا |
| و- داده ها و ورودی های یک برنامه | |





زندگی می کنم
 حتی اگر بهترین هایم را از دست بدهم ...!
 چون این زندگی کردن است
 که بهترین های دیگر را برایم می سازد
 بگذار هر چه از دست می رود برود ...!
 من آن را می خواهم که به التماس آلوده نباشد،
 حتی "زندگی" را.

"ارنستو چه کوارا - شخصیت اصلی انقلاب کوبا"

فصل دوم

آشنایی با محیط ویژوال استودیو (Visual Studio)



آشنایی با محیط ویژوال استودیو (Visual Studio)

- ▶ **محیط تولید برنامه متمرکز یا IDE (Integrated Development Environment) :**
به محیط برنامه نویسی که در آن می توان تمام مراحل کد نویسی، ترجمه، اشکال یابی و سرانجام اجرا را انجام داد را IDE می گویند.
- ▶ به عنوان مثال شرکت مایکروسافت یک IDE بسیار قوی برای برنامه نویسی فراهم کرده است که ویژوال استودیو نام دارد.
- ▶ ویژوال استودیو یک محیط برنامه نویسی قوی برای تولید برنامه های کاربردی تحت ویندوز است که زبان های برنامه نویسی VB, C++, C# را پشتیبانی می کند.

آشنایی با محیط ویژوال استودیو (Visual Studio)

Express 2012 for Windows Desktop

Start

New Project... ←

Open Project...

Connect to Team Foundation Server...

Recent

ali

New Project

Recent

Installed

Templates

Visual Basic

Visual C# ←

Windows

Test

Visual C++

Visual Studio Solutions

Samples

Online

Sort by: Default

Search Installed Templates (Ctrl+E)

Icon	Template Name	Language
Windows Forms Application	Windows Forms Application	Visual C#
WPF Application	WPF Application	Visual C#
Console Application	Console Application	Visual C# ←
Class Library	Class Library	Visual C#

Type: Visual C#

A project for creating a command-line application

Name: ConsoleApplication1

Location: C:\Users\Ali\Desktop\ Browse...

Solution name: ConsoleApplication1

Create directory for solution

Add to source control

OK Cancel

شکل - نحوه ایجاد یک پروژه جدید

mohsen.rezghjoo@modares.ac.ir

13

آشنایی با محیط ویژوال استودیو (Visual Studio)

- ▶ در بخش Name باید نام پروژه را تایپ کنیم. در این جا نامی مطابق هدف برنامه ای که می خواهیم بنویسیم انتخاب می کنیم.
- ▶ در بخش Location محل ذخیره سازی پروژه تعیین می گردد.
- ▶ در بخش Solution name نامی را برای پوشه که ایجاد می شود معین می کند که فایل های مربوط به یک یا چند پروژه را در آن نگهداری می کند.

Name: ConsoleApplication1

Location: C:\Users\Ali\Desktop\ Browse...

Solution name: ConsoleApplication1

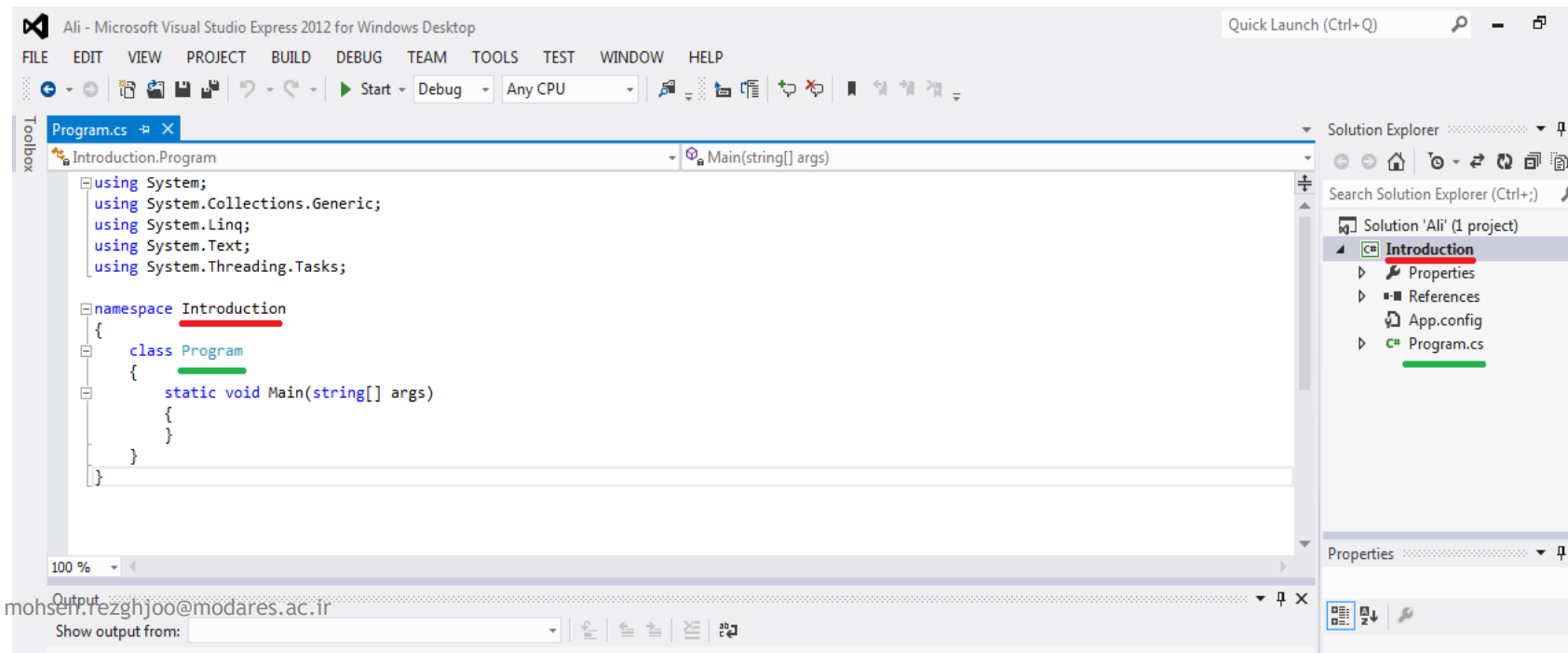
Create directory for solution

Add to source control

OK Cancel

آشنایی با محیط ویژوال استودیو (Visual Studio)

- ▶ به عنوان مثال در قسمت name نام پروژه را Introduction و در قسمت Solution name نام پوشه را Ali می نویسیم. بعد دکمه Ok را کلیک می کنیم. پنجره زیر که محیط ویژوال استودیو است ایجاد می گردد.



آشنایی با محیط ویژوال استودیو (Visual Studio)

► قسمت های مختلف محیط ویژوال استودیو عبارتند از :

نوار منو (Menu bar) و نوار ابزار (Tool Bar) :

منوهای مختلف به همراه ابزارهای پر کاربرد در بالا صفحه در این قسمت قرار دارد.

پنجره ویرایشگر برنامه :

متن برنامه نگهداری می شود یعنی همان جایی که اقدام به کد نویسی می کنیم. در این قسمت می توان چندین برنامه را همزمان به صورت باز نگه داشت.

پنجره لیست خطا (Error List) :

بعد از **ترجمه دستورات** با کلید **F6 یا F7** ، در صورتی که در برنامه اشکال تایپی یا ساختاری داشته باشد در پایین صفحه در این پنجره خطاهایشان لیست می شوند.

پنجره Solution Explorer :

در سمت راست صفحه قرار دارد که ساختار پروژه به همراه تمامی فایل های موجود در آن نشان داده می شود. نام این پنجره را مرورگر پروژه می نامیم چون به وسیله آن به تمام اجزای پروژه دسترسی داریم.

آشنایی با محیط ویژوال استودیو (Visual Studio)

- ▶ اگر هر کدام از قسمت های مختلف محیط ویژوال استودیو باز نمی باشد می توانید از **منوی View** اقدام به اضافه کردن بخش مورد نظر بکنید.
- ▶ در پنجره Solution Explorer ، نام کلاس های مربوط به پروژه به همراه پسوند cs. (مخفف C#) نوشته شده است که **این فایل ها همان فایل های متن برنامه** می باشند که به طور خودکار ایجاد گردیده است. نام این بخش ها را مطابق با عملکرد برنامه در نظر می گیریم.
- ▶ در پنجره ویرایشگر برنامه بعد از دستور namespace (که جزو کلمات کلیدی یا رزرو شده می باشد) نام پروژه نوشته می شود. با استفاده از دستور namespace یک **فضای نامی جدید** برای **سازماندهی و دسته بندی پروژه های بزرگ** تعریف یا ایجاد می کنیم.

آشنایی با محیط ویژوال استودیو (Visual Studio)

- ▶ **ترجمه برنامه :**
- ▶ بعد از نوشتن دستورات برای ترجمه دستورات و اشکال یابی دستورات قبل از اجرا، کلید F6 یا F7 را فشار می دهیم.
- ▶ اگر در هنگام تایپ یک دستور غلط املایی موجود باشد زیر آن **یک خط قرمز رنگ** کشیده می شود که نشان دهنده خطا در تایپ دستورات یا مشکل دستوری است.
- ▶ همچنین اگر دستورات دارای خطایی باشند این خطاها در پنجره Error List به ترتیب لیست می شوند.
- ▶ در پنجره Error List با دابل کلیک بر روی خطاها ، به صورت اتوماتیک به پنجره ویرایشگر برنامه و محل خطای مزبور هدایت می شوید.
- ▶ توصیه می شود اول دستورات برنامه را به صورت کامل بنویسید و بعد در حالت بروز خطا با خواندن توضیح خطا و شماره خط برنامه، خطاها را به **ترتیب از بالا به پایین** تصحیح کنید.
- ▶ اگر بعد از ترجمه دستورات خطایی وجود نداشت، در قسمت پنجره Error List گزینه Succeeded ظاهر می شود.**الان زمان آن است که دستورات خود را اجرا کنیم.**

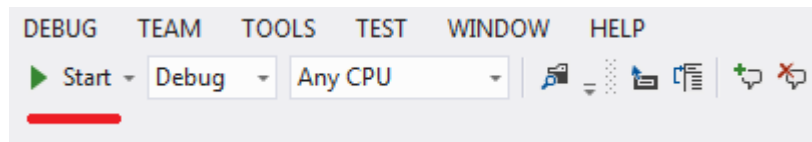
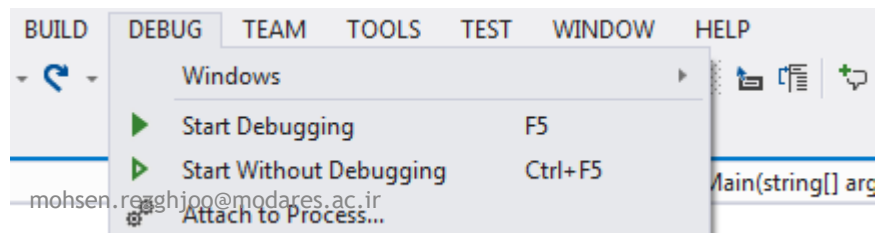
آشنایی با محیط ویژوال استودیو (Visual Studio)

▶ اجرای برنامه :

1- با استفاده از کلید F5 یا (از منوی DEBUG گزینه Start Debugging) یا (گزینه Start موجود در نوار ابزار).

نکته: در این حالت اجرای برنامه، برای ثابت ماندن صفحه کنسول یا خط فرمان (: cmd command prompt) باید دستور Console.ReadKey(); را در آخر دستورات بنویسیم.

2- با استفاده از کلید ترکیبی Ctrl+F5 یا (از منوی DEBUG گزینه Start Without Debugging).



آشنایی با محیط ویژوال استودیو (Visual Studio)

▶ درک هوشمند یا IntelliSense :

یکی از امکانات ارزشمند در هنگام نوشتن دستورات در ویژوال استودیو، درک هوشمند (IntelliSense) می باشد که در هنگام تایپ دستورات برنامه با نوشتن چند حرف از نام دستور (نام کلاس، متد، مغتیر و) منو یا لیستی از دستورات مرتبط با آن چند حرف نمایش داده می شود.

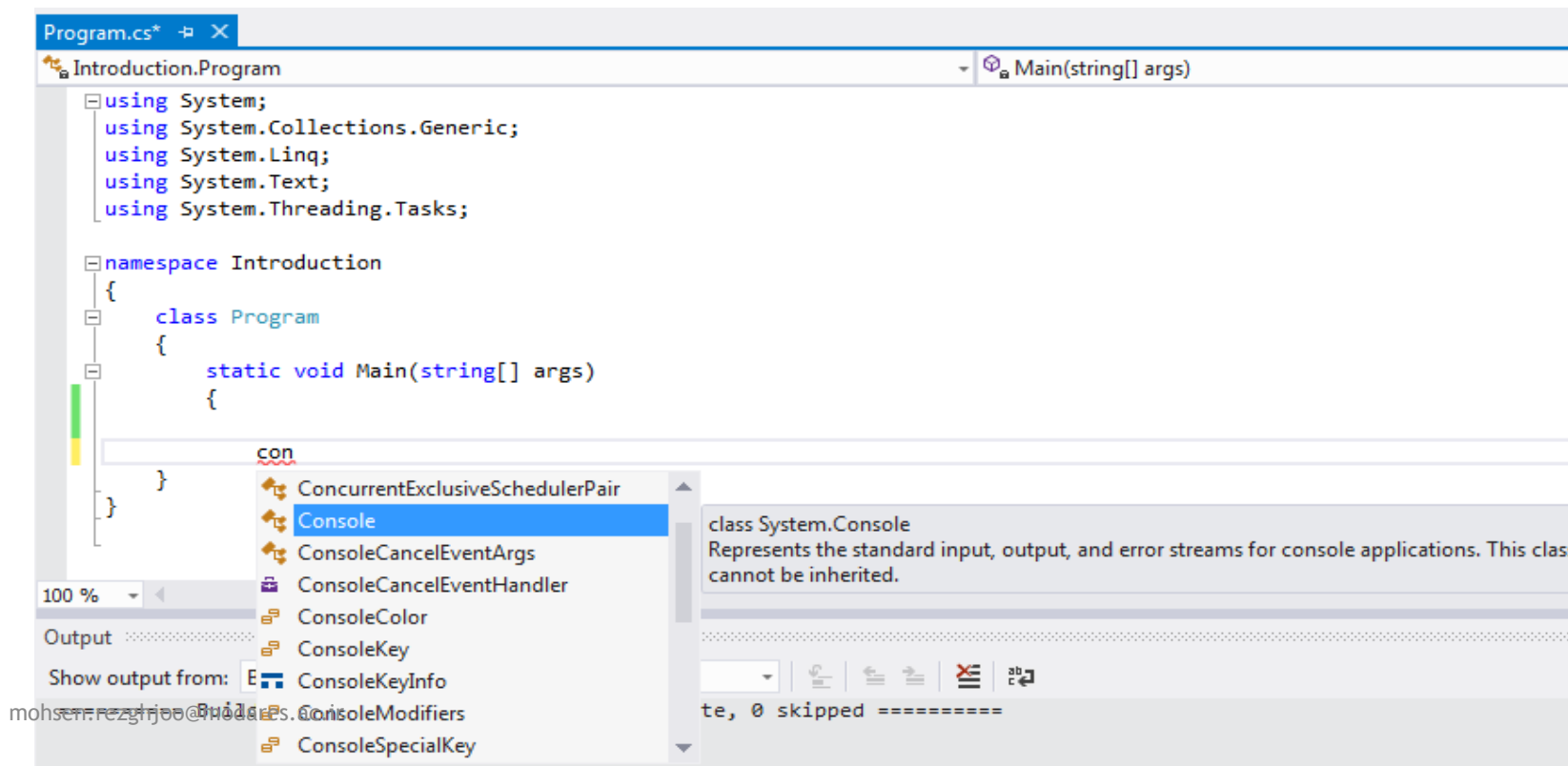
در این لیست ایجاد شده می توان با فلش های بالا و پایین اقدام به حرکت کرد و سپس کلمه مورد نظر را برگزید.

با زدن کلید **Tab** و یا **Enter** دستور مورد نظر به صورت خودکار در بین دستورات برنامه قرار داده می شود و مکان نما موس نیز در ابتدای کلمه قرار می گیرد و دیگر نیازی نیست تمام دستور را تایپ کنید.

در داخل متن برنامه نیز می توان با زدن کلید های ترکیبی **CTRL+SPACE** لیست یا کادر مربوط به IntelliSense را ظاهر کرد و دستورات مورد نظر خود را از آن انتخاب کنیم.

آشنایی با محیط ویژوال استودیو (Visual Studio)

درک هوشمند یا IntelliSense : ►



The screenshot shows the Visual Studio IDE with a C# file named Program.cs. The code defines a class Program with a static Main method. The cursor is positioned at the end of the Main method, and IntelliSense is displaying a list of classes from the System namespace, including Console. The Console class is highlighted, and its documentation is shown in the right-hand pane. The documentation states: "class System.Console Represents the standard input, output, and error streams for console applications. This class cannot be inherited." Below the documentation, there is a small preview of the console output, showing the text "te, 0 skipped =====".

```
Program.cs* [X]
Introduction.Program [Main(string[] args)]
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Introduction
{
    class Program
    {
        static void Main(string[] args)
        {
            con
            }
        }
    }
}
```

ConcurrentExclusiveSchedulerPair
Console
ConsoleCancelEventArgs
ConsoleCancelEventHandler
ConsoleColor
ConsoleKey
ConsoleKeyInfo
ConsoleModifiers
ConsoleSpecialKey

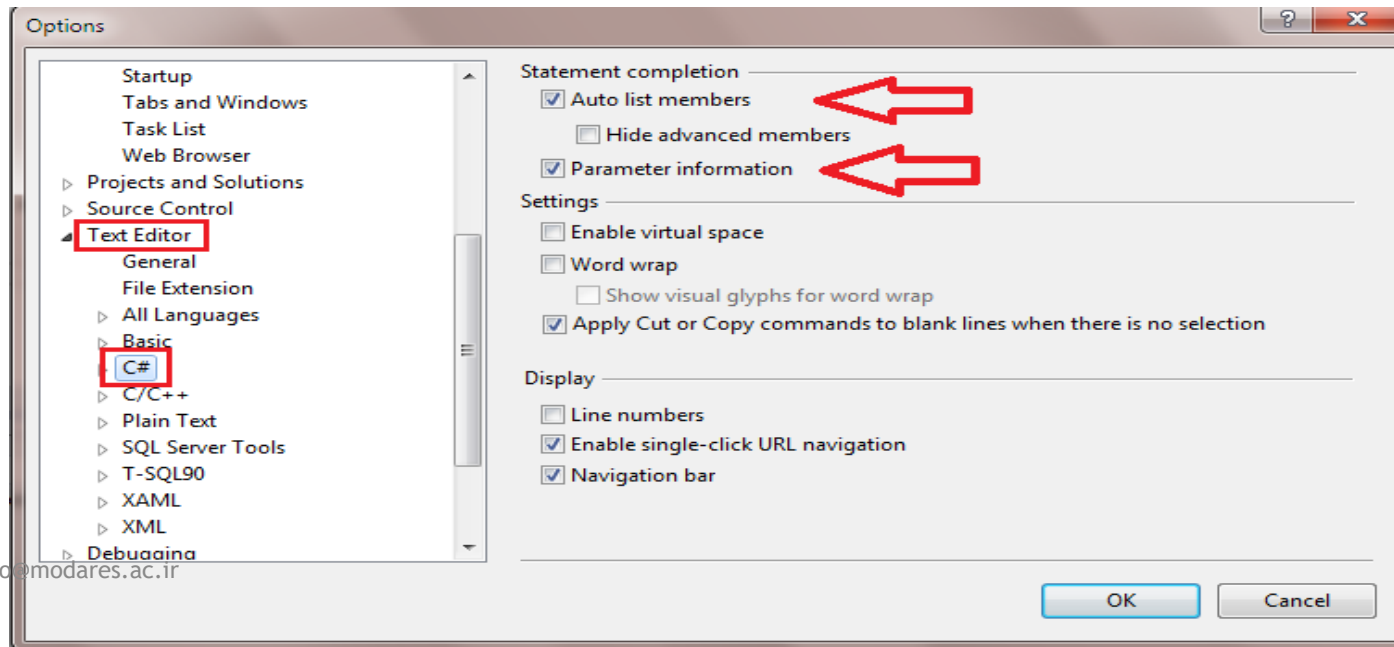
class System.Console
Represents the standard input, output, and error streams for console applications. This class cannot be inherited.

te, 0 skipped =====

آشنایی با محیط ویژوال استودیو (Visual Studio)

▶ درک هوشمند یا IntelliSense :

اگر امکان IntelliSense فعال نیست و از آن نمی توانید استفاده کنید، باید از طریق گزینه Tools موجود در نوار منو، وارد قسمت Options شده و طبق شکل زیر گزینه های Auto list members و Parameter information را فعال کنید.



آمدی جانم به قربانت ولی حالا چرا
بی وفا حالا که من افتادم از پا چرا؟

نوشدارویی و بعد از مرگ سهراب آمدی
سنگدل این زودتر می خواستی حالا چرا؟

عمر ما را مهلت امروز و فردای تو نیست
من که یک امروز مهمان توام فردا چرا؟

نازینا ما به ناز تو جوانی داده ایم
دیگر اکنون با جوانان نازکن با ما چرا؟

وه که با این عمرهای کوتاه بی اعتبار
اینهمه غافل شدن از چون منی شیدا چرا؟

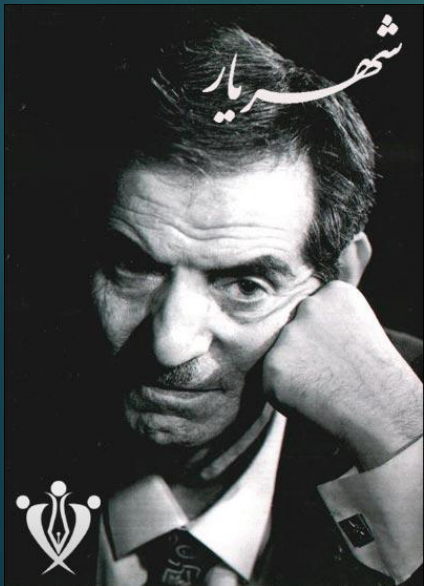
شور فرادم پیرمشن سر به زیر افکنده بود
ای لب شیرین جواب تلخ سر بالا چرا؟

ای شب هجران که یک دم در تو چشم من تفت
اینقدر با بفت خواب آلود من لالا چرا؟

آسمان چون جمع مشتاقان پریشان می کند
در شکفتم من نمی باشد ز هم دنیا چرا؟

در خزان هجر گل ای بلبک طبع حزین
خامشی شرط وفاداری بود غوغا چرا؟

شهریارا بی حیب خود نمی کردی سفر
این سفر راه قیامت میروی تنها چرا؟



فصل سوم

آشنایی با زبان برنامه نویسی سی شارپ (C#)



زبان برنامه نویسی سی شارپ (C#)

- ▶ زبان C# یک زبان برنامه نویسی سطح بالا، شی گرا و همه منظوره است که توسط شرکت مایکروسافت ابداع و توسعه داده شده است.
- ▶ زبان C# شباهت زیادی به زبان های ++C و جاوا دارد، اما برخلاف زبان جاوا که متن باز است این زبان در انحصار شرکت سازنده آن قرار دارد.
- ▶ زبان C# به مانند زبان ++C ، C و جاوا **نسبت به حروف کوچک و بزرگ حساس است** به عنوان مثال یعنی Main و main را متفاوت در نظر می گیرد.
- ▶ **جهت مطالعه:**

(منظور از شی گرایی زبان (Object Oriented Language) یعنی هر تابع یا داده ای به صورت یک شی هست با خصوصیات خودش که میتواند در هر برنامه ای کاربرد داشته باشد و نیازی به نوشتن دوباره آن نیست و میتواند با هر شی دیگری هم ترکیب شده و یک شی واحد را تشکیل دهد. در واقع این رو برای این به کار میبرن که خصوصیات هر شی برای خودش باشد و نتوان آن را بدون اجازه تغییر داد که به این حالت ، کپسوله کردن میگویند تا توابع دیگر نتوانند داده های این شی رو تغییر بدهند.)

زبان برنامه نویسی سی شارپ (C#)

▶ الگوی یک برنامه ساده به زبان C#

یک برنامه کاربردی نوشته شده به زبان C# شامل **مجموعه ای از کلاس ها** است که هر یک از آنها نیز شامل **تعدادی متد** هستند.

```
class یک نام دلخواه
{
    static void Main()
    {
        دستورات مربوط به انجام یک کار
    }
}
```

زبان برنامه نویسی سی شارپ (C#)

▶ کلاس (class) :

- ❖ کلاس به عنوان یک قالب یا الگویی می باشد که در آن داده هایی تعریف می شود که این داده ها مربوط به یک موضوع است. بنابراین در داخل کلاس، عملیات بر روی داده ها و یا الگوریتم انجام کار تعریف می شود.
- ❖ در داخل یک کلاس متد یا دستورات خاصی تعریف شده است.
- ❖ بعضی از کلمات مانند `void` , `static` , `class` کلمات شناخته شده برای زبان سی شارپ هستند که معنی و مفهوم ثابتی دارند (معمولا به رنگ آبی می باشند) به این نوع کلمات ، کلمات کلیدی یا رزرو شده (Reserved words) می گویند و نامی که به وسیله برنامه نویس برای نام کلاس یا انتخاب می شود را شناسه (Identifier) گفته میشود.

نام کلاس `class`

```
{
```

تعریف داده ها

و عملیات بر روی آنها

```
}
```

زبان برنامه نویسی سی شارپ (C#)

▶ متد (Method) :

- ❖ متد مجموعه ای از دستورات است که برای انجام یک کار لازم است و هر متد مطابق با عملکردش نام گذاری می شود.
- ❖ به عنوان مثال می توان به متدهای `Write()` ، `WriteLine()` ، `ReadKey()` ، `ReadLine()` و نام برد.
- ❖ هر متد دارای یک جفت پارانتز باز و بسته است که در آن ممکن است ورودی هایی که برای انجام یک کار لازم است ذکر شود همچنین در انتها تعریف هر رکورد علامت سیمی کالن (;) ذکر می شود.
- ❖ متد `Main()` بدنه اجرایی برنامه است و هر دستوری که در این قسمت نوشته می شود به وسیله کامپیوتر به ترتیب اجرا می شود.
- ❖ ممکن است در یک برنامه متدهای زیادی تعریف شود اما حتما باید متدی به نام متد `Main()` تعریف شود چون نقطه آغاز برنامه است و اجرای یک برنامه از اولین دستور داخل آن شروع می شود.
- ❖ به نحوه نوشتن کلمات `class` , `static` , `void` و `Main` دقت کنید که فقط اول کلمه `Main` با حروف بزرگ نوشته شده است در غیر این صورت کامپایلر خطا می گیرد (زبان سی شارپ نسبت به حروف کوچک و بزرگ حساس است).

زبان برنامه نویسی سی شارپ (C#)

متد (Method) : ►

```
static void Main()  
{  
    ; دستور شماره ۱  
    ; دستور شماره ۲  
    ; دستور شماره ۳  
  
    ادامه دستورات  
}
```

ساختار اولیه دستورات برنامه نویسی سی شارپ

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            //
        }
    }
}
```

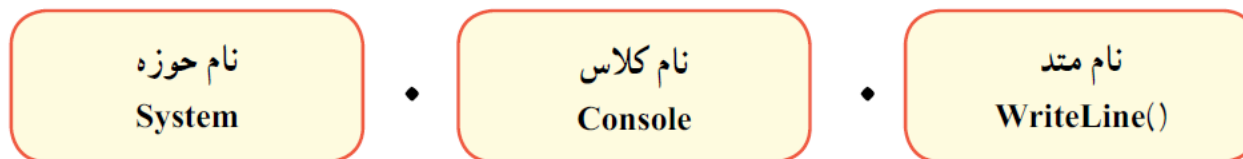
زبان برنامه نویسی سی شارپ (C#)

▶ **طریقه استفاده از متد های آماده :**

▶ تعداد زیادی متد در کلاس های آماده زبان سی شارپ وجود دارد که هر یک برای انجام کاری در نظر گرفته شده اند.

▶ برای استفاده از یک متد باید **اول نام فضا** **یه حوزه** سپس **نام کلاس** **مربوطه** و سپس **نام خود متد** را ذکر کرد و برای جدا کردن آن ها از یکدیگر از **علامت نقطه** استفاده می کنیم و در آخر نیز بعد از درج پارانتز باز و بسته علامت سیمی کالن را نیز قرار دهیم. (در آخر هر دستوری که به پایان می رسد باید علامت سیم کالن قرار داد)

▶ به عنوان مثال متد WriteLine() از کلاس console که برای نمایش اطلاعات بر روی صفحه نمایش یا عملیات ریاضی به کار می رود را به صورت زیر تعریف می کنیم: (به حروف کوچک و بزرگ دقت کنید)



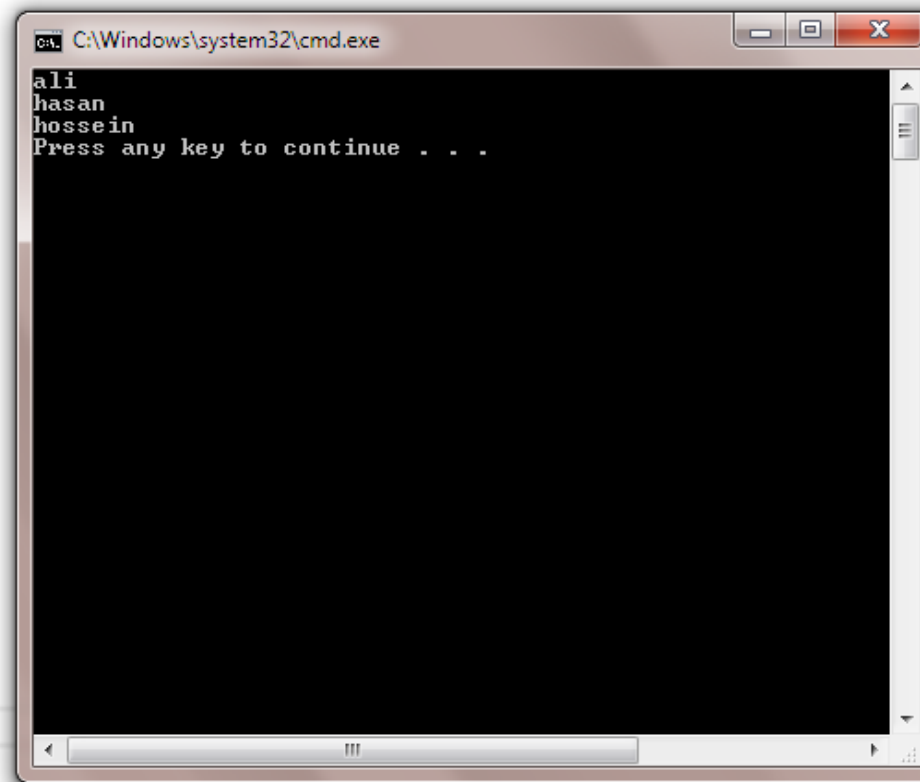
متد Write() و WriteLine()

- ▶ از این دو متد برای نمایش اطلاعات بر روی صفحه نمایش و همچنین نمایش یک عبارت ریاضی استفاده می شود.
- ▶ در هنگام استفاده از متد WriteLine() ، این متد یک سطر ایجاد می کند و اطلاعات را در آن قرار می دهد ولی در متد Write() هیچ سطری ایجاد نمی شود.
- ▶ برای نمایش اطلاعات در خروجی باید آن ها را ما بین علامت دابل کوتیشن (") قرار دهیم ولی برای نمایش عبارت ریاضی، عبارت ریاضی را به صورت خالی در داخل پارانتز می نویسیم.
- ▶ حروف، علامت ها و عبارتی که مابین علامت های نقل قول (دابل کوتیشن) نوشته می شود را **رشته (String)** می نامند. این حروف می توان فارسی، انگلیسی یا عیره باشد. و هر یک از حروف و علامت های یک رشته را یک **کاراکتر (Character)** می نامند.
- ▶ به عنوان مثال "Mohammad" یا "Ali123" هر کدام یک رشته می باشند و Mohammad از 8 کاراکتر و Ali123 نیز از 5 کاراکتر تشکیل شده است. چنانچه فاصله در رشته وجود داشته باشد، فاصله نیز یک کاراکتر محسوب می شود.
- ▶ **نکته :** برای این که در اول تعریف کلیه متدها عبارت System نوشته نشود ما این فضای نامی را در ابتدای دستورات به وسیله using معرفی می کنیم. یعنی در ابتدای دستورات عبارت using System; را می نویسیم.

متد WriteLine()

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("ali");
            Console.WriteLine("hasan");
            Console.WriteLine("hossein");
        }
    }
}
```

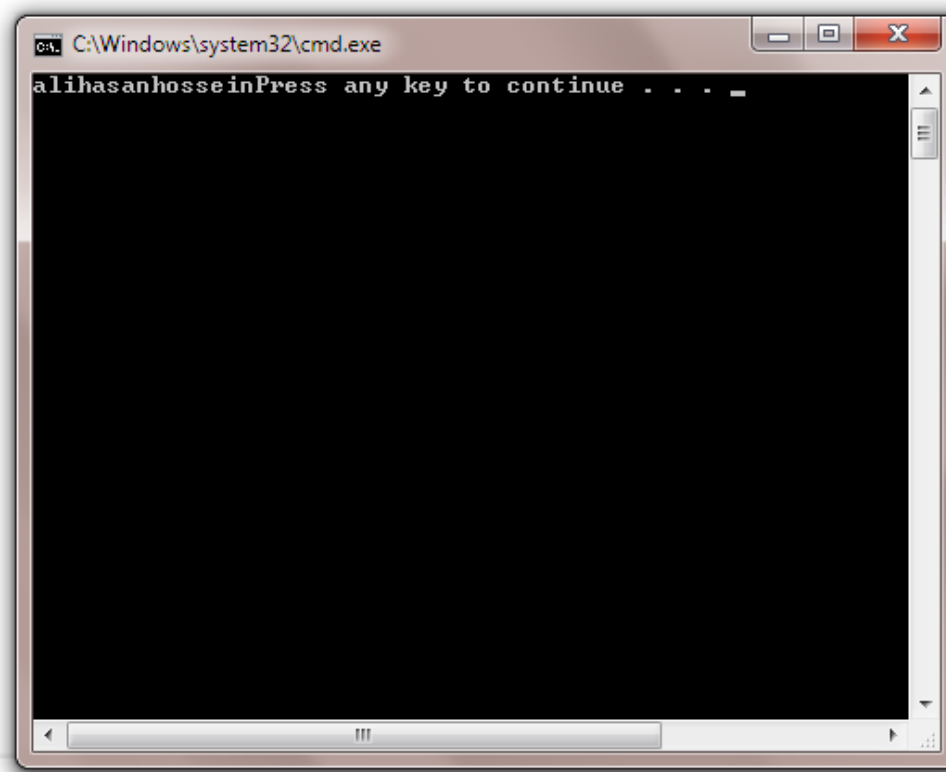


The screenshot shows a Windows command prompt window titled "cmd.exe" with the path "C:\Windows\system32\cmd.exe". The window displays the output of the program: "ali", "hasan", "hossein", and "Press any key to continue . . .".

متد Write()

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

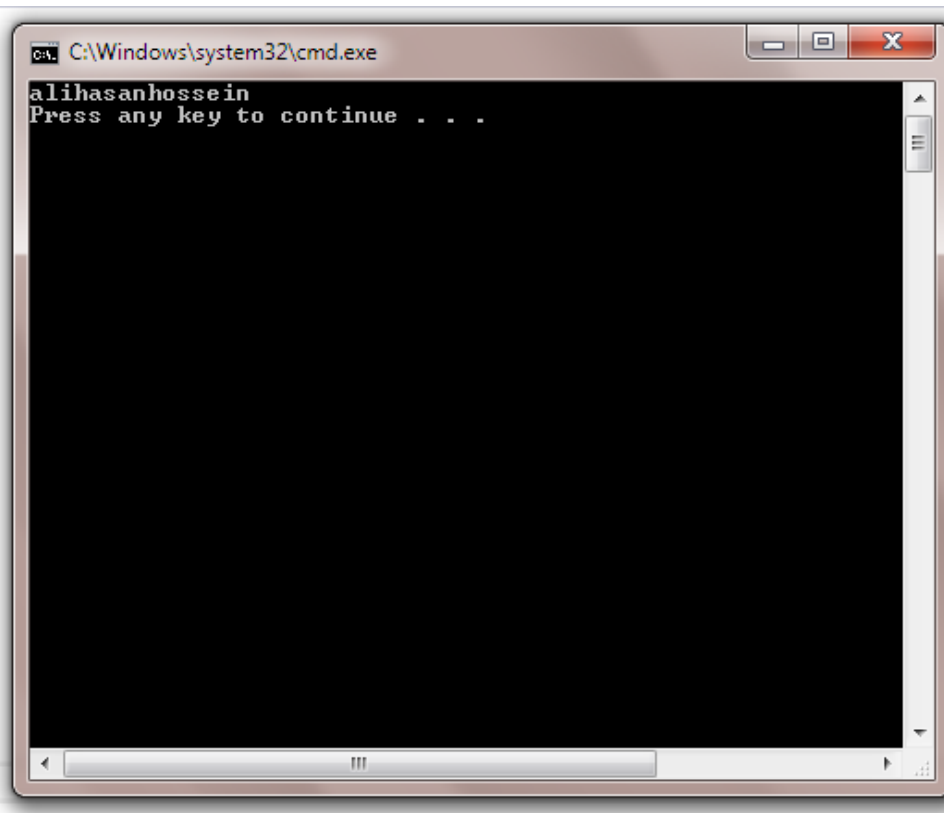
namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("ali");
            Console.Write("hasan");
            Console.Write("hossein");
        }
    }
}
```



متد Write() و WriteLine()

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("ali");
            Console.Write("hasan");
            Console.WriteLine("hossein");
        }
    }
}
```

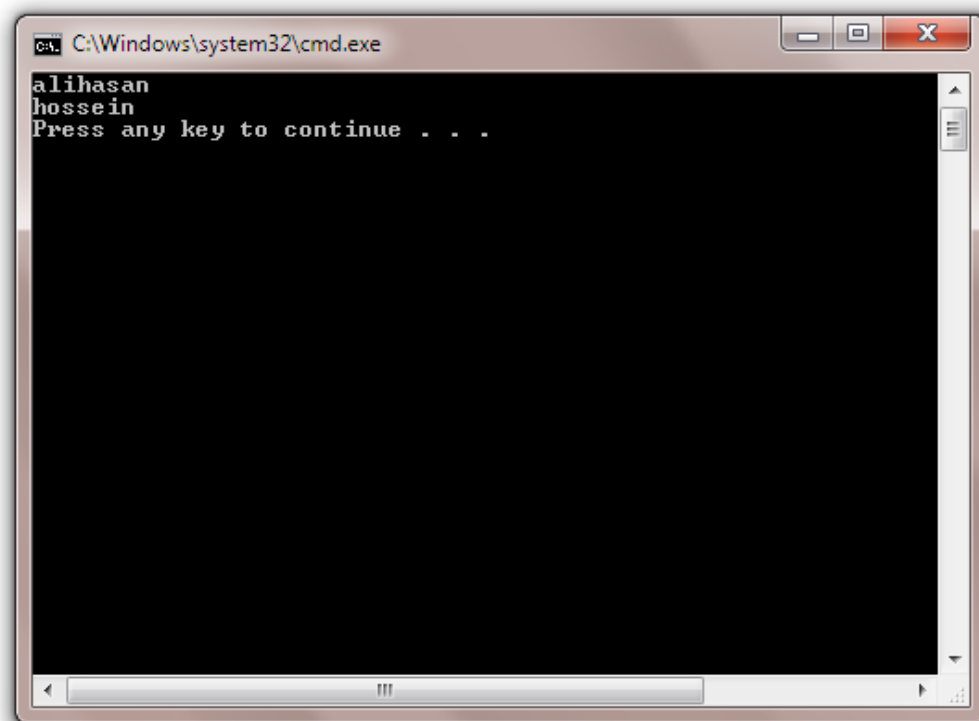


The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed on a black background with white text: "alihanhossein" on the first line and "Press any key to continue . . ." on the second line. A vertical line on the left side of the code editor in the image points to the corresponding code in the program.

متد Write() و WriteLine()

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("ali");
            Console.WriteLine("hasan");
            Console.WriteLine("hossein");
        }
    }
}
```

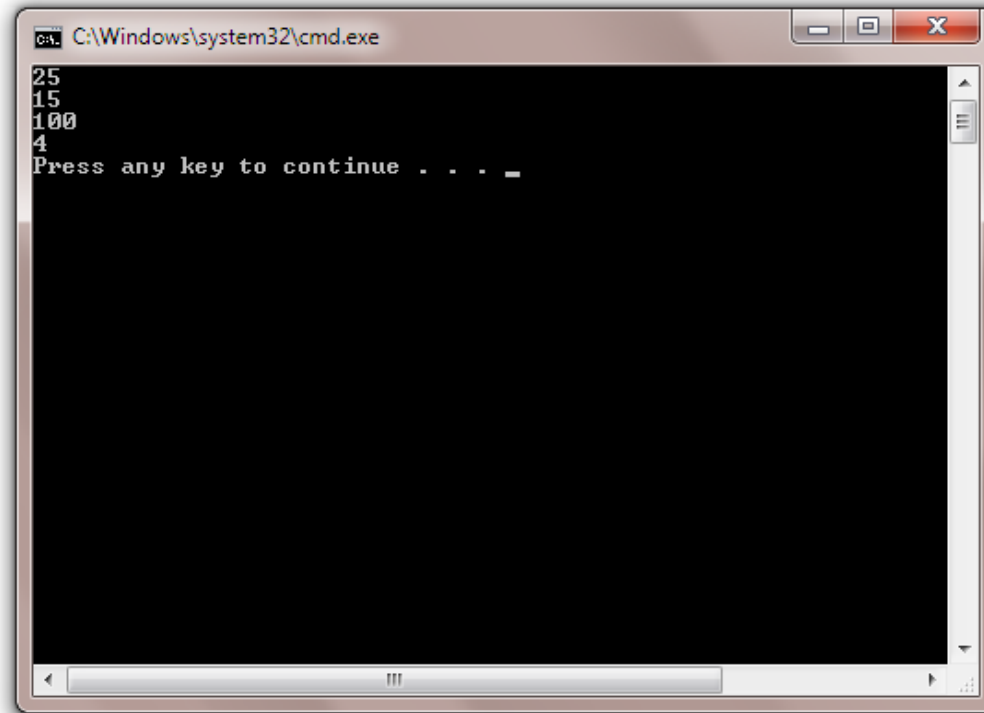


The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed on a black background with white text: "alihasan", "hossein", and "Press any key to continue . . .".

متد WriteLine()

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(20+5);
            Console.WriteLine(20-5);
            Console.WriteLine(20*5);
            Console.WriteLine(20/5);
        }
    }
}
```



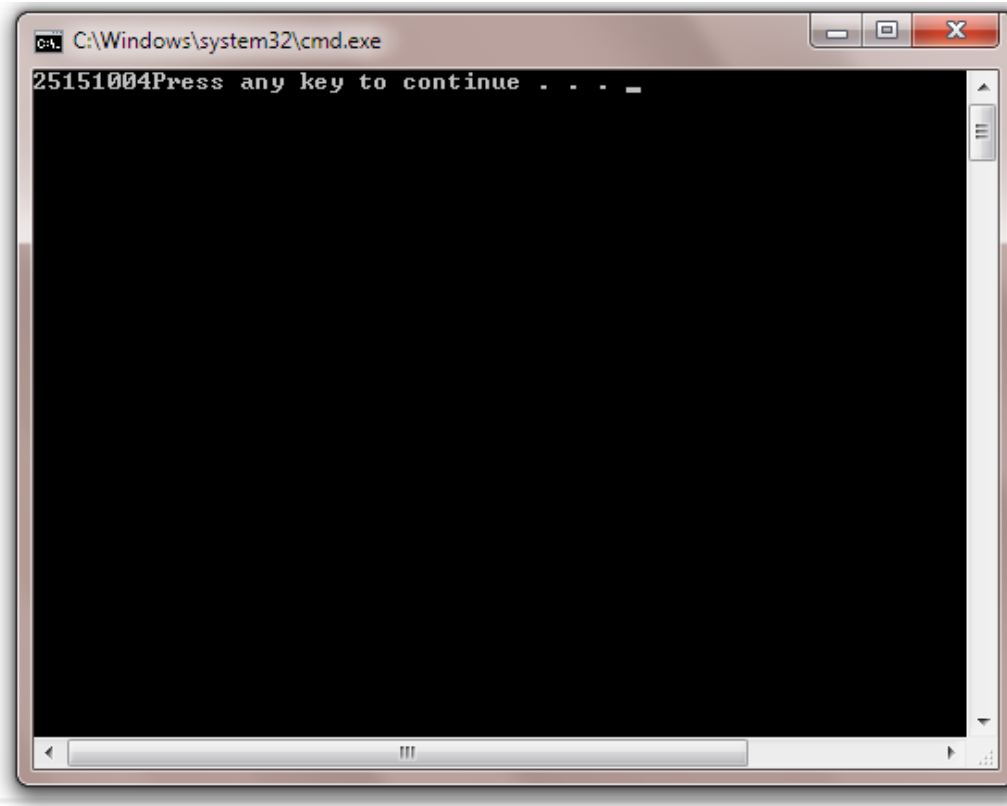
The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed as follows:

```
25
15
100
4
Press any key to continue . . . _
```

متد Write()

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

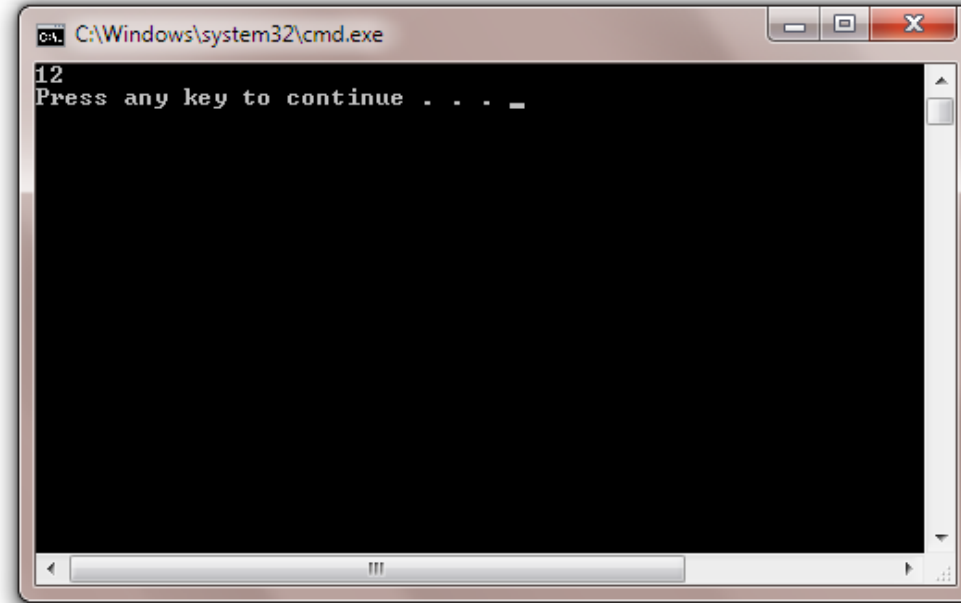
namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write(20+5);
            Console.Write(20-5);
            Console.Write(20*5);
            Console.Write(20/5);
        }
    }
}
```



متد WriteLine()

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(25/2);
        }
    }
}
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window contains the following text: "12" on the first line and "Press any key to continue" on the second line. The window has a standard Windows title bar with minimize, maximize, and close buttons.