

به نام خدا

جزوه آموزش برنامه نویسی به زبان C++

جلسه اول

ساختار C++

Basics Of C++

تهیه شده توسط: سعید

دانشجوی رشته کامپیوتر (نرم افزار)

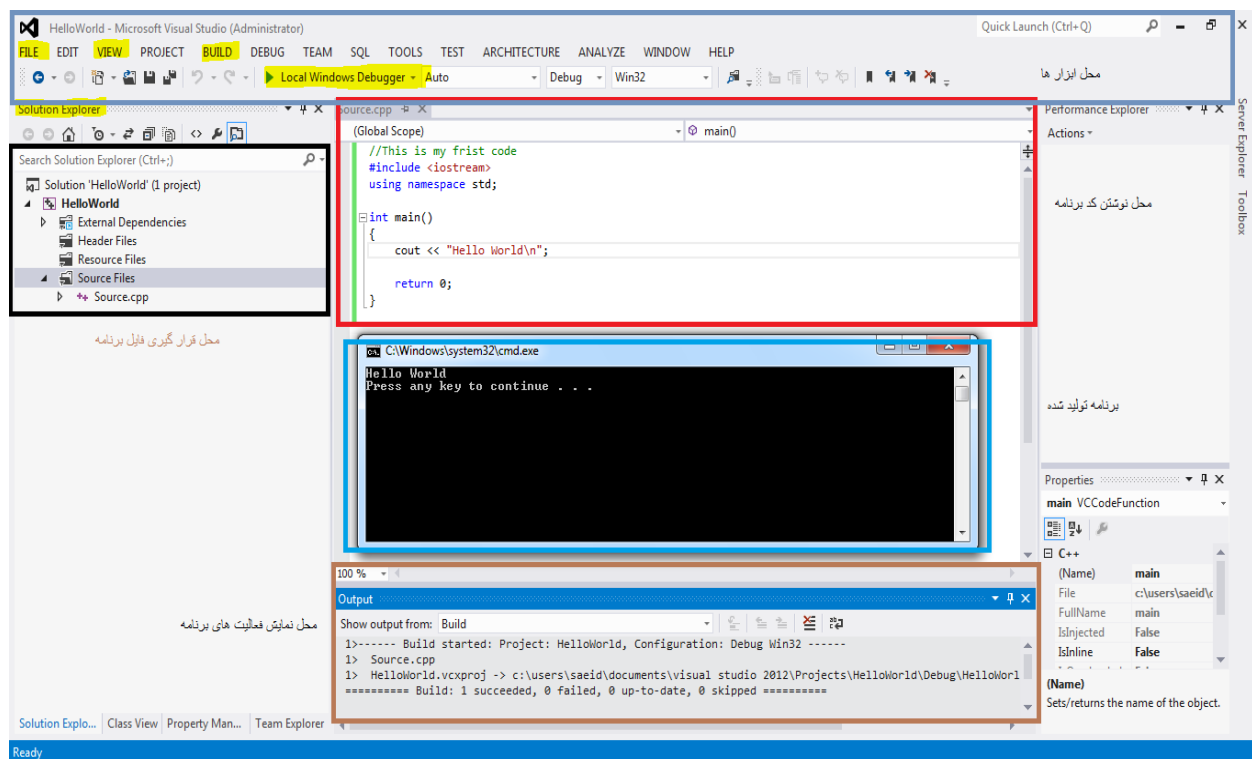
Saeid9063@hotmail.com

Saeid9063@yahoo.com

به نام خدا

شاید بهترین راه برای یادگیری یک زبان برنامه نویسی، نوشتن یک برنامه ابتدایی و ساده در محیط یک نرم افزار برای آشنایی با ساختار آن زبان باشد .

در آغاز عکسی از برنامه Visual Studio C++ را نشان می دهیم تا با محیط نرم افزار C++ آشنا شوید . هرچند نرم افزارهای دیگره هم در این خصوص وجود دارند مانند Borland C++ یا Turbo C++ .



محیط برنامه Visual Studio

برنامه ای را که در بالا می بینید سورس کد اولین برنامه ما و همچنین ابتدایی ترین ساختار برنامه C++ می باشد. قبل از هر چیز به بررسی سطحی و آشنایی اولیه با خطوط برنامه Hello world! می پردازیم:

```
// my first program in c++
#include <iostream.h>
int main()
{
    std::cout << "Hello world" ;
    return 0 ;
}
```

// my first program in c++

این یک خط Comment یا توضیحات می باشد. در برنامه نویسی، توضیحات خطوطی هستند که کامپایل نمی شوند و فقط برای خوانایی برنامه بکار برده می شوند. در برنامه های بزرگتر و با زیاد شدن خطوط برنامه، توضیحات به فهم راحتتر برنامه برای برنامه نویس و دیگر کسانی که کد را مورد بررسی قرار می دهند کمک شایانی می نماید .

برای بوجود آوردن توضیحات در برنامه C++ از دو الگو می توان استفاده نمود :

- اگر بخواهیم توضیحات را در یک خط قرار دهیم از علامت // قبل از توضیحات استفاده می کنیم. در مثال قبل ما از این الگو بهره بردیم .
- در صورت زیاد بودن توضیحات واشغال چند سطر از برنامه توسط آن از علامت /* */ استفاده کرده و توضیحات را در بین آن قرار می دهیم .

```
/* my
first program
in c++ */
```

خطوطی که با علامت # شروع شده و در بالای برنامه قرار می گیرند خطوط فرمان به پردازشگر می باشند که در C++ در اصطلاح فایل سرآیند نامیده می شوند

#include <iostream>

از فایل های سرآیند بعنوان کتابخانه های C++ یاد می کنند که از قبل نوشته شده اند و ما برای استفاده از برخی از توابع و روالها از آنها استفاده می کنیم . کامپایلر فقط کلمات کلیدی را می شناسد و همانطور که گفته شد برای استفاده از یک سری دستورات و توابع مانند دستورات ورودی و خروجی و ... باید از این فایل های سرآیند استفاده نماییم و اگر استفاده نکنیم امکان برنامه نویسی بوجود نخواهد آمد که به تفصیل در آینده در مورد این فایلها و مورد استفاده شان صحبت خواهیم کرد.

نکته : در کامپایلر های قدیمی تر و در کتاب های قدیمی در آخر کتابخانه حرف h. قرار میدادند که دیگر نیازی به این کار نیست.

`int main ()`

این خط تعریفی برای تابع اصلی برنامه می باشد. در واقع برنامه با کامپایل از این نقطه شروع و پردازش می شود. هر برنامه C++ باید دارای تابع `main()` باشد. در این تابع است که بلوکها و خطوط برنامه نوشته میشود. باید گفته شود هرچند خطوطی بالاتر از تابع اصلی نوشته شده اما بخاطر داشته باشید که برنامه از این نقطه شروع خواهد شد. عبارت `int` به معنی `integer` یا اعداد یک کلمه کلیدی می باشد و برای تعریف متغیرهای از نوع صحیح بکار می رود که در ادامه آموزش برنامه نویسی بطور مفصل مورد بررسی قرار می گیرد `Main`. نامی برای تابع اصلی است که تغییر نمی کند و `()` علامتی است که در ادامه هر تابعی قرار می گیرد که توابع نیز مفصلاً در ادامه مورد بحث قرار می گیرند و اطلاعات فوق جنبه آشنایی با آنها را دارد .

}

آکولاد باز در واقع شروع یک بلوک از دستورات را تعریف می کند که در این برنامه بدنه تابع اصلی (`main`) را در بر می گیرد و با علامت

{

یا آکولاد بسته پایان بلوک را تعیین می کنیم .

`cout << "Hello world" ;`

تابع `>>cout` در تابع سرآیند `iostream` در زبان C++ قرار دارد و موجب ارسال اطلاعات به خروجی و چاپ و نمایش آن بر روی مانیتور کاربر می شود. با نوشتن این دستور عبارت `Hello world!` بر روی صفحه نمایش پس از کامپایل بدون خطای برنامه به نمایش در می آید. از علامت " " برای متغیرهای رشته ای که در اصطلاح `string` گفته می شوند استفاده می شود. از علامت ; نیز در انتهای هر دستور در C++ استفاده می گردد تا بوسیله آن پایان آن دستور را به پردازشگر اعلام نمود .

`return 0 ;`

این عبارت مقدراً صفر را به تابع در برگیرنده خود که در این مثال تابع اصلی است برمی گرداند که این مورد در مبحث توابع و انواع بازگشتی آن توضیح داده خواهد شد .

`namespace`

باید بگم که فضاهای نام هم در ساختار C++ نقش اساسی دارند. این فضاها مجموعه دیگری از کتابخانه های C++ می باشند که در استفاده از بعضی عناصر مانند رشته ها و ... کمک شایانی به کاربران می کنند.

فضا های نام با استفاده از الگوی زیر قابل استفاده اند و بعد از فایل های کتابخانه ای `include` در برنامه قرار می گیرند که در بخش انواع یک نمونه از اونا رو استفاده میکنیم.

```
// my first program in c++
#include <iostream.h>
Using namespace std;
int main()
```

```
{
    cout << "Hello world" ;
    return 0 ;
}
```

برخی از ویژگیهای زبان : C++

انعطاف پذیری و غنای بالا

زبان برنامه نویسی سیستم است و با آن می توان برنامه های سیستمی را نوشت، بدین معنی که مستقیماً می تواند با سخت افزار و نرم افزار ارتباط برقرار نماید .

زبان شی گراست

Case sensitive است ، یعنی نسبت به کوچکی و بزرگی حروف حساس بوده و بین این دو تمایز قائل است. توصیه می شود که برنامه ها را با حروف کوچک بنویسید While. برابر نیست با WHILE

برخی از ویژگیهای دستورات C++

هر دستور باید به ; ختم شود .

حداکثر طول یک دستور، 255 کاراکتر است .

هر دستور می تواند در یک و یا چند سطر نوشته شود .

در هر سطر می توان چندین دستور را نوشت.(این کار توصیه نمی شود) .

توضیحات می توانند بین /* و */ در چندین سطر و یا بعد از // و در فقط یک سطر نوشته شوند .

کلمات کلیدی در C++

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

این کلمات، کلمات کلیدی هستند و کامپایلر فقط این کلمات را می شناسد و هرآنچه غیر از کلمات کلیدی در برنامه ++C نوشته شود باید برای کامپایلر درست تعریف شود. در این بین توابعی برای گرفتن اطلاعات و یا چاپ اطلاعات و چیزهای دیگری وجود دارد که برای استفاده از آنها باید از فایل سرآیند مربوطه استفاده نماییم که از قبل نوشته شده اند مثل توابع `cin` , `cout` که توابع ورودی خروجی هستند که در فایل سرآیند `iostream` وجود دارند و همچنین تابع `getch` که برای زدن یک کلید از صفحه کلید است و در فایل سرآیند `conio` قرار دارد .