

بسمه تعالی



درسنامه برنامه نویسی کامپیوتری
گروه فیزیک
دانشکده علوم پایه، دانشگاه قم

جلسه اول:

اولین برنامه در ++C

محمد رضا مظفری

۱۶ مهر ۱۳۹۹

چرا نباید برنامه‌های علمی را در ++C نوشت

شایان ذکر است که ++C بهترین زبان برای هر موقعیتی نیست. بعضی از افراد معتقدند که زبان های دیگر ممکن است سریعتر باشند. بسیاری از برنامه نویسان علمی بر این باورند که Fortran همیشه از نظر سرعت پردازش بهترین عملکرد را خواهد داشت و ++C را بر اساس ویژگی هایی مانند تعقیب اشاره‌گر و جستجوی روش مجازی کنار می‌گذارند که منجر به اجرای کد با سرعت غیربهبوده در حین فرایند پردازش می‌شود (نگران نباشید اگر این اصطلاحات را نشنیده‌اید، یا معنی آنها را نمی‌دانم - شاید هرگز نیازی به این کار نداشته باشید!). این موضوع ممکن است حقیقت داشته باشد، اما این واقعیت که شی‌گرایی منجر به خوانایی بیشتر برنامه‌ها علمی می‌شود، می‌تواند این زبان را به یک زبان پیشرفته، منعطف و معقول تبدیل کند.

بعضی اوقات زبانهای دیگر برای یک کار تخصصی بهتر هستند. زبان های نوشتاری ^۱ مانند Perl و Python برای پردازش متن و دستکاری رشته‌ها ایده‌آل هستند. اگر می‌خواهید ستون اعداد را از میان پرونده‌ها جمع بزنید، می‌توانید یک برنامه ++C بنویسید، اما اجرای یک اسکریپت کوتاه و یکبار مصرف بسیار سریعتر است.

برخی از زبان‌ها برای نوشتن برنامه‌های نمونه آماده ^۲ یا ترسیم نمودارها بهتر هستند. Matlab در زمینه نمونه سازی سریع بسیار سرآمد است. برای مثال یک الگوریتم جبر خطی خاص بر روی طیف وسیعی از ماتریس‌ها با اندازه‌ها و ساختارهای مختلف، فقط چند خط در Matlab کد نیاز است. اما ++C ممکن است مجبور باشید چندین فایل بنویسید و آنها را در برابر کتابخانه‌های دیگری کامپایل کنید (در این ارتباط بعداً صحبت خواهیم کرد). Matlab همچنین از مزیت یک محیط توسعه گرافیکی کاملاً یکپارچه ^۳ برخوردار است و بسیاری از کارهای برنامه نویسی را بدون نیاز به ابزارهای اضافی به راحتی انجام می‌دهد. علاوه بر این، Matlab دارای یک محیط داخلی برای رسم نمودار است، بنابراین اگر می‌خواهید به سرعت نتایج الگوریتم های خود را نمایش دهید، Matlab بهترین انتخاب شما خواهد بود.

بنابراین زبان ++C در هر شرایطی ممکن است بهترین انتخاب نباشد. با این حال، شرایط بسیاری وجود دارد که ++C برای یک مسئله خاص مناسب ترین حالت را دارد (که در رشته‌های علوم مانند فیزیک و ریاضی می‌توان بیشمار از این مسائل خاص پیدا کرد). بحث بالا ممکن است کافی باشد تا

¹ Scripting languages

² Prototype programs

³ Integrated development environment (IDE)

شما را متقاعد کند که ارزش آن را دارد که با C++ شروع به کار کنید.

هدف از این درس

بیشتر برنامه های C++ برای محاسبات علمی^۴ را می توان با استفاده از تنها بخشی از کل توانایی های زبان C++ بطور موثر نوشت. این درسنامه بر جنبه هایی از زبان C++ تمرکز دارد که شما به احتمال زیاد با آنها در پروژه های کارشناسی و یا کارشناسی ارشد خود روبرو خواهید شد. هنگام نوشتن برنامه های خود، ممکن است نیاز به درک یکی از ویژگی های پیشرفته تر زبان داشته باشید که در بخشهای انتهای دروه آموزش زبان C++ آنرا تا حدی معرفی خواهیم کرد.

```

1  #include <iostream>
2
3  int main ()
4  {
5      /* This is a comment and will be ignored by the compiler
6         Comments are useful to explain in English what
7         the program does */
8
9      // Print "Hello World" to the screen
10     std::cout << "Hello World" << std::endl;
11
12     return 0;
13 }
```

شکل ۱: برنامه ی Hello World

اولین برنامه در C++

معرفی هر زبان برنامه نویسی معمولاً با استفاده از برنامه ای که در آن متن Hello World را روی صفحه نمایش (مونیتور) چاپ می کند، آغاز می شود. یک مثال ساده از برنامه C++ که این کار را انجام می دهد در شکل ۱ نشان داده شده است.

در خط ۱ این کد، فایل سربرگ^۵ iostream را درج می کنیم. نام iostream مربوط به جریان

^۴ Scientific computing

^۵ Header file

ورودی و خروجی می‌شود. در هر برنامه ++C که داده‌ها ورودی را از صفحه کلید^۶ وارد می‌کند یا داده‌های خروجی را در صفحه فرمان^۷ یعنی همان صفحه نمایش مشکی نشان می‌دهد. دومین ویژگی قابل توجه این است که بخشی از کد وجود دارد که:

- با خطی از کد بصورت

```
3 int main ()
```

شروع می‌شود (خط ۳ در شکل ۱ را نگاه کنید)

- بدنه کد بین براکت‌های مجعد باز { در خط ۴ و بسته } در خط ۱۳ محصور می‌شود.

- کد داخل براکت‌های مجعد با

```
12 return 0;
```

در خط ۱۲ به پایان می‌رسد.

بخشی از کد بین براکت‌های مجعد قرار دارد شامل دستورالعملهایی است که می‌خواهیم کامپیوتر آنها را اجرا کند. توجه داشته باشید که توضیحات در خطوط ۵، ۶، ۷ و ۹ برای کمک به خواندن و فهمیدن محتوای کد درج شده است.

```
5 /* This is a comment and will be ignored by the compiler
6 Comments are useful to explain in English what
7 the program does */
8
9 // Print "Hello World" to the screen
```

هر چیزی که بین شروع‌کننده توضیحات (یعنی `/*`) و پایان‌دهنده توضیحات (یعنی `*/`) قرار می‌گیرد و یا هر خطی که با `/**` شروع می‌شود بعنوان توضیحات می‌باشد و هنگامی که برنامه توسط کامپیوتر اجرا می‌شود، نادیده گرفته می‌شوند.

در ادامه بر روی خطوط ۱۰ و ۱۲ تمرکز می‌کنیم،

این خطوط کد هر کدام شامل یک دستورالعمل برای رایانه هستند و به عنوان دستورات شناخته می‌شوند.

⁶ Keyboard

⁷ Console or Terminal or Command(cmd)

```

10     std::cout << "Hello World" << std::endl;
11
12     return 0;

```

توجه داشته باشید که عبارتها با نقطه-ویرگول^۸ ; پایان می یابند. در حال حاضر کافی است که خواننده بداند که خط ۱۰ برنامه، خطی از کد است که کامپیوتر را به چاپ مطالب درون گیومه‌ها^۹ بر روی صفحه نمایش هدایت می کند. endl یک خط جدید را نشان می دهد. بنابراین عبارت Hello World و به دنبال آن یک خط جدید روی صفحه چاپ می شود. کلمه cout مختصر شده‌ی عبارت خروجی کنسول^{۱۰} یعنی چاپ روی صفحه نمایش است.

برای ذخیره یک برنامه C++ معمولاً از پسوند cpp استفاده می‌شود. این انتخاب کاملاً شخصی است، بطوریکه می‌توان از پسوندهای .cxx ، .c++ ، .cc ، و .C. نیز استفاده کرد.

کلمه int در ابتدای خط ۳

```

3 int main ()

```

نشان می دهد که آخرین خط کد درون براکت‌های مجعد یک مقدار صحیح را برمی گرداند. این کار با استفاده از عبارت در خط ۱۲

```

12     return 0;

```

انجام می شود. بازگشت مقدار صفر به کامپیوتر نشان می دهد که برنامه بدون بروز مشکلی به پایان رسیده است.

قبل از شروع به توضیح که چگونگی کامپیوتر Hello World را روی صفحه نمایش چاپ می‌کند، مکث می کنیم تا در مورد برخی از سبک‌ها نوشتاری که باید از آنها آگاه باشید، بحث کنیم. در برنامه نشان داده شده در شکل ۱ مشاهده می‌شود که تمام خطوط برنامه بین براکت‌های مجعد دارای تورفتگی^{۱۱} هستند.

این نوع نوشتار اجباری نیست. با این حال، در هنگام برنامه‌نویسی ایجاد این فرورفتگی‌ها در خطوط یک عمل استاندارد است که باعث افزایش قابلیت خوانایی برنامه و فهم آن می‌شود. در قسمت‌های جلوتر

⁸semi-colon

⁹quotations

¹⁰Console output

¹¹indentation

```

4 {
5     /* This is a comment and will be ignored by the compiler
6     Comments are useful to explain in English what
7     the program does */
8
9     // Print "Hello World" to the screen
10    std::cout << "Hello World" << std::endl;
11
12    return 0;
13 }
```

وقتی در برنامه بیش از یک جفت براکت مجعد نیاز می‌شود، واضح‌تر می‌توان این موضوع را احساس کرد. تعداد فضاهاى فرورفته کاملاً به تصمیم برنامه‌نویس ارتباط دارد.

نکته دیگری که باید در نظر داشته باشید این است که هنگام اجرای برنامه، از همه فضاهاى سفید چشمپوشی می‌شود. نکته آخر اینکه طول خطوط در C++ به اندازه و خواست برنامه نویس می‌باشند. به طور کلی توصیه می‌شود یک خط طولانی را به طور بالقوه در چندین خط تقسیم کنید. در آینده هنگام نوشتن جملات پیچیده‌تر، این موضوع را نشان خواهیم داد.

کامپایل یک برنامه C++

یک برنامه C++ باید قبل از اجرا، کامپایل شود. روش‌های مختلفی وجود دارد که قصد داریم درباره‌ی آنها بحث کنیم.

استفاده از محیطهای توسعه یکپارچه

وقتی اولین قدم‌های خود را برای یادگیری یک زبان برنامه‌نویسی جدید بر می‌دارید، ممکن است نخواهید زمان زیادی را برای نصب نرم‌افزار جدید سرمایه‌گذاری کنید. به همین دلیل، توصیه می‌شود که نوشتن برنامه‌ها را در یک ویرایشگر متن^{۱۲} مانند notepad++^{۱۳} و یک کامپایلر خط فرمان^{۱۴} مانند MinGW^{۱۵} شروع کنید (که در بخش بعد درباره آنها صحبت خواهیم کرد).

با این حال، با بزرگ شدن برنامه ممکن است نیاز به مدیریت چندین فایل باشد که هر کدام حاوی

¹²Text editor

¹³<https://notepad-plus-plus.org>

¹⁴Command line compiler

¹⁵<http://www.mingw.org>

قسمت‌های مختلف از برنامه هستند. در این شرایط برای توسعه برنامه، توصیه می‌شود که از یک محیط توسعه یکپارچه (IDE) استفاده شود.

عملکرد IDE های مختلف با توجه به سطح پیچیدگی آنها متفاوت است. ویژگی‌های مشترک IDE ها در زیر ذکر شده است. اگر تمام اصطلاحات استفاده شده را به طور کامل درک نمی‌کنید نگران نباشید، کم‌کم این واژه‌ها روشن می‌شوند.

- یک ویرایشگر برنامه معمولاً شامل برجسته‌سازی و رنگ آمیزی کلمات کلیدی، تورفتگی‌های خودکار خطوط برنامه و شناسایی ساختارهای غیرقانونی برنامه‌نویسی می‌باشد.
- ویرایش با آگاهی از زمینه برنامه که چه توابع و متغیرهای برای فراخوانی و استفاده وجود دارد.
- ساخت اتوماتیک برنامه بدون دخالت کاربر، این کار با آماده سازی Makefile در ویرایشگر انجام می‌شود.
- امکان نوشتن، ذخیره و کامپایل برنامه بطور پیوسته.
- اشکال‌زدایی گرافیکی^{۱۶} این امکان را فراهم می‌کند تا کاربر بتواند برنامه را در نقاط مهم مکث داده و وضعیت داخلی متغیرها را در آن نقاط بررسی کند.

معرفی IDE های پر استفاده

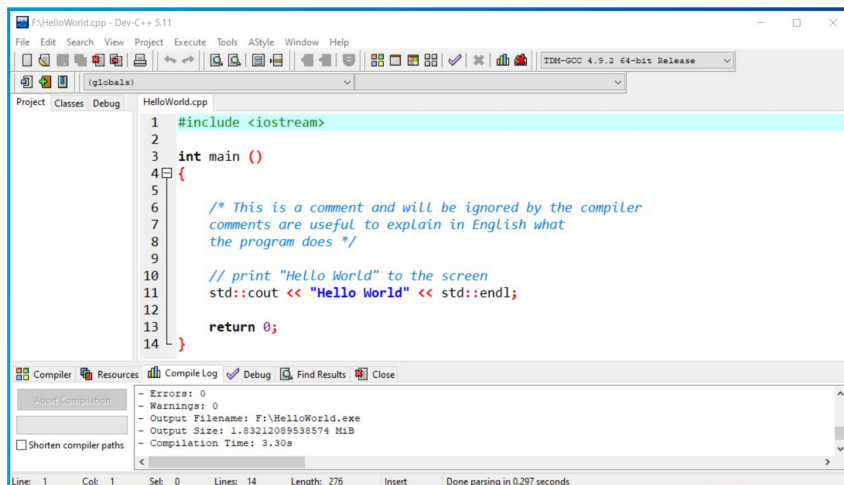
● Dev-C++

یکی از پر کاربردترین و پر طرفدارترین IDE ها برای برنامه‌نویسی در محیط C/C++ است. این IDE رایگان و البته ساده، فقط برای سیستم عامل ویندوز وجود دارد. از ویژگیهای این IDE، برنامه نویسی به دو صورت پروژه‌ای و تک فایل است. این نرم افزار را می‌توان از آدرس زیر دانلود کرد،

<https://sourceforge.net/projects/orwelldevcpp>

لوگو و محیط این نرم‌افزار در شکل ۲ داده شده است. کامپایلر پیش فرض در این IDE، MinGW نام دارد که یک کامپایلر رایگان می‌باشد.

¹⁶graphical debugging



شکل ۲: لوگو و محیط نرم افزار Dev-C++

● CodeLite

یکی دیگر از IDE های رایگان که علاوه بر C/C++ قابلیت برنامه نویسی در PHP و JavaScript را هم دارد. این IDE قابل استفاده برای تمامی سیستم عاملهای Windows، Mac و Linux می باشد. برنامه نویسی در آن بسیار جذاب است. کامپایلر پیش فرض ندارد و برای نسخه ویندوزی آن ابتدا نیاز به نصب کامپایلر خط فرمان MinGW است. این نرم افزار را می توان از آدرس زیر دانلود کرد،

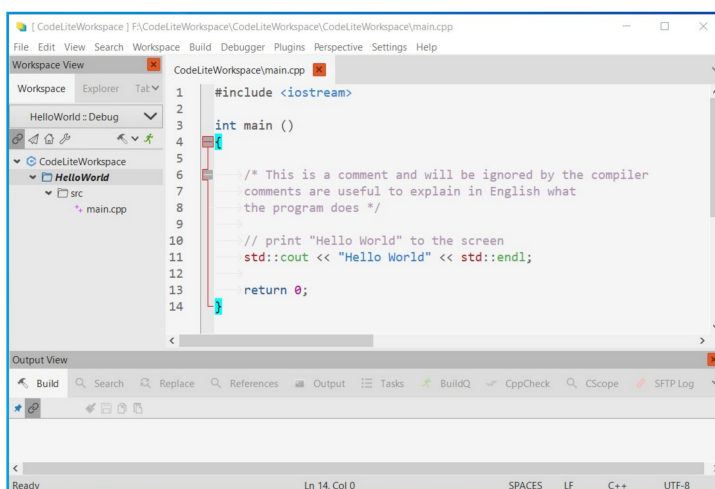
<https://codelite.org>

لوگو و محیط این نرم افزار در شکل ۳ داده شده است. در این IDE، برنامه نویسی فقط بصورت پروژه ای امکان پذیر است.

● Cevelop

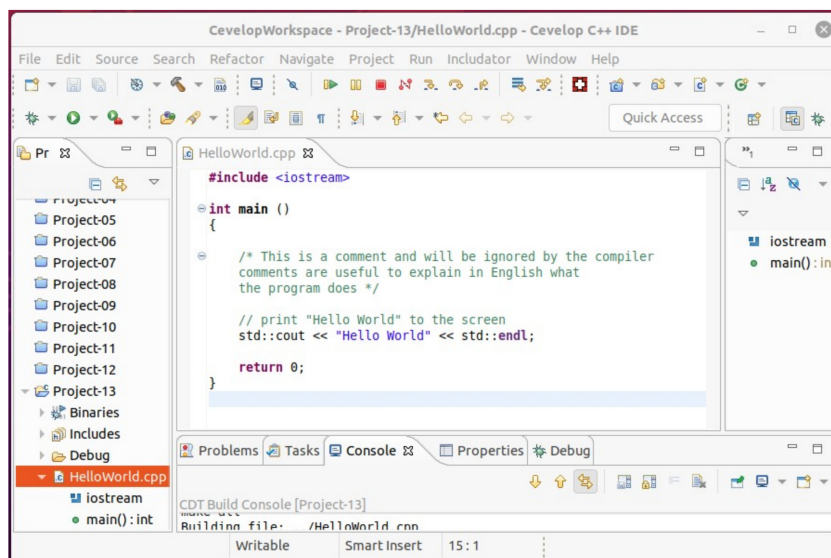
یکی دیگر از IDE های رایگان برای برنامه نویسی در C/C++ است. این IDE نیز قابل استفاده برای تمامی سیستم عاملهای Windows، Mac و Linux می باشد. برنامه نویسی در آن بسیار راحت و جذاب است. کامپایلر پیش فرض آن در ویندوز clang است. اگرچه شرایط اضافه کردن کامپایلر خط فرمان MinGW به این IDE وجود دارد. این نرم افزار را می توان از آدرس زیر دانلود کرد،

<https://www.cevelop.com>



شکل ۳: لوگو و محیط نرم‌افزار CodeLite

لوگو و محیط این نرم‌افزار در شکل ۴ داده شده است. در این IDE، برنامه نویسی فقط بصورت پروژه‌ای امکان‌پذیر است.



شکل ۴: لوگو و محیط نرم‌افزار Cevelop

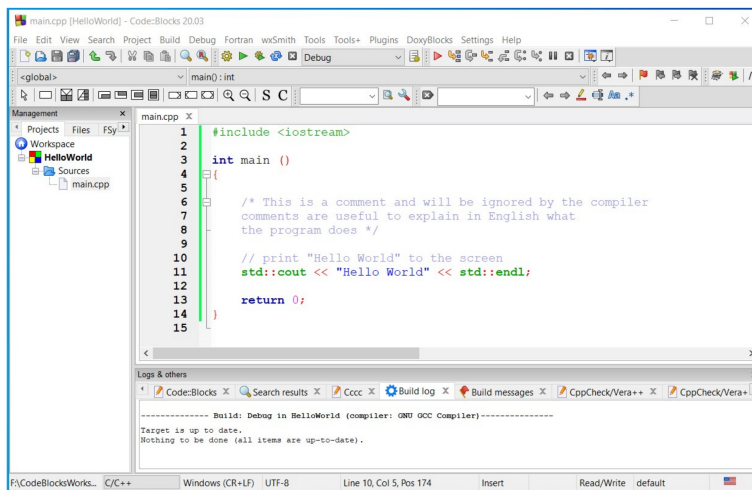
● Code::Blocks

یکی دیگر از IDE های رایگان برای برنامه‌نویسی در C/C++ و Fortran است. این IDE نیز قابل استفاده برای تمامی سیستم‌عاملهای Windows، Mac و Linux می‌باشد. برنامه‌نویسی در آن راحت است. کامپایلر پیش‌فرض آن در ویندوز MinGW است. این نرم‌افزار را می‌توان از

آدرس زیر دانلود کرد،

<http://www.codeblocks.org>

لوگو و محیط این نرم‌افزار در شکل ۵ داده شده است. در این IDE، برنامه نویسی فقط بصورت پروژه‌ای امکان‌پذیر است.



شکل ۵: لوگو و محیط نرم‌افزار Code::Blocks

کامپایل در خط فرمان

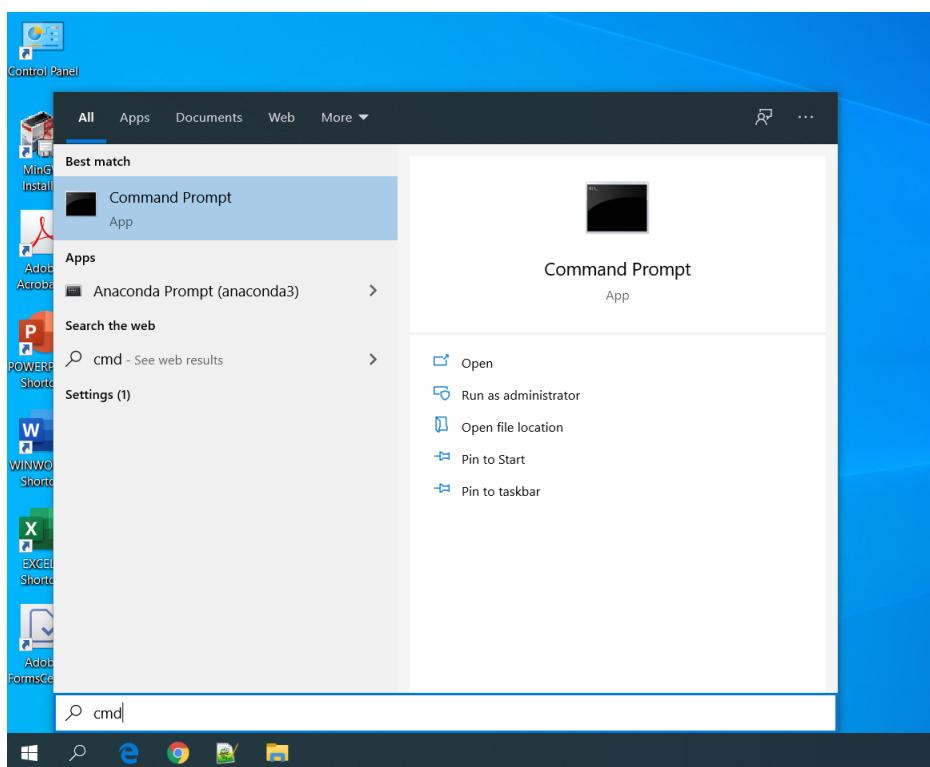
یکی از متداول‌ترین کامپایلرهای خط فرمان در سیستم عامل ویندوز MinGW^{۱۷} نام دارد. این کامپایلر شامل کامپایلرهای gcc^{۱۸} می‌باشد که برنامه‌های تحت C/C++ و Fortran را پشتیبانی می‌کند. برای کامپایل کردن برنامه‌های یاد شده نیاز به محیط خط فرمان cmd است. این کامپایلر را می‌توان از آدرس زیر دانلود کرد،

<http://www.mingw.org>

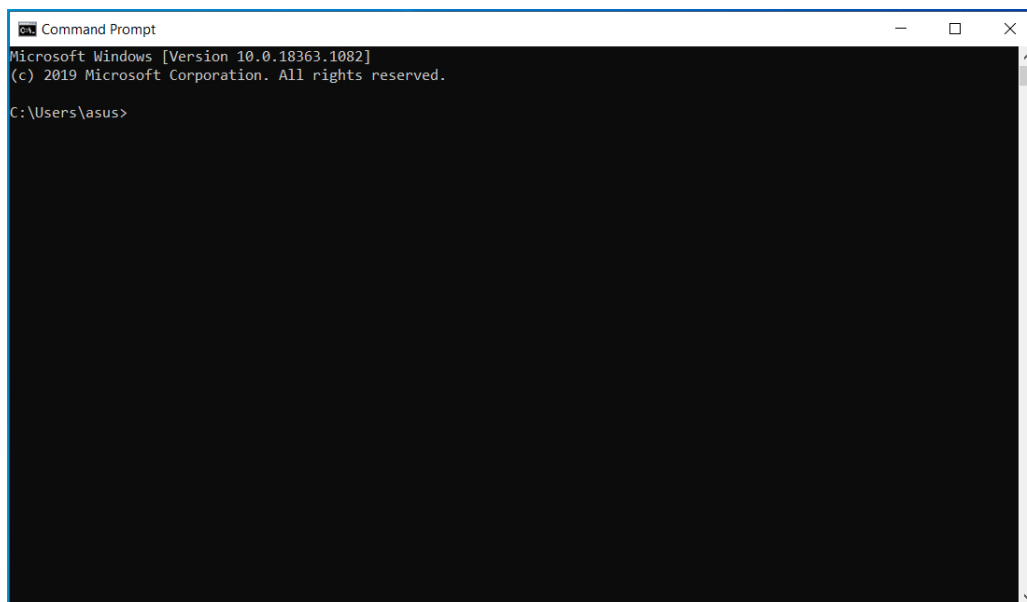
برای بکارگیری کامپایلرهای MinGW در سیستم عامل ویندوز می‌توان از command prompt در start ویندوز استفاده کرد. شکل فراخوانی خط فرمان ویندوز و صفحه آن در شکل و نشان داده شده است.

¹⁷Minimalist GNU for Windows

¹⁸GNU Compiler Collection



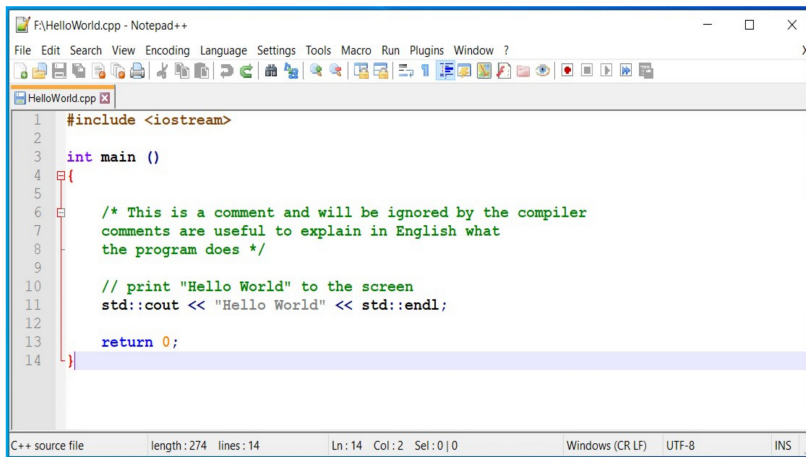
شکل ۶: فراخوانی خط فرمان ویندوز



شکل ۷: فراخوانی خط فرمان ویندوز

در حالتی که قصد داشته باشید از خط فرمان برای کامپایل و اجرای برنامه‌ها استفاده کنید، نیاز به یک ویرایشگر مناسب است که توانایی برجسته‌سازی و رنگ آمیزی کلمات کلیدی زبان مورد نظر را داشته باشد. یک ویرایشگر مناسب در ویندوز که بسیاری از زبانهای برنامه‌نویسی را پشتیبانی می‌کند،

notepad++ است. لوگو و محیط این نرم افزار در شکل ۸ داده شده است.



```
1 #include <iostream>
2
3 int main ()
4 {
5
6     /* This is a comment and will be ignored by the compiler
7     comments are useful to explain in English what
8     the program does */
9
10    // print "Hello World" to the screen
11    std::cout << "Hello World" << std::endl;
12
13    return 0;
14 }
```

The screenshot shows the Notepad++ interface with a C++ source file named 'HelloWorld.cpp'. The code is as follows:



شکل ۸: لوگو و محیط نرم افزار notepad++