



به نام خدا

دستور کار کارگاه مبانی کامپیوتر و برنامه‌نویسی

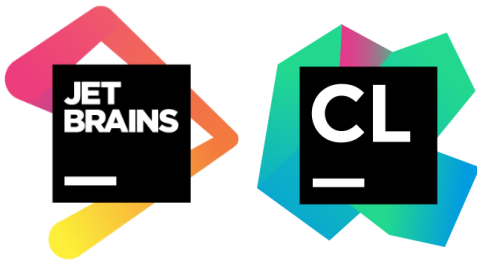


جلسه چهارم (پیش‌نیاز)

خودآموز آشنایی با محیط برنامه‌نویسی JetBrains' CLion

فهرست مطالب

- ۱ خودآموز آشنایی با محیط برنامه‌نویسی JetBrains' CLion
- ۲ مراحل نصب
- ۷ اجرای (Run) اولین برنامه
- ۹ رفع خطا (Debug)
- ۱۲ ذخیره (Save)، بازیابی (Load) و بستن
- ۱۲ تحقیق کنید! (اختیاری)



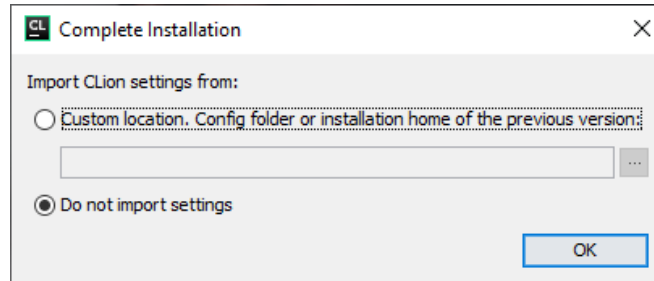
هدف این خودآموز آشنایی با محیط برنامه‌نویسی CLion از شرکت JetBrains است که در ادامه این درس به عنوان محیط استاندارد برنامه‌نویسی شما برای زبان C محسوب خواهد شد.

مراحل نصب

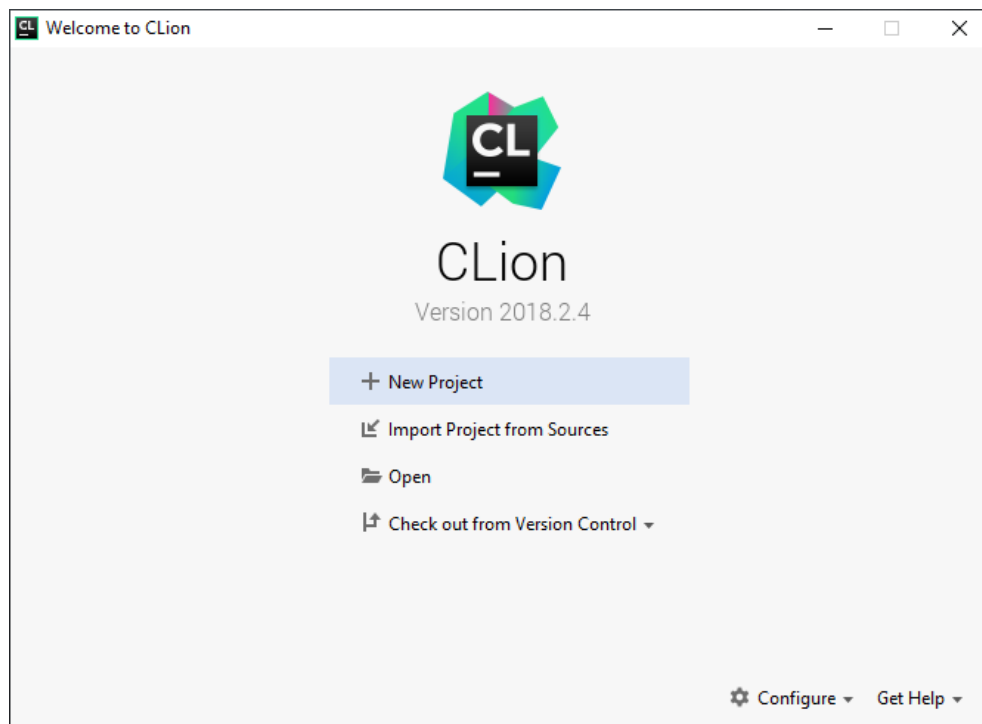
ابتدا محیط برنامه‌نویسی را نصب نمایید. می‌توانید این نرم‌افزار را از طریق لینک زیر دانلود کنید.

<https://www.jetbrains.com/clion/download/>

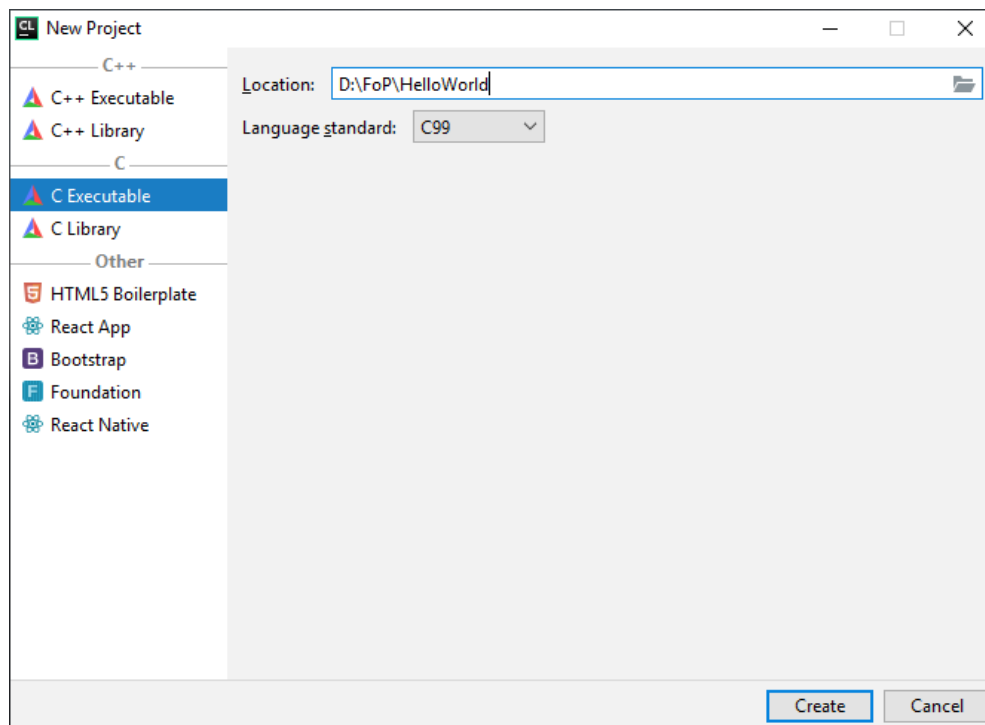
پس از نصب برنامه صفحه زیر ظاهر می‌شود. اگر این برنامه را برای بار اول نصب می‌کنید و نیازی به اضافه کردن تنظیمات نسخه قبلی را نداشتید از گزینه Do not import settings استفاده کنید. اما در صورتی که این نرم‌افزار را قبلاً نصب کرده بودید و خواستید از تنظیمات همان نسخه استفاده کنید، از گزینه Custom location... استفاده کنید.



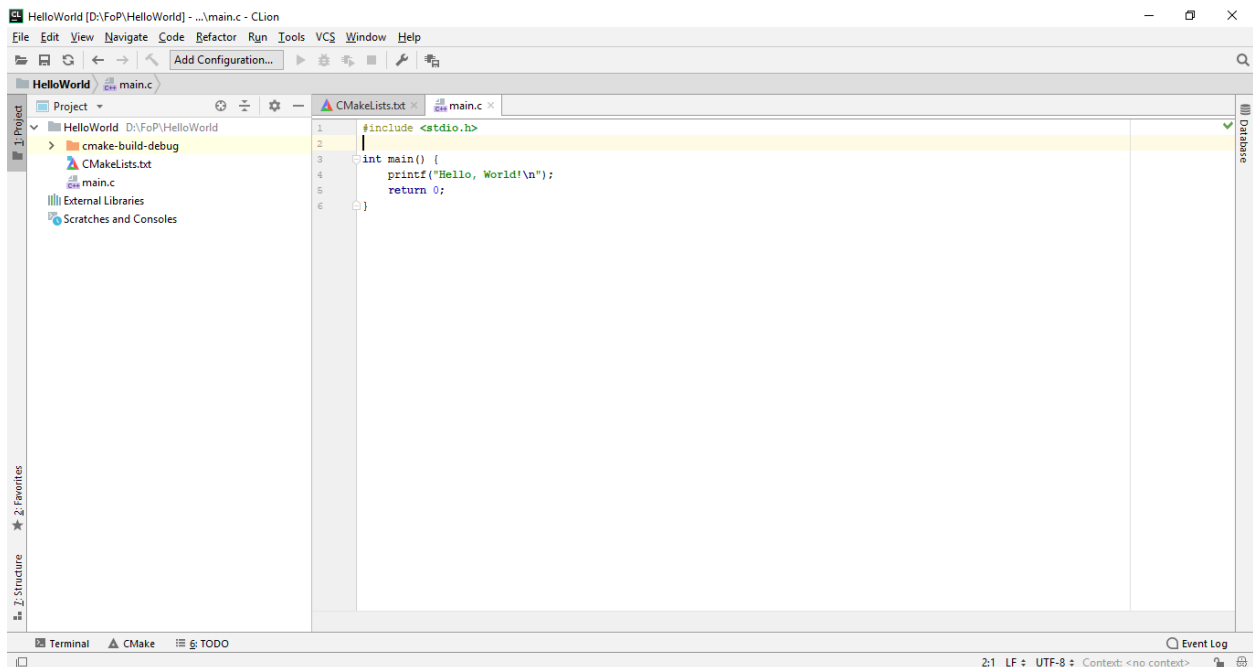
پس از این مرحله صفحه دیگری ظاهر می‌شود و می‌توانید تنظیمات دلخواه خود را انتخاب کنید. در ادامه می‌توانید در صفحه زیر پروژه جدید بسازید. برای این کار روی گزینه New Project کلیک کنید تا مراحل ایجاد پروژه جدید نمایش داده شود.



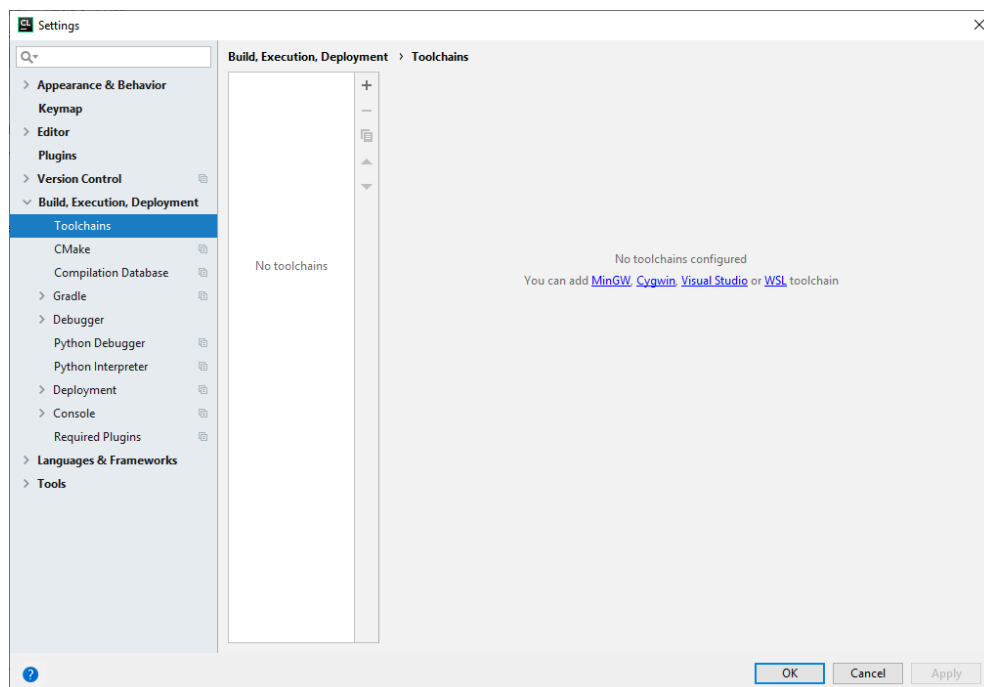
با کلیک روی گزینه New Project صفحه زیر ظاهر می‌شود. از قسمت سمت چپ C Executable را انتخاب کنید سپس در قسمت Location آدرس پروژه را تعیین کنید و با انتخاب گزینه Create پروژه را بسازید.



پنجره جدیدی به شکل زیر ایجاد می‌شود که می‌توان فرایندهای برنامه‌نویسی را در بخش‌های مختلف آن انجام داد. از این پس می‌توانید فرایند ایجاد پروژه را از مسیر **File | New Project...** انجام دهید. می‌توانید مسیر **View | Toolbar** را فعال کنید تا امکانات بیشتری مشاهده کنید.



در ابتدا وارد مسیر **File | Settings | Build, Execution, Deployment | Toolchains** شوید.



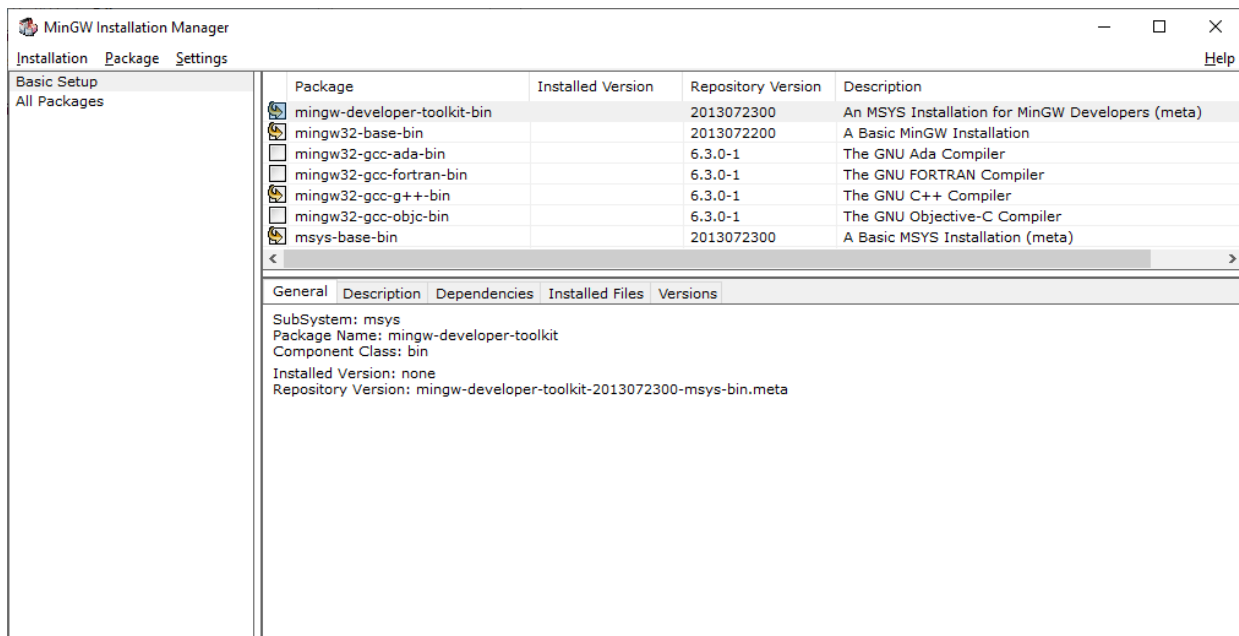
در صورتی که لیست خالی بود باید یک محیط اضافه کنید. برای این کار یک محیط MinGW اضافه خواهیم کرد. ابتدا به سایت <http://mingw.org> مراجعه کنید و از قسمت دانلود نرم‌افزار نصب را دانلود کنید. برای دانلود به لینک زیر خواهید رفت.

<https://osdn.net/projects/mingw/releases/>

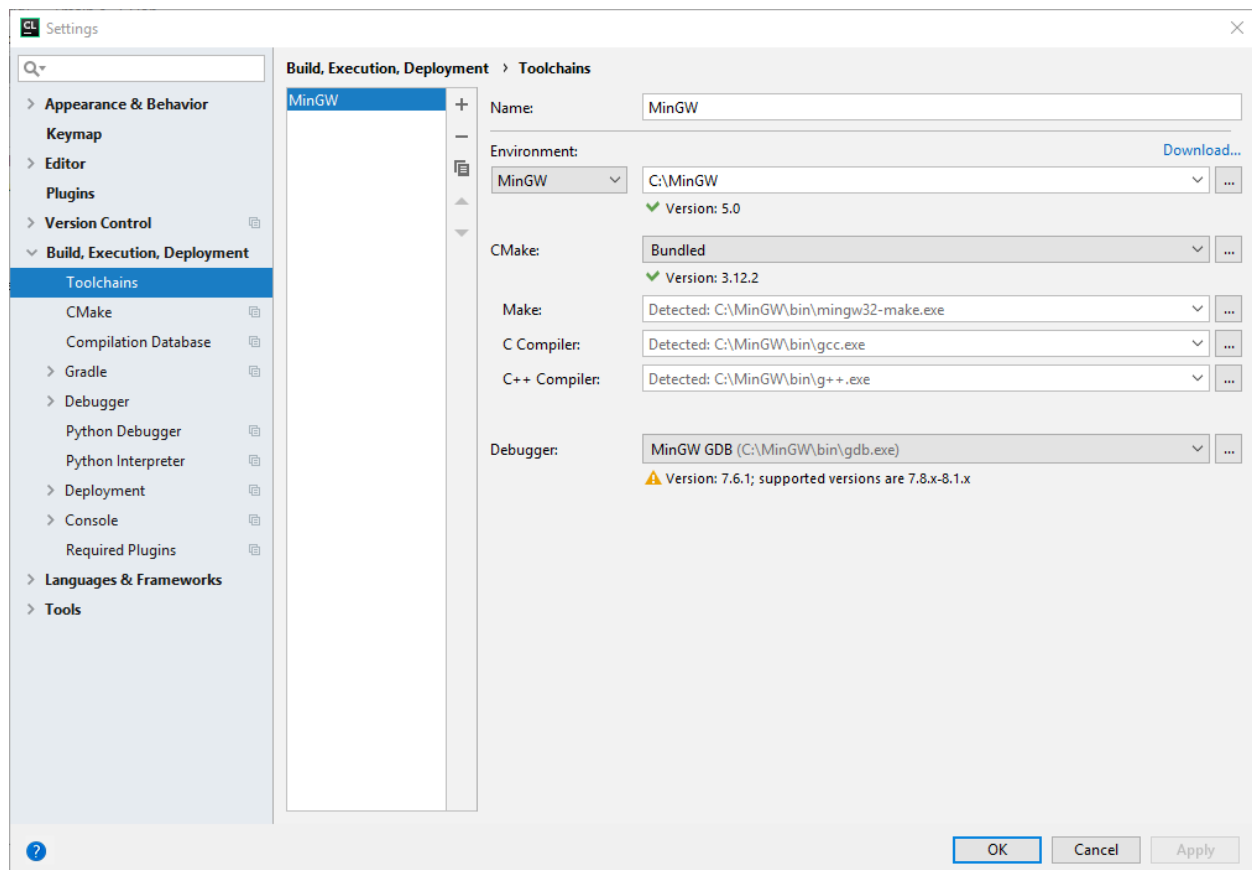
حال برنامه را نصب و اجرا کنید. با اجرای آن پنجره زیر نمایش داده می‌شود. موارد زیر را فعال کنید.

- mingw-developer-toolkit-bin
- msys-base-bin
- mingw32-base-bin
- mingw32-gcc-g++-bin

حال مسیر Installation | Apply Changes را اجرا کنید. با اجرای این مرحله محیط دانلود و نصب می‌شود.



حال دوباره به مسیر File | Settings | Build, Execution, Deployment | Toolchains بروید. با کلیک روی علامت مثبت یک محیط اضافه کنید. مورد MinGW را به عنوان نوع محیط انتخاب کنید. مسیر نصب را انتخاب کنید و منتظر بمانید تا همه موارد تایید شوند.




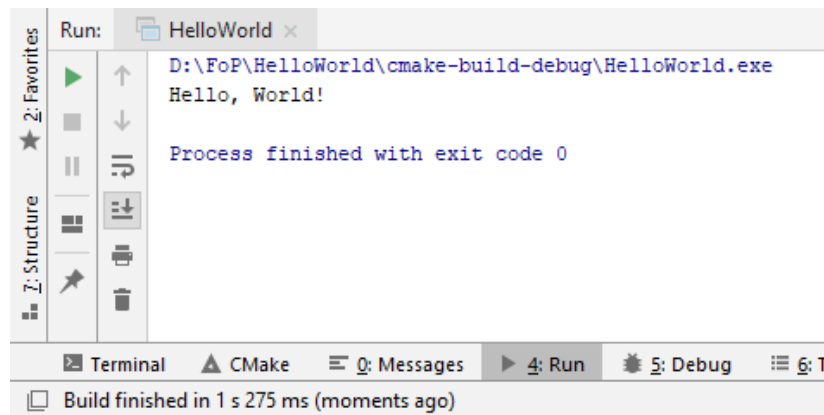
با مشاهده پیام‌های بالا محیط شما آماده می‌شود. گزینه OK را انتخاب و به صفحه اصلی بازگردید.

اجرای (Run) اولین برنامه

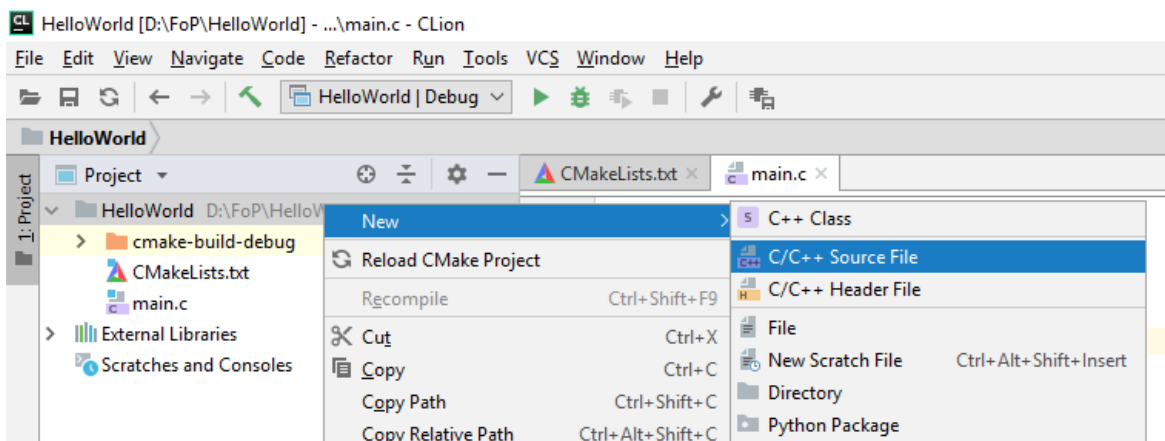
اولین برنامه‌ای که به اجرای آن خواهیم پرداخت! Hello World نام دارد. با ایجاد پروژه این برنامه به صورت پیش‌فرض وجود دارد.

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello, World!\n");
5     return 0;
6 }
```

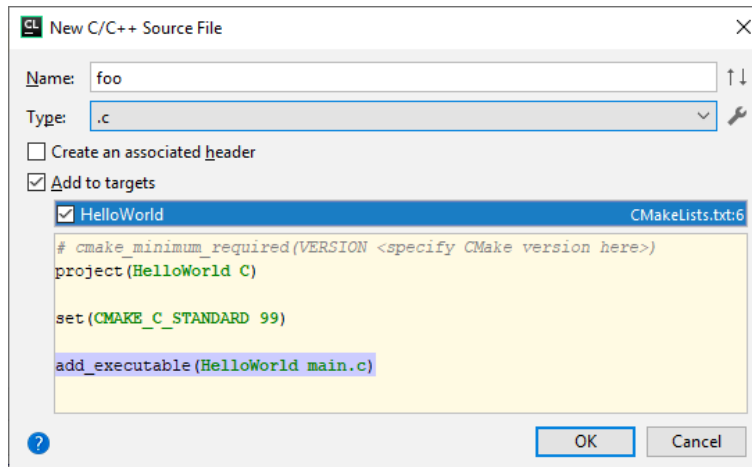
ابا کلیک روی دکمه  برنامه را Compile و اجرا کنید و در نتیجه باید بتوانید پنجره زیر را مشاهده کنید. می‌توانید از کلید میانبر Shift + F10 برای اجرای برنامه استفاده کنید.



برای ساختن فایل‌های بیشتر می‌توانید در پنجره Project روی پوشه مورد نظر کلیک راست کرده و مانند شکل زیر مسیر C/C++ | New | Source File را انتخاب کنید.




در پنجره جدیدی که باز می‌شود، نام مناسبی برای فایل اختصاص دهید و دقت کنید که برای گزینه Type حتما مورد c. را انتخاب کنید.



رفع خطا (Debug)

خطاهای برنامه‌نویسی موارد معمولی هستند که در فرایند برنامه‌نویسی رخ می‌دهند. روش‌های مختلفی برای آن‌ها وجود دارد. تصویر زیر نمونه‌ای از یک کد با خطای نحوی (Syntax Error) است.

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello, World!\n");
5     return 0;
6 }
```

همانطور که مشخص است خطای موجود در این خط با خط‌چین قرمز مشخص شده است. با کلیک روی  برنامه را می‌سازیم (Build) می‌کنیم). پیام زیر نمایش داده می‌شود و به این معنی است که برنامه build نشده است. برنامه همواره باید قبل از اجرا build شود و نرم‌افزار CLion این کار را برایمان انجام می‌دهد. دلیل استفاده از این دکمه در اینجا نمایش تفاوت خطای بالا و خطای بعدی است.



در این پیام خطی که خطا در آن رخ داده است و نوع خطایی که رخ داده است نمایش داده شده است. با رفع خطا می‌توان دوباره به ساخت برنامه پرداخت. اما کد زیر را در نظر بگیرید.

```
1 #include <stdio.h>
2
3 int main() {
4     int a = 1;
5     int b = 0;
6     int c = 0;
7     c = a / b;
8     printf("%d\n", c);
9     return 0;
10 }
```


این برنامه به ظاهر مشکلی ندارد و وقتی پروژه را می‌سازیم و پروژه با موفقیت ساخته می‌شود. اما وقتی برنامه را اجرا می‌کنیم، برنامه با موفقیت اجرا نمی‌شود. اگر برنامه با موفقیت اجرا شود باید مقدار c در خروجی چاپ شود ولی این مقدار در خروجی نمایش داده نمی‌شود. (نشانه دیگری نیز برای عدم اجرای موفق برنامه پیدا کنید!)



این دست از خطاها ممکن است همواره رخ ندهند! برای مثال اگر مقدار b از کاربر دریافت شود ممکن است بعضی وقت‌ها کاربر مقدار صفر و بعضی وقت‌ها مقداری غیر از صفر وارد کند. به همین دلیل پیدا کردن آن‌ها قدری سخت‌تر است. یک راه برای شروع رفع خطای این دست پرینت کردن مقادیر است تا خطاهایی را که ممکن است رخ دهد پیدا کنیم. اما راه آسان‌تر برای این کار استفاده از امکان `debugging` نرم‌افزار CLion است. سمت راست شماره خط در خط `int a = 1;` کلیک کنید تا یک Breakpoint ساخته شود و به شکل زیر تغییر یابد.

```

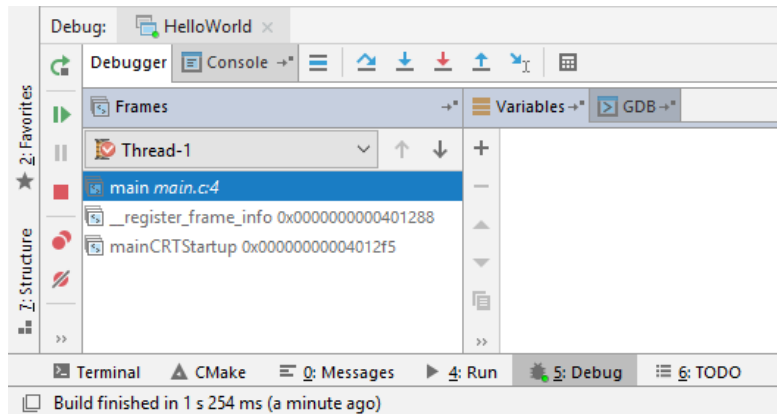
1      #include <stdio.h>
2
3      int main() {
4          int a = 1;
5          int b = 0;
6          int c = 0;
7          c = a / b;
8          printf("%d\n", c);
9          return 0;
10     }
```


حال با کلیک روی دکمه  برنامه را در حالت `debug` اجرا می‌کنیم.

```

1      #include <stdio.h>
2
3      int main() {
4          int a = 1;
5          int b = 0;
6          int c = 0;
7          c = a / b;
8          printf("%d\n", c);
9          return 0;
10     }
```

خط آبی نشان‌دهنده خطی که برنامه در حال حاضر تا آن اجرا شده است و خط بعدی که اجرا می‌شود است. پنجره زیر پنجره `Debug` است که دارای امکاناتی است که برای `debug` کردن مفید هستند.

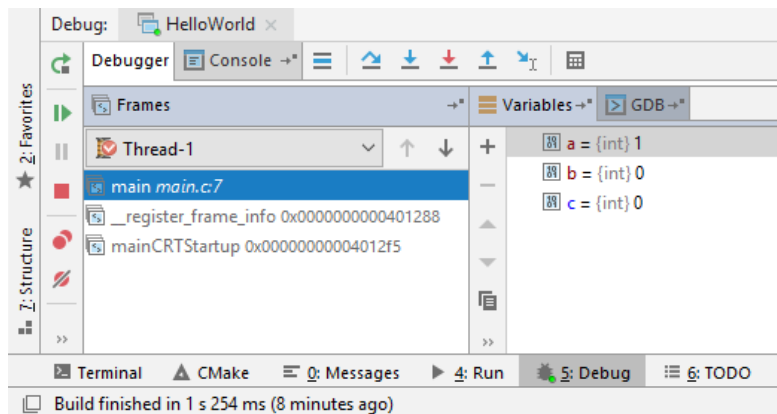


با کلیک روی دکمه  که نام Step Over دارد می‌توان خط مشخص شده را اجرا کرد. با دکمه F8 نیز می‌توان این کار را انجام داد. دو بار دیگر نیز F8 را می‌زنیم تا برنامه به حالت زیر برود (دکمه‌های دیگر پس از یادگیری توابع (function) استفاده خواهند شد).

```

1      #include <stdio.h>
2
3      int main() {
4          int a = 1; a: 1
5          int b = 0; b: 0
6          int c = 0; c: 0
7          c = a / b;
8          printf("%d\n", c);
9          return 0;
10     }
```

با رسیدن به این خط پنجره Debug به حالت زیر می‌رسد.



در قسمت Variables متغیرهای تعریف شده و مقادیر فعلیشان نمایش داده می‌شود. می‌توان با استفاده از کلید جمع متغیرهای دیگر یا ترکیبی از آن‌ها را اضافه کرد. حال مشخص است که در خط بعدی باید a / b محاسبه شود و از طرفی با توجه به قسمت Variables مقدار b برابر صفر است؛ پس تقسیم بر صفر رخ می‌دهد که این تقسیم تعریف نشده است و باعث می‌شود اجرای برنامه با مشکل روبه‌رو شود. حال با تغییر مقدار a به ۴ و b به ۲ برنامه را دوباره اجرا می‌کنیم و نتیجه را مشاهده می‌کنیم.

ذخیره (Save)، بازیابی (Load) و بستن

با مسیر File | Save All یا کلید ترکیبی Ctrl + S می‌توانید تغییرات خود را ذخیره کنید. با مسیر File | Open... می‌توانید پروژه دیگری را باز کنید. با مسیر File | Open Recent می‌توانید پروژه‌های قبلی خود را باز کنید. در نهایت می‌توانید با مسیر File | Close Project به صفحه اول برگردید یا با File | Exit نرم‌افزار CLion را ببندید.

تحقیق کنید! (اختیاری)

- در رابطه با کلمات زیر تحقیق کنید.

- IDE
- Compiler
- Editor
- Debugger

- در رابطه با کامپایل، ویرایش و دیباگ در محیط لینوکس تحقیق کنید. از کلیدواژه‌های زیر نیز استفاده کنید.

- gcc
- vim
- gdb

- شاید برایتان جالب باشد که بدانید کلمه bug از کجا آمده است و چرا موقع رفع ایرادهای برنامه از debugger (سوسک‌کش!) استفاده می‌کنیم.

موفق باشید