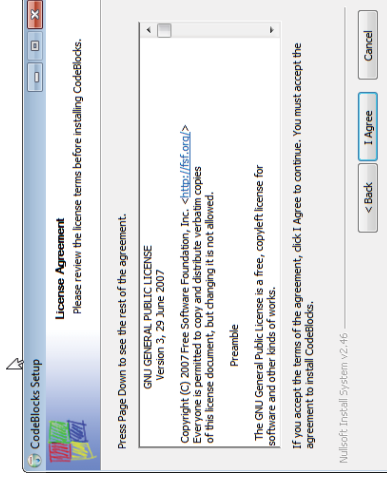
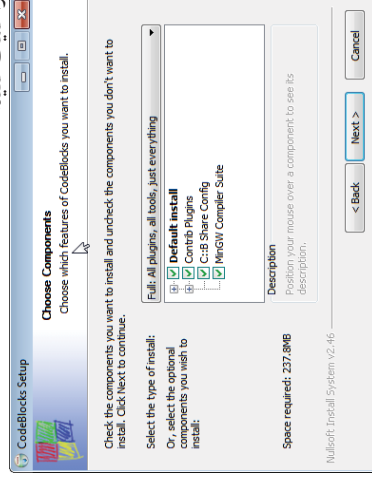
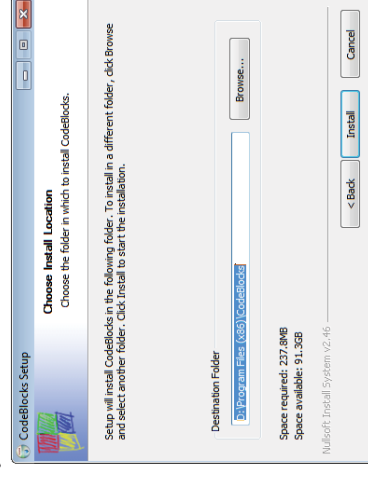




۳. کلید Next را کلیک کنید.



۴. در این مرحله می‌توانید محل نصب را انتخاب کنید و سپس کلید Install را فشار دهید.



راهنمای نصب کامپایلر و اجرای برنامه‌های C++

۱.۲ راهنمای نصب کامپایلر CodeBlocks

اگر کاربر ویندوز هستید، ابتدا به یکی از آدرس بروید و فایل نصب را بارگذاری کنید.

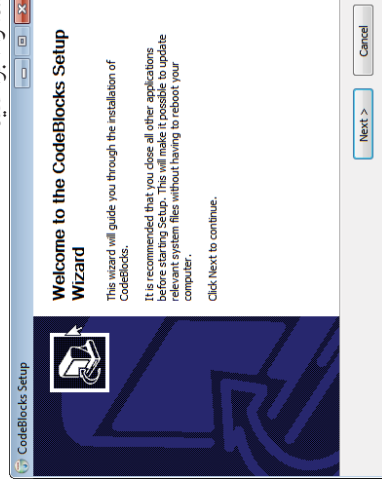
<http://sourceforge.net/projects/codeblocks/files/Binaries/13.12/Windows/codeblocks-13.12mingw-setup.exe/download>

یا

<http://prdownload.berlios.de/codeblocks/codeblocks-13.12-setup.exe>

برای نصب مراحل زیر را طی کنید:

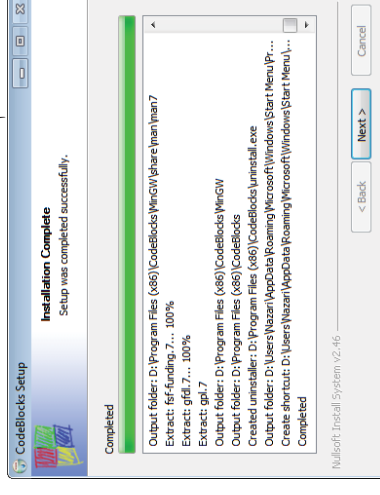
۱. فایل دانلود شده را اجرا کنید.



کلید Next را کلیک کنید.

۲. روی I Agree کلیک کنید.

۵. منظر بهمانید تا نصب انجام شود.



پوست آ. راهنمای نصب کامپایلر و اجرای برنامه‌های C++

۱۵۶

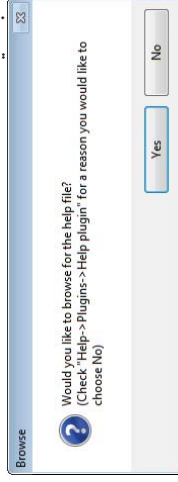
۱۵۵

نصب و تنظیم *HELP* برای *CODEBLOCKS*

۴. در قدم بعد در CodeBlocks از منوی Settings گزینه Environment را انتخاب کنید.



۷. با زدن کلید OK در مرحله بعد برای انتخاب مسیر فایل Help در پنجره نمایش داده شده Yes را انتخاب کنید.



۸. مسیر فایل Help را تعیین کرده و فایل را انتخاب کنید.



۹. به منظور تعیین کلید F۱ برای باز کردن این فایل جهت Help نرم‌افزار گزینه

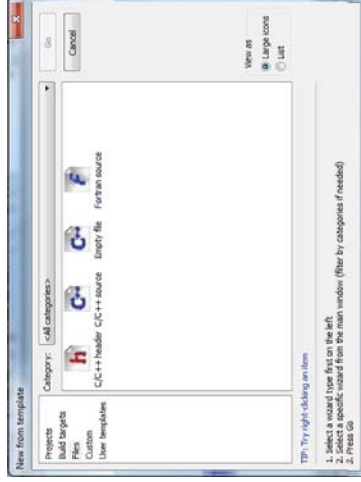
This is the default help file (shortcut: F1)

را تیک بزنید و سپس OK را انتخاب کنید.

۵. در سمت چپ پنجره باز شده Help file را انتخاب کنید.

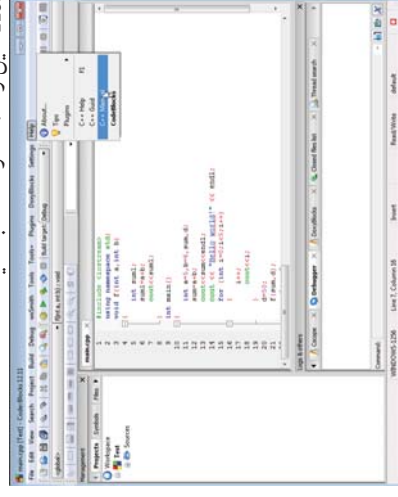


۶. کلید Add را فشار داده و در قسمت اضافه کردن title نام دلخواهی را تایپ کنید.



۱. تنظیمات به پایان رسید. برای باز کردن این فایل Help کافست روی کلمه مورد نظر در محیط نرم افزار کلیک کرده و کلید F۱ را فشار دهید.

البته می‌توانید بیش از یک فایل را به همین روش جهت Help نرم افزار تنظیم کنید که در این حالت برای یکی از فایل ها کلید F۱ را به عنوان کلید میانبر انتخاب کنید. برای دیدن بقیه فایل ها می‌توانید از منوی Help، این راهنماها را انتخاب کنید.

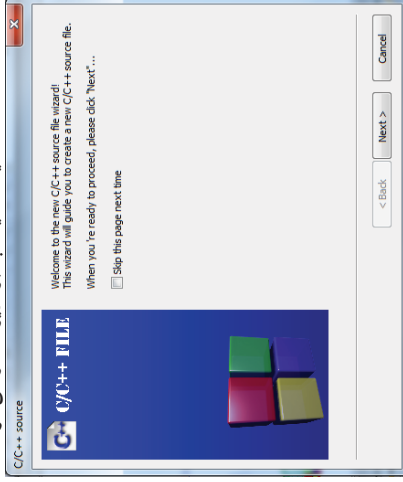


آ. شیوه ورود برنامه و اجرای آن

آ. نوشتن برنامه در CodeBlocks (روش اول)

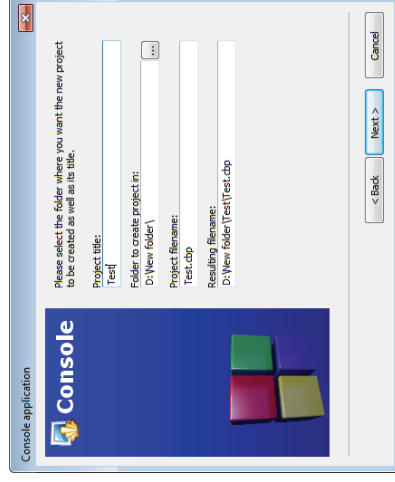
۱. از منوی File گزینه New و سپس File را انتخاب کنید. پنجره زیر ظاهر می‌شود.

۲. روی آیکن C/C++ Source کلیک کنید. پنجره زیر ظاهر می‌شود.

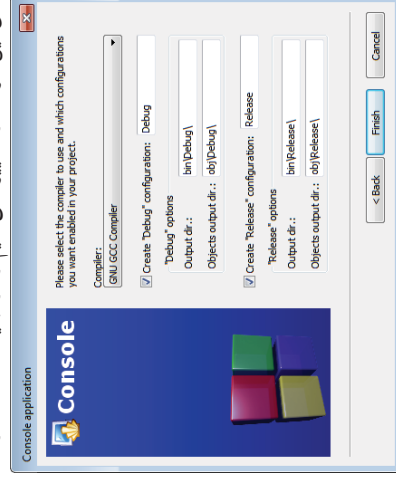


۳. روی Next کلیک کنید. پنجره زیر ظاهر می‌شود.

۵. در این مرحله نام و محل ذخیره پروژه را مشخص کنید. در قسمت Project title نام و در قسمت Project in Folder to create project را مشخص کنید. محل ذخیره را تعیین کنید و سپس Next را فشار دهید.



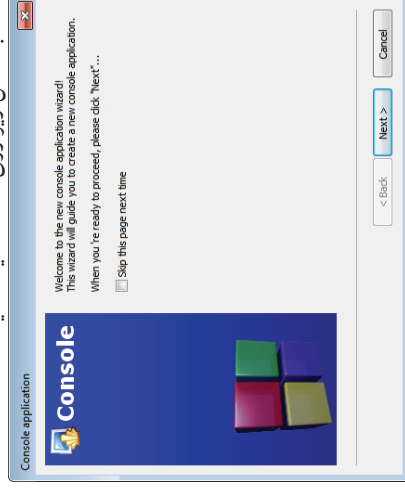
۶. پیش‌فروش‌های این مرحله را تغییر نمی‌دهیم و در نهایت Finish را فشار می‌دهیم.



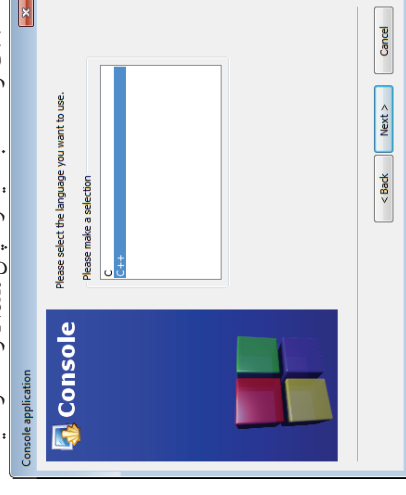
نرم‌افزار محیط برنامه‌نویسی جدیدی را برای شما باز خواهد کرد.

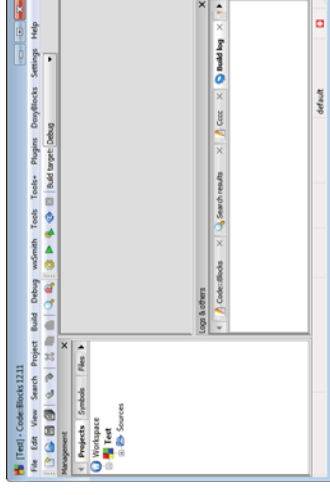


۳. در پنجره باز شده به شکل زیر روی کلیک کنید.



۴. در کاردر باز شده C++ را انتخاب کنید و سپس Next را فشار دهید.



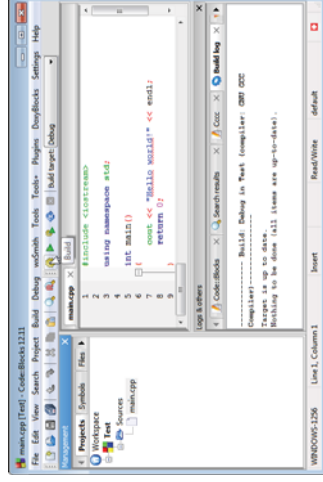


بعد از نوشتن برنامه ابتدا باید آن را Build کنید. منظور از Build، ترجمه برنامه به زبان ماشین است. برای اجرای آن، لازم است ابتدا برنامه به زبان ماشین تبدیل شود. روش این دو کار در قسمت بعد آمده است.

۳.۳.۲ روش ترجمه و اجرای برنامه

برای این کار می‌توانید از سه روش استفاده کنید:

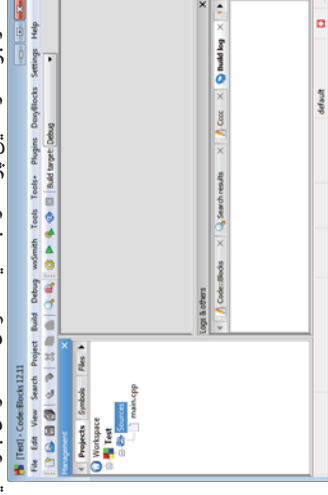
۱. از منوی Build اولین گزینه را اجرا کنید.
۲. همچنین می‌توانید با استفاده از کلیدهای ترکیبی Ctrl+F9 این کار را انجام دهید.
۳. گزینه Build را در بالای صفحه کلیک کنید. اکنون این کار یک جرخ دنده زرد رنگ است.



برای اجرای برنامه نیاز است که برنامه run شود که برای این کار نیز می‌توان به سه روش عمل کرد:

۱. از منوی Build گزینه run را انتخاب کنید.

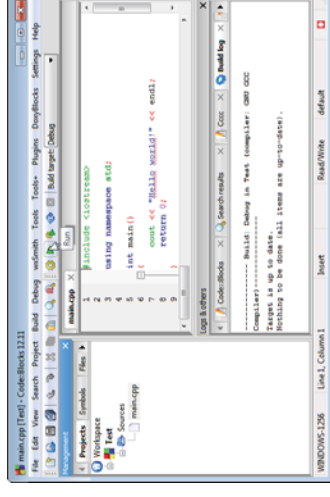
در سمت چپ محیط این نرم‌افزار، پنل Management را مشاهده می‌کنید. اولین گزینه در این پنل Project هست. در این قسمت می‌توانید نام پروژه خود را ببینید و در پایین آن پوشه Sources وجود دارد. این پوشه را با کلیک کردن + کنار آن باز کنید.



در این پوشه فایل main.cpp وجود دارد. این فایل در زمان ایجاد برنامه به طور خودکار ساخته می‌شود.

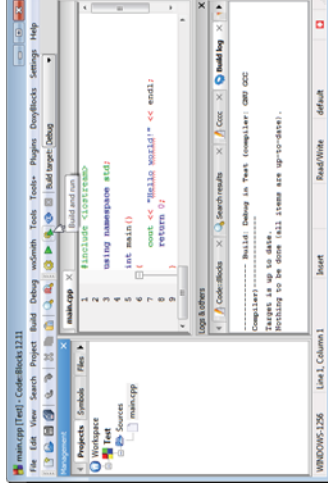
با کلیک کردن روی این فایل و باز کردن آن، محیط برنامه‌نویسی برای شما در سمت راست باز می‌شود. دستورهای پیش فرضی در این محیط وجود دارد.

۲. از کلیدهای ترکیبی $Ctrl+F10$ استفاده کنید.
۳. گزینه Run را در بالای صفحه اجرا کنید. اکنون اجرا مثلث سبز رنگ است.

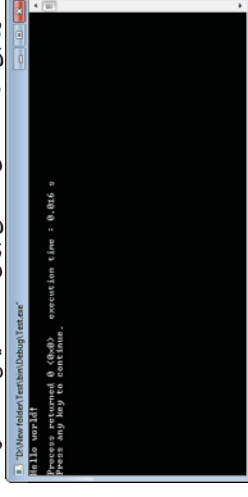


البته می‌توانید اعمال Build و Run را با هم اجرا کنید. که برای این کار نیز می‌توان از سه طریق عمل کرد:

۱. از منوی run and build را انتخاب کنید.
۲. از کلید $F9$ استفاده کنید.
۳. گزینه Build and run را در بالای صفحه اجرا کنید.



با اجرای برنامه خروجی در صفحه‌ای مشکلی برای شما نمایش داده خواهد شد.



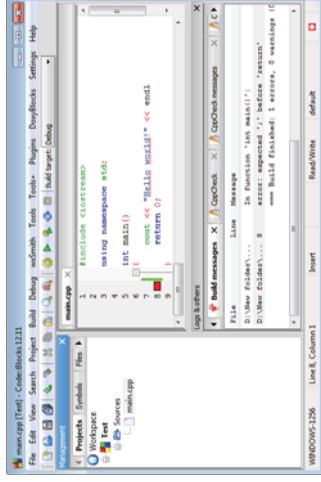
۴.آ خطاگیری- گرامری و منطقی

خطاهای برنامه را می‌توان به دو دسته تقسیم کرد:

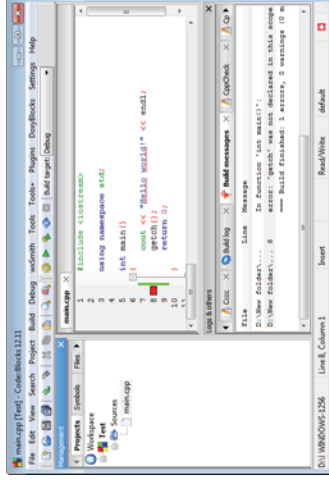
۱. خطای گرامری
۲. خطای منطقی

۱.۴.آ خطای گرامری:

این دسته از خطاها به خطاهایی مربوط می‌شود که معمولاً کاربران مبتدی در نوشتن دستورات مرکب می‌شوند و دستورات را به شکل صحیح نمی‌نویسند. بیشترین خطا در این زمینه فراموش کردن نوشتن ؛ (سمی کالن) است. می‌توانید نوع خطا و محل خطا را در پایین صفحه در قسمت Build messages مشاهده کنید. در محیط نرم‌افزار نیز یک مربع قرمز ابتدای خطی که در آن خطا وجود دارد، ظاهر می‌شود.



البته اگر خطا به نوشتن ؛ مربوط باشد نرم‌افزار شماره خطا را خط بعد مشخص می‌کند. دسته دیگر خطاهای این نوع، نوشتن فایل‌های کتابخانه‌ای توابعی است که در برنامه استفاده شده است.



۲.۴.۲ خطاهای منطقی

منظور از این نوع، خطاهایی است که در اجرای برنامه مشکلی ایجاد نمی‌کند و خروجی برنامه نیز ظاهر می‌شود اما جواب‌های درستی به دست نمی‌دهد.

تشخیص این خطا به مراتب سخت‌تر از خطای نوع اول است.

با محدود کردن اجرای برنامه و مشاهده مقادیر متغیرهای تعریف شده می‌توانید خطا را شناسایی کنید.

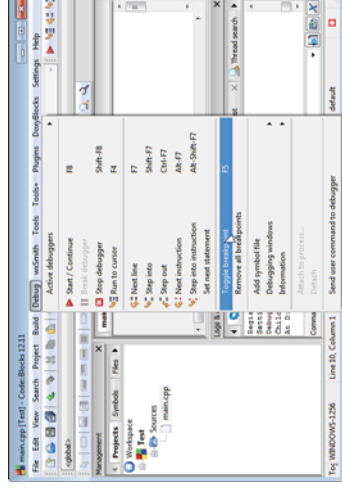
برای دیدن مقدار متغیرها از منوی Debug و زیر منوی Debugging windows گزینه Watches را انتخاب کنید. در پنجره باز شده نام متغیرها را تایپ کنید و سپس با محدود کردن اجرا می‌توانید مقادیر این متغیرها را مشاهده کنید و به این ترتیب محل بروز خطا را تشخیص دهید.

در قدم بعد باید محدوده اجرای برنامه را مشخص کرد. به این منظور از Watchpoint استفاده می‌شود. زمانی که برای دسترسی Breakpoint مشخص شود برنامه با رسیدن به این دستور مکث می‌کند و وارد محیط دیباگ می‌شود. سپس می‌توان از دستورات Step و سایر موارد استفاده کرد. برای تعیین Breakpoint یک دستور می‌توان از روش‌های زیر استفاده کرد:

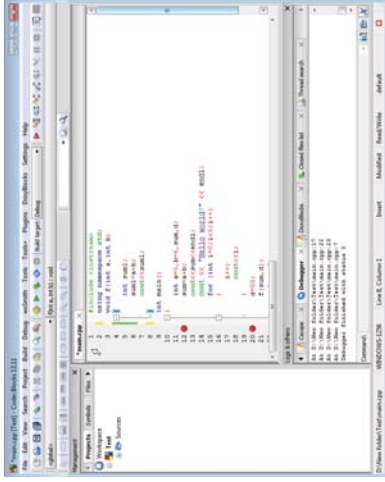
۱. بر روی خط دستور راست کلیک کرده و گزینه breakpoint را انتخاب کنید.

۲. بر روی خط دستور کلیک کرده و از منوی Debug گزینه breakpoint را انتخاب کنید.

۳. بر روی خط دستور کلیک کرده و کلید F۵ را فشار دهید.



در ابتدای خطی که برای آن Breakpoint تعیین می‌کنید یک دایره قرمز قرار می‌گیرد.



بعد از تعیین Breakpoint ها برای اجرای برنامه می‌توانید از منوی Debug گزینه Start/Continue را انتخاب کنید یا از کلید F۸ استفاده کنید. برنامه تا رسیدن به اولین Breakpoint اجرا می‌شود و صفحه خروجی نمایش داده می‌شود.
در ادامه موارد زیر به شما این امکان را می‌دهند که مراحل پیشروی دستورات را همان گونه که تمایل دارید انتخاب کنید:

۱. Next line: این گزینه برنامه را خط به خط اجرا می‌کند. می‌توانید از منوی Debug انتخاب

کنید یا کلید F۷ را فشار دهید

۲. Step into: عملکرد این گزینه شبیه Next line هست با این تفاوت که در Next line زمانی که با دستور فراخوانی یک تابع روبرو شود، وارد آن تابع نمی‌شود و مکان نما به خط بعدی منتقل می‌شود. اما در Step into اجرای برنامه به آن تابع منتقل می‌شود و مکان نما ابتدای اولین دستور تابع قرار می‌گیرد و بعد از اجرای کامل آن تابع به تابع فراخوانی کننده بر می‌گردد. این گزینه را می‌توانید از منوی Debug انتخاب کنید یا از کلیدهای ترکیبی Shift+F7 استفاده کنید.

۳. Step out: این گزینه از جهت منطقی عملکردی بر عکس Step into دارد. اگر اجرای Step into یا دلیل دیگری وارد یک تابع شده‌اید می‌توانید با اجرای Step out آن خارج شده و مکان نما به تابع فراخوانی کننده منتقل می‌شود. این گزینه را می‌توانید از منوی Debug انتخاب کنید یا از کلیدهای ترکیبی Ctrl+F7 استفاده کنید.

برای حذف کردن Breakpoint می‌توان روی آن کلیک کرد. همچنین برای حذف همه Breakpoint می‌توان از منوی Debug گزینه Remove all breakpoints را انتخاب کرد.

گزینه دیگری غیر از Breakpoint نیز برای محدود کردن اجرا وجود دارد و آن Run to cursor

هست.

ابتدا مکان نما را روی خطی که قصد دارید عمل اجرا تا آن مکان انجام شود قرار دهید. سپس از منوی Debug گزینه Run to cursor را انتخاب کنید یا کلید F۴ را فشار دهید. در آغاز خط مشخص شده، مثلث زرد رنگی قرار می‌گیرد و برنامه بلافاصله تا این دستور اجرا شده و سپس مکث می‌کند.