

Appendix

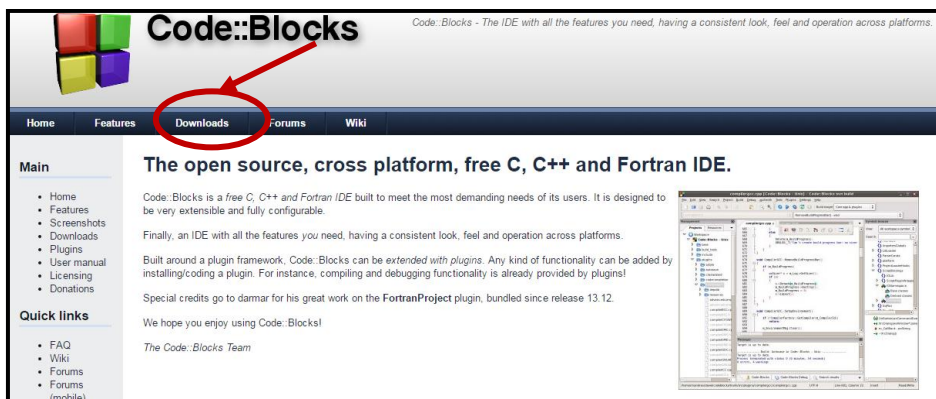
CodeBlocks Getting Started

Appendix: CodeBlocks Getting Started

▪ **How to Download "CodeBlocks":**

For developing any c language application, we need to use some program to write and compile c code statements and that's what we call IDE programs (Integrated Development Environment) in programming life, the most common IDE for c language is "**CodeBlocks**". To get code blocks IDE on your machine just follow the steps:

1. Go online for: www.codeblocks.org



2. Click on "Downloads" Tab

3. Click "Download the binary release":



Appendix: CodeBlocks Getting Started

4. Choose "Windows" then for "codeblocks-16.01mingw-setup.exe"

Click on "SourceForge.net" Installation

Please select a setup package depending on your platform:

- Windows XP / Vista / 7 / 8.x / 10
- Linux 32-bit
- Linux 64-bit
- Mac OS X

NOTE: For older OS'es use older releases. There are releases for many OS version and platforms on the [Sourceforge.net](#) page.

NOTE: There are also more recent *nightly builds* available in the [forums](#) or (for Debian and Fedora users) in Jens' [Debian repository](#) and [Jens' Fedora repository](#). Please note that we consider nightly builds to be *stable*, usually.

NOTE: We have a [Changelog for 16.01](#), that gives you an overview over the enhancements and fixes we have put in the new release.

 Windows XP / Vista / 7 / 8.x / 10:

File	Date	Download from
codeblocks-16.01-setup.exe	28 Jan 2016	Sourceforge.net or FossHub
codeblocks-16.01-setup-nonadmin.exe	28 Jan 2016	Sourceforge.net or FossHub
codeblocks-16.01-nosetup.zip	28 Jan 2016	Sourceforge.net or FossHub
codeblocks-16.01mingw-setup.exe	28 Jan 2016	Sourceforge.net or FossHub
codeblocks-16.01mingw-nosetup.zip	28 Jan 2016	Source[SF.NET download]Hub
codeblocks-16.01mingw_fortran-setup.exe	28 Jan 2016	Sourceforge.net or FossHub

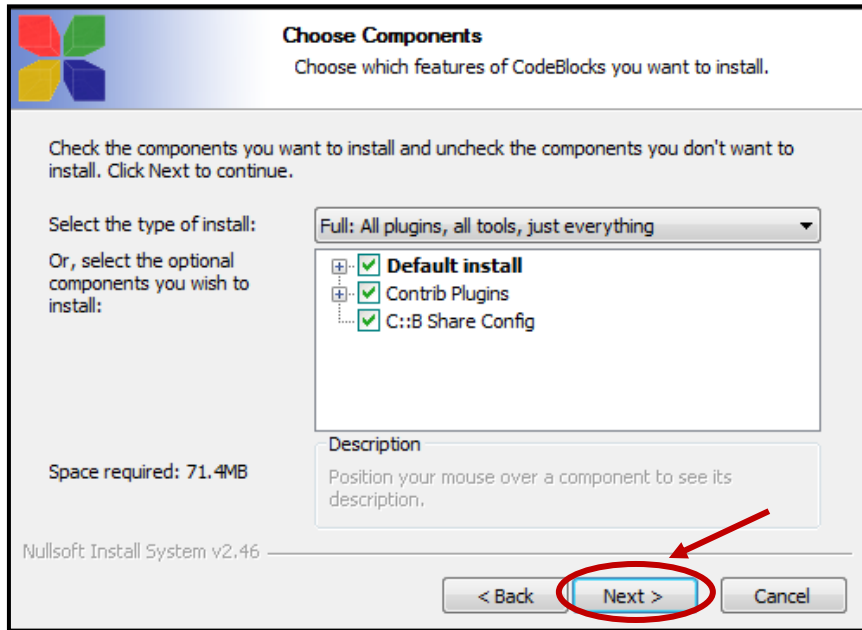
■ Installation of "CodeBlocks":

1. Double click on the ".exe file" of CodeBlocks that you have downloaded:

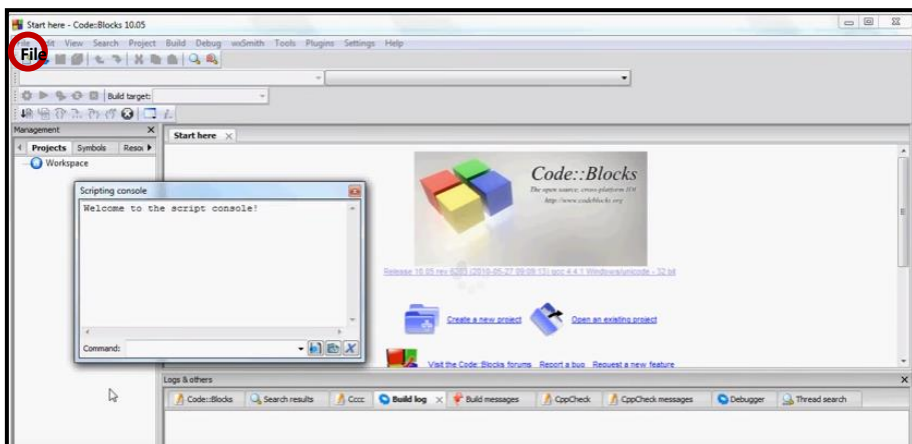


Appendix: CodeBlocks Getting Started

2. You Will get Installation windows, so Press Next --> I agree --> Next --> Install:

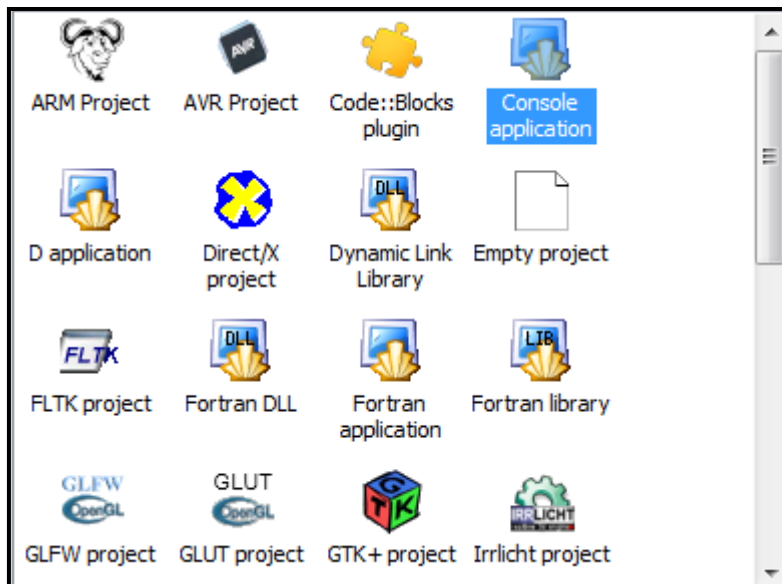


- Your first c project on "CodeBlocks":
 1. Double click on "CodeBlocks" icon to run the program:

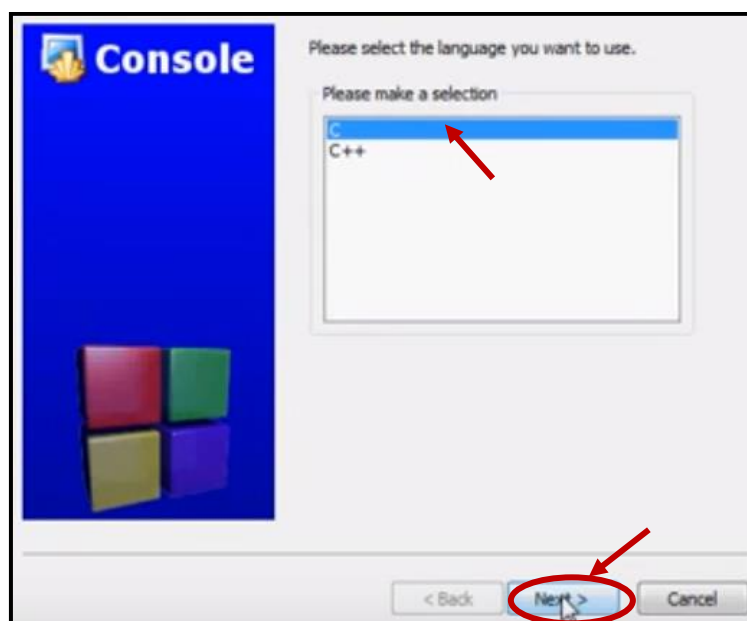


Appendix:
CodeBlocks Getting Started

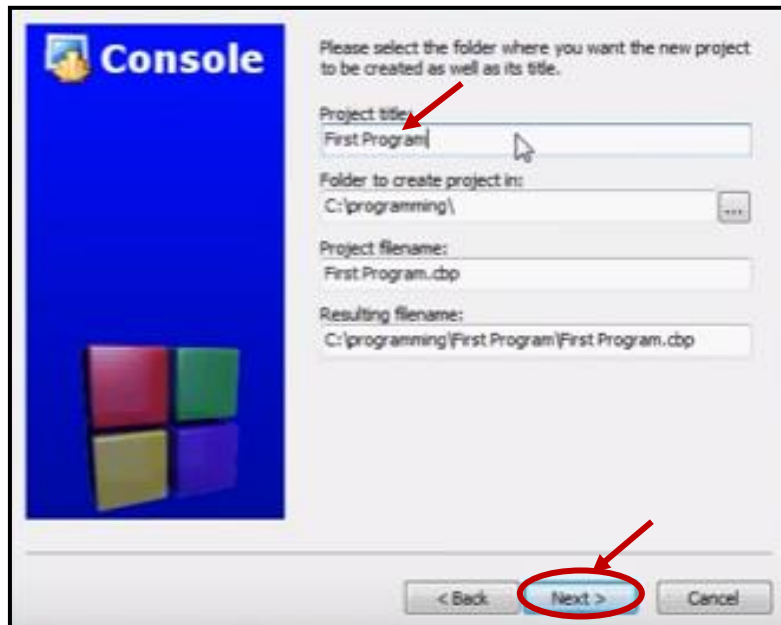
2. from "File" menu --> new --> Console application



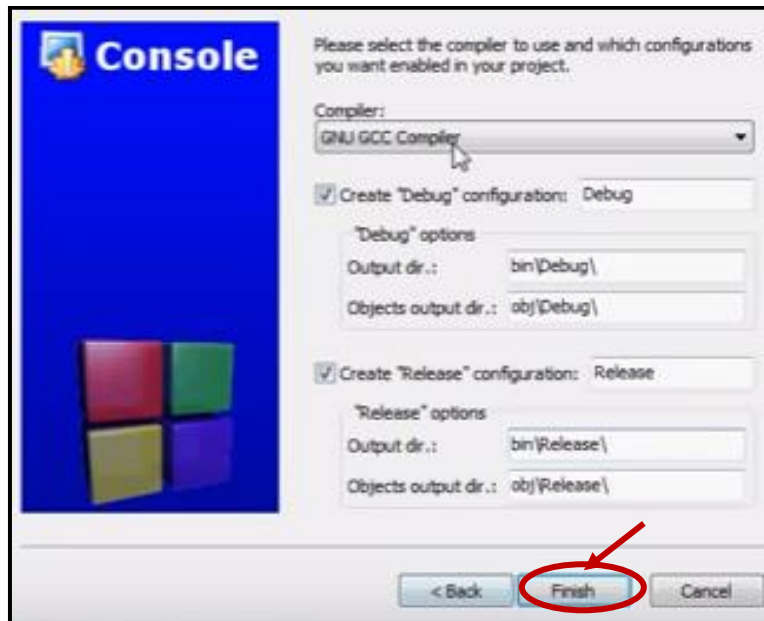
3. Select C --> Click Next:



4. Type your "Project title" as first Program or whatever you want:

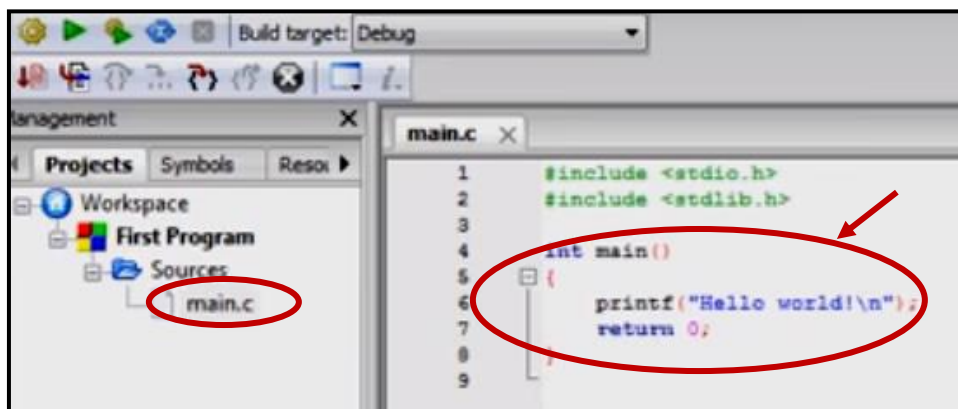


5. Just Click on Finish:



6. Write the following statement on "main.c" file:

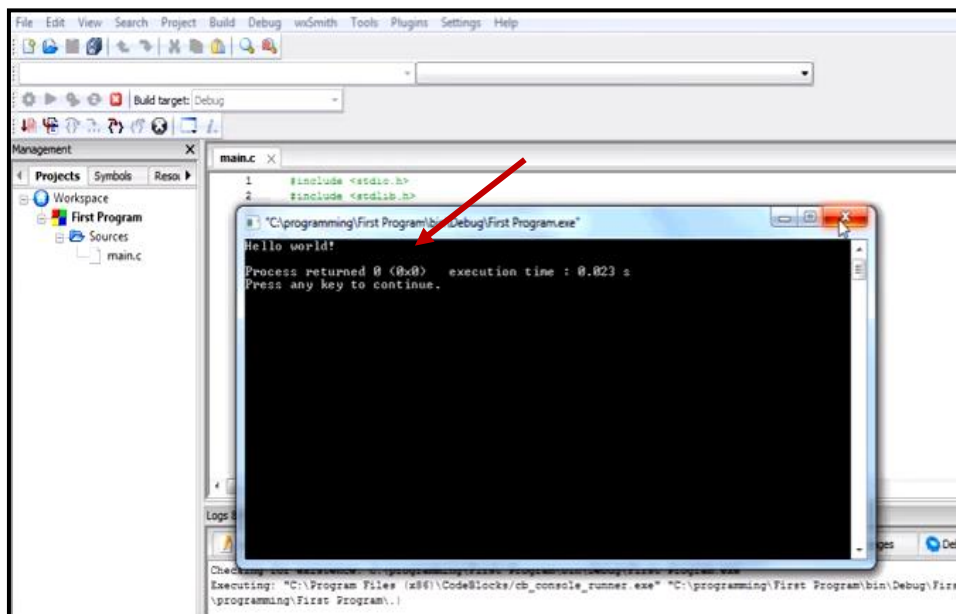
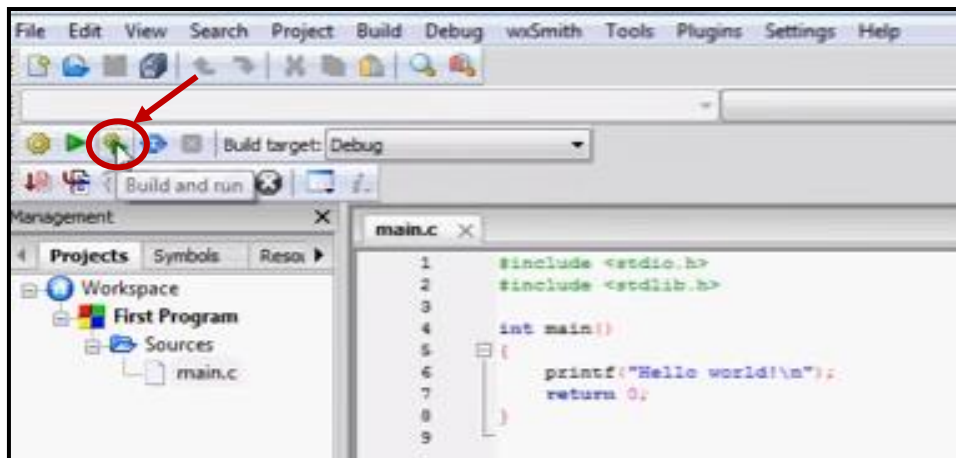
```
printf("Hello world!\n");
```



Appendix: CodeBlocks Getting Started

****NOTE: This statement is just to print "Hello World!" as "output" to the user screen followed by an empty "new line" as "\n" is for starting a new line.**

7. Now Click on "Build & Run" to get the program "output" to the console screen:



- All C programs are divided into units called Functions.
- A function is similar to subroutine in BASIC or a function in PASCAL
- Note that **main** is a function
- Every C program consists of one or more functions — this program has only one
- No matter how many functions there are in a C program, `main( )` is the one to which control is passed from the operating system when the program is run — it is the first function executed
- The word "Void" preceding "main" specifies that the function `main( )` will not return a value. The second "void," in parentheses, specifies that the function takes no arguments
- The whitespace characters (space, tab, newline) are invisible to the compiler

LAB 01

Write and execute the following file

```
Void main (void){printf("I am Fashel");}
```

- Indentation of blocks of code enclosed in braces is an important aspect of making C programs readable
- The program line `printf("I am Fashel");` causes the phrase in quotes to be printed on the screen
- The word **printf** is actually a function name
- Since **printf** is a function, it is followed by **parentheses**
- The phrase is a function **argument** — information passed from `main( )` to the function `printf( )`
- The parentheses following the name just to make it clear that we are talking about a function and not a variable or something else
- "I am Fashel" is a string constant — string constant is surrounded by quotes; this is how the compiler recognizes a string
- Notice that we have used the function `printf` in our program, but there is no code for this function in our program. Where is `printf` stored? It is in `stdio.h` file — standard input output header file
 - When compiling the program, the compiler realizes that `printf` is not a function included in your own program source file, so it leaves a message to that effect for the linker
- C distinguishes between uppercase and lowercase

- Exploring the **printf** function

Printf

```
Void main(void)
```

```
{
```

```
    Printf("this is the number two:%d",2)
```

```
}
```

- Can you guess what message will be printed when you run it

```
this is the number two:2
```

- The function `Printf()` can be given more than one argument
- These two arguments are separated by a comma
 - `"this is the number two:%d"`
 - `2`
- The `Printf()` function takes the value on the right of the comma and plugs it into the string on the left – in place of **format specifier**
- Why not simply put the number 2 into the original string?
 - `Printf("this is the number two:%d",2)`

- It wouldn't make any difference in what is printed out, since 2 is a constant
- Variables can also be used in the `Printf( )` function giving it the capability of changing what it prints while the program is running