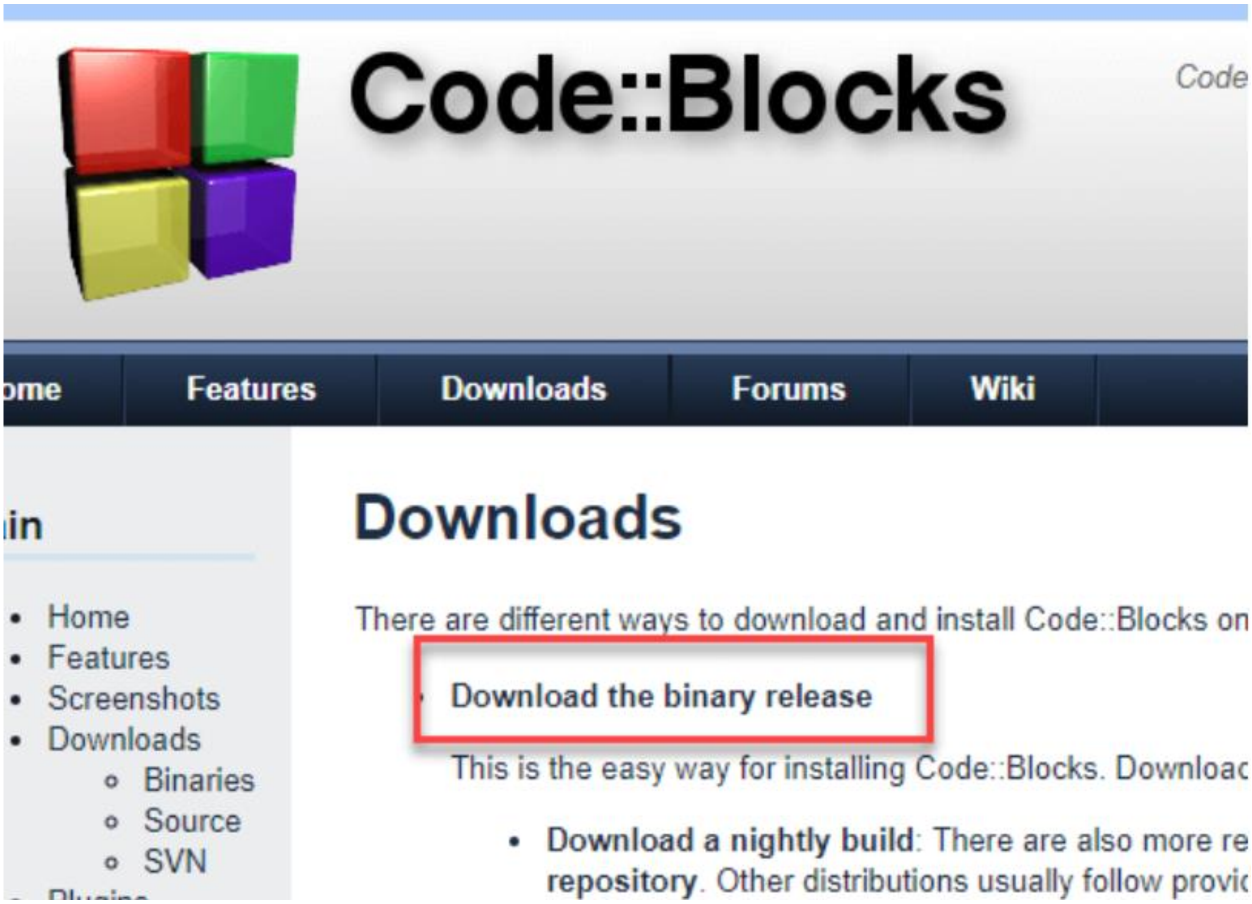
Practical Guide

Programming Fundementals

How to set up C Environment on your windows device

It is recommended to an open-source Integrated Development environment named **Code::Blocks.** This has a compiler, editor and debugger in one package bundled. You are required to go through following step to set up C environment on your device.



Downloads

There are different ways to download and install Code::Blocks on

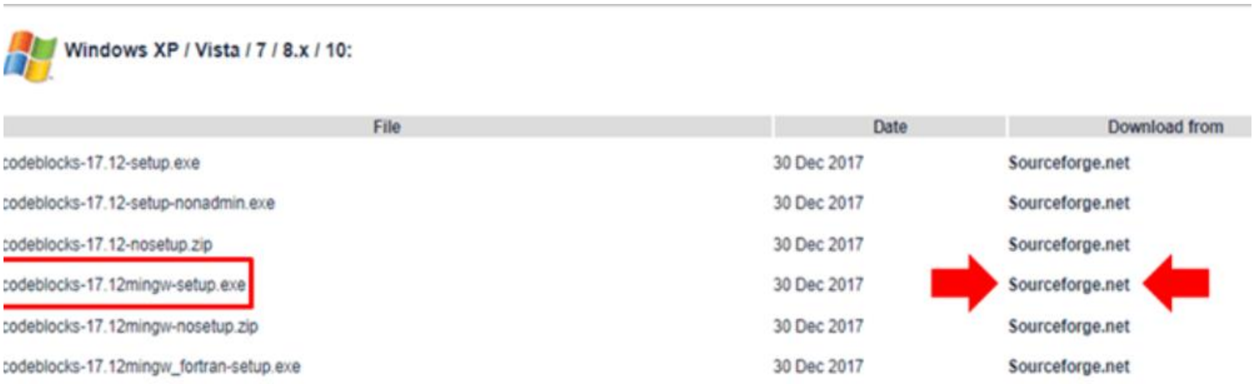
- Download the binary release

  This is the easy way for installing Code::Blocks. Downloac

  - **Download a nightly build**: There are also more re **repository**. Other distributions usually follow provic

Step 1 : Go to http://www.codeblocks.org/downloads.

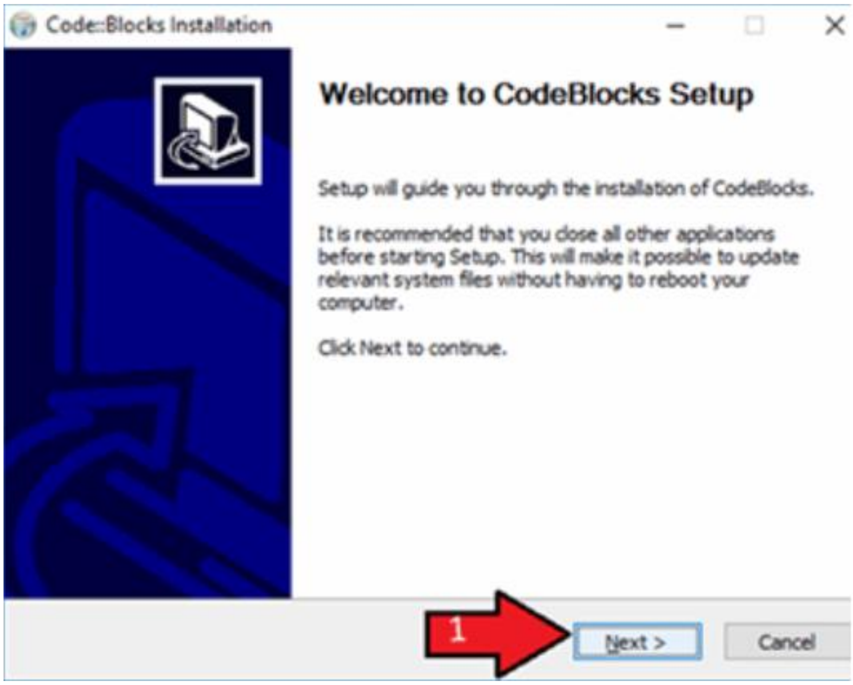
Step 2 : Select the installer with GCC Compiler, e.g., codeblocks-17.12mingw-setup.exe which includes **MinGW's GNU GCC compiler** and **GNU GDB debugger** with Code: Blocks source files.



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

File	Date	Download from
codeblocks-17.12-setup.exe	30 Dec 2017	Sourceforge.net
codeblocks-17.12-setup-nonadmin.exe	30 Dec 2017	Sourceforge.net
codeblocks-17.12-nosetup.zip	30 Dec 2017	Sourceforge.net
codeblocks-17.12mingw-setup.exe	30 Dec 2017	Sourceforge.net
codeblocks-17.12mingw-nosetup.zip	30 Dec 2017	Sourceforge.net
codeblocks-17.12mingw_fortran-setup.exe	30 Dec 2017	Sourceforge.net

Step 3: Run the downloaded installer and accept the default options.



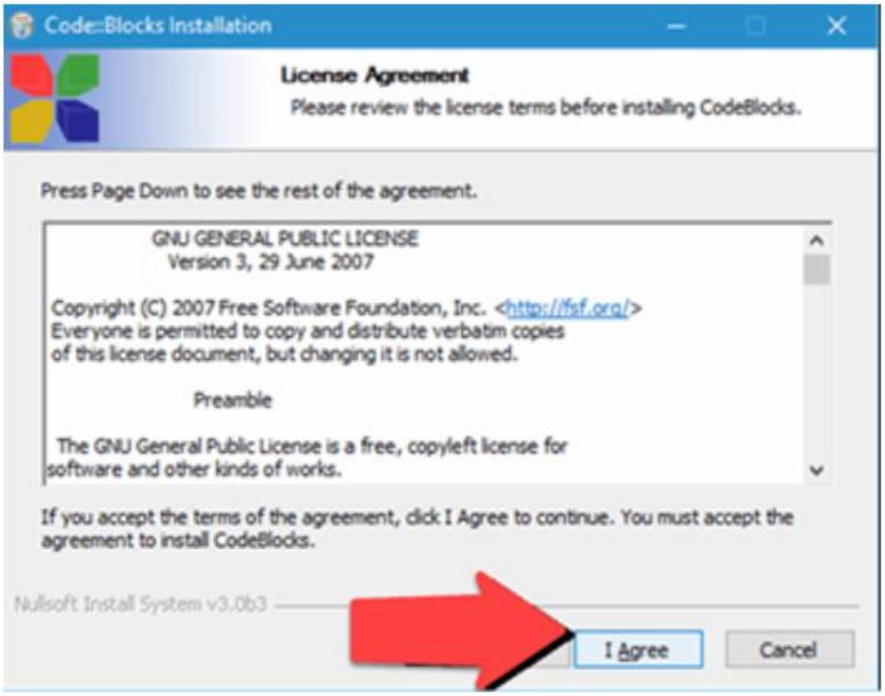
Welcome to CodeBlocks Setup

Setup will guide you through the installation of CodeBlocks.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Step 4: Accept the Agreement.



Code::Blocks Installation

License Agreement

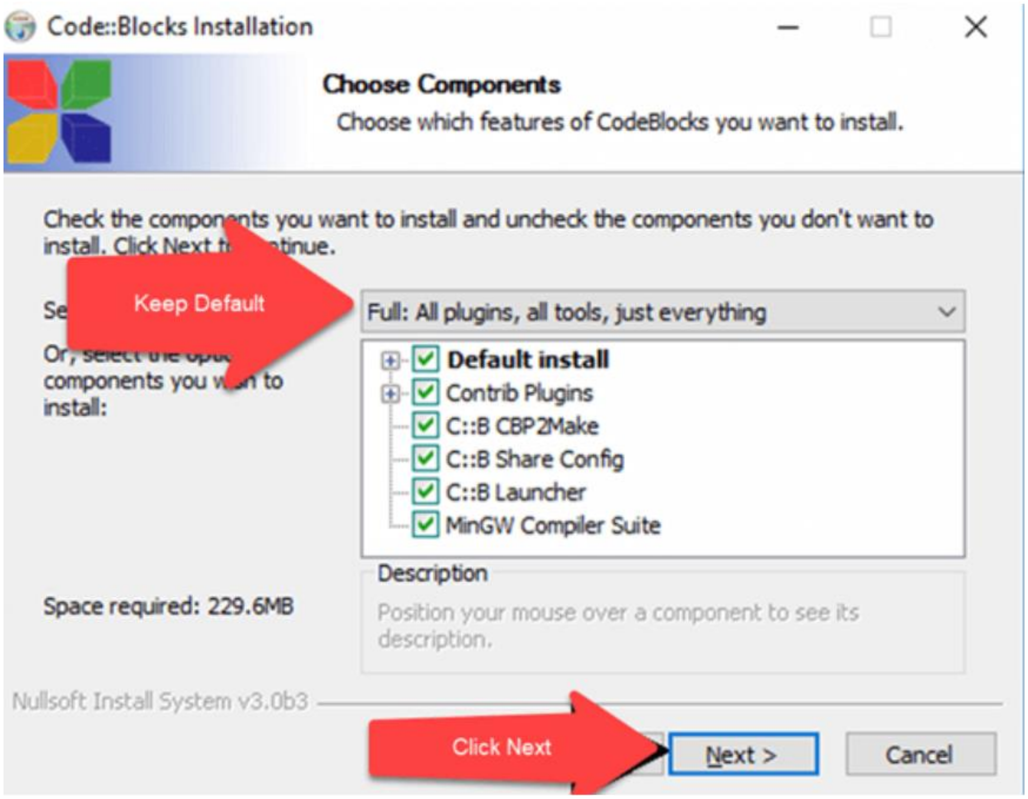
Please review the license terms before installing CodeBlocks.

Press Page Down to see the rest of the agreement.

```
            GNU GENERAL PUBLIC LICENSE
              Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                    Preamble

  The GNU General Public License is a free, copyleft license for
software and other kinds of works.
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install CodeBlocks.

Step 5: Keep the component selection default and click Next.



Code::Blocks Installation

Choose Components

Choose which features of CodeBlocks you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select the type of install: Full: All plugins, all tools, just everything

Or, select the optional components you wish to install:

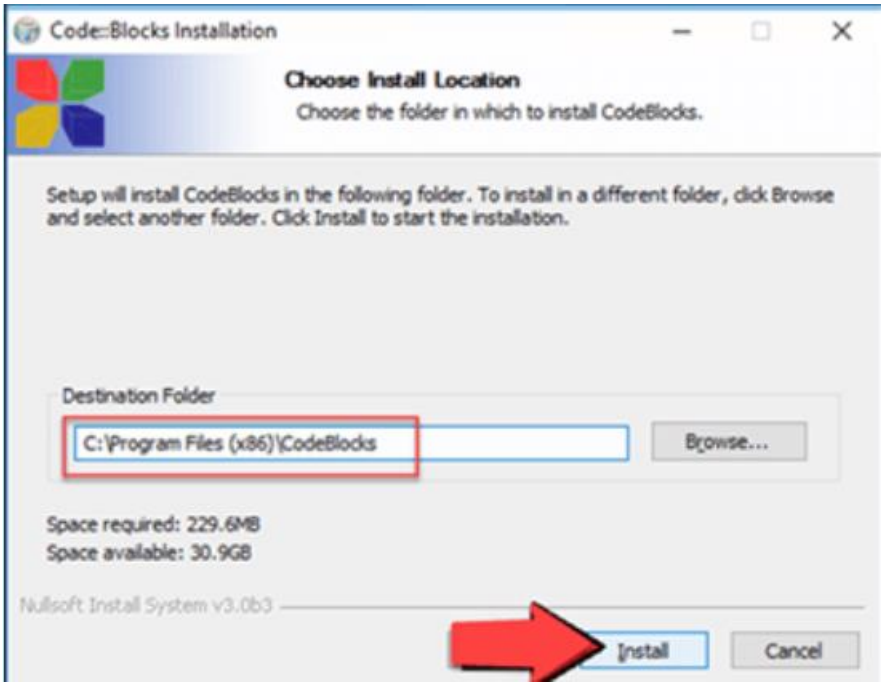
- ☑ **Default install**
- ☑ Contrib Plugins
- ☑ C::B CBP2Make
- ☑ C::B Share Config
- ☑ C::B Launcher
- ☑ MinGW Compiler Suite

Space required: 229.6MB

Description

Position your mouse over a component to see its description.

Step 6: change the installation folder and click Next.



Choose Install Location

Choose the folder in which to install CodeBlocks.

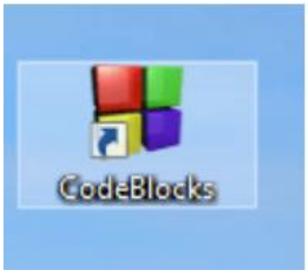
Setup will install CodeBlocks in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files (x86)\CodeBlocks

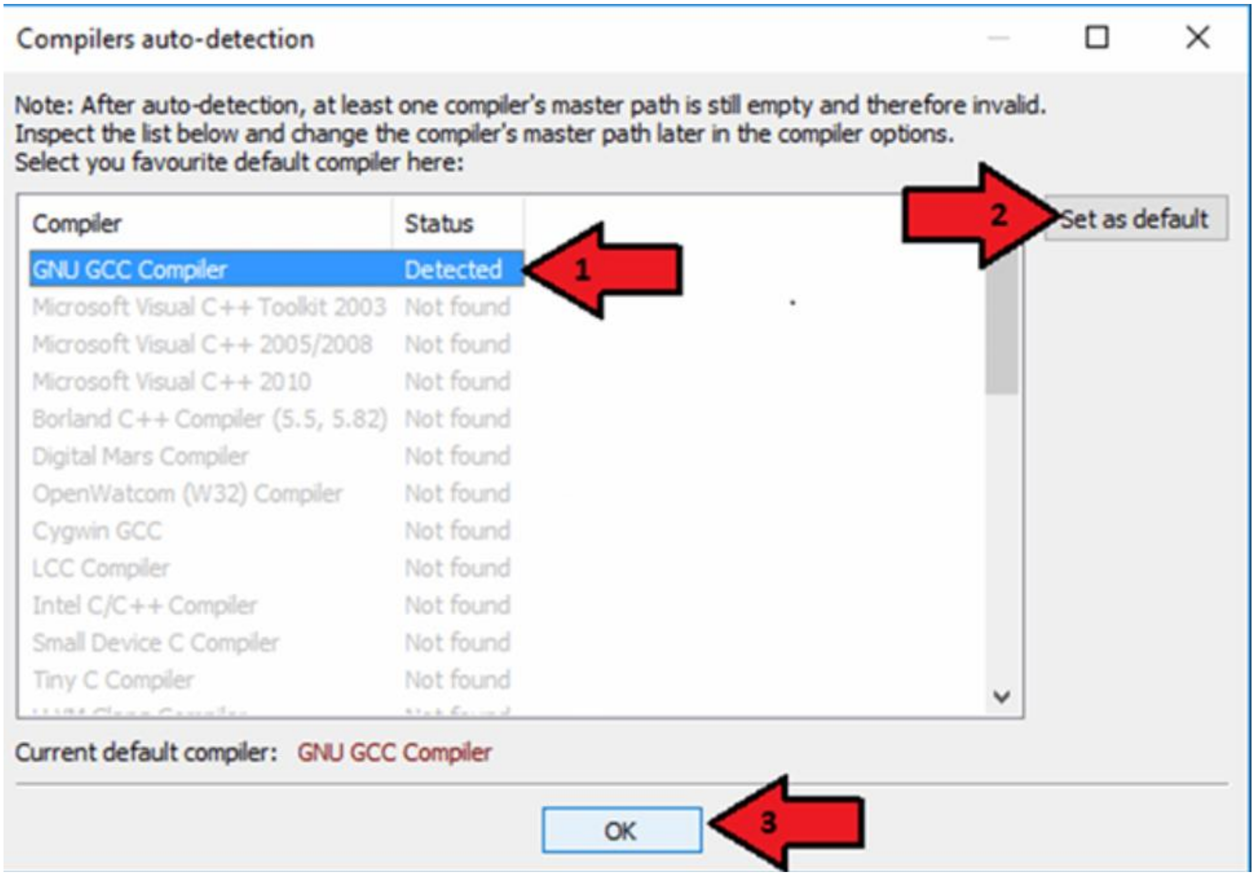
Space required: 229.6MB
Space available: 30.9GB

Nullsoft Install System v3.0b3

Step 7: Double click on the icon and launch.



Step 8: detect the gcc compiler automatically, set it as default.



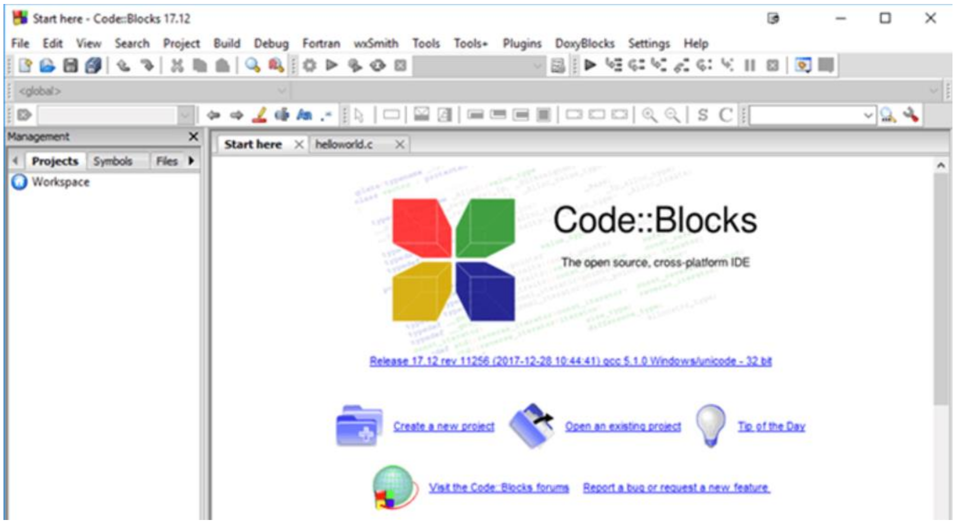
Compilers auto-detection

Note: After auto-detection, at least one compiler's master path is still empty and therefore invalid.
Inspect the list below and change the compiler's master path later in the compiler options.
Select you favourite default compiler here:

Compiler	Status
GNU GCC Compiler	Detected
Microsoft Visual C++ Toolkit 2003	Not found
Microsoft Visual C++ 2005/2008	Not found
Microsoft Visual C++ 2010	Not found
Borland C++ Compiler (5.5, 5.82)	Not found
Digital Mars Compiler	Not found
OpenWatcom (W32) Compiler	Not found
Cygwin GCC	Not found
LCC Compiler	Not found
Intel C/C++ Compiler	Not found
Small Device C Compiler	Not found
Tiny C Compiler	Not found
LLVM Clang Compiler	Not found

Current default compiler: GNU GCC Compiler

Step 9: Check the IDE Home screen.

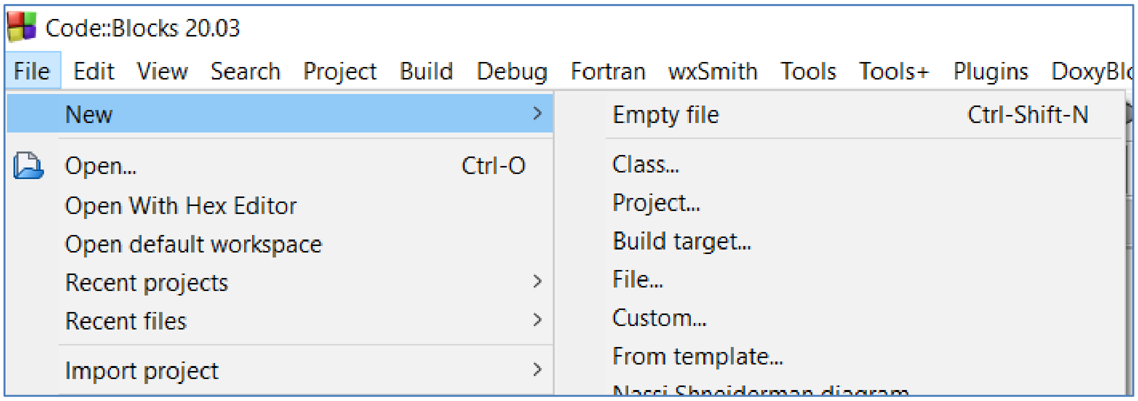


Start here - Code::Blocks 17.12

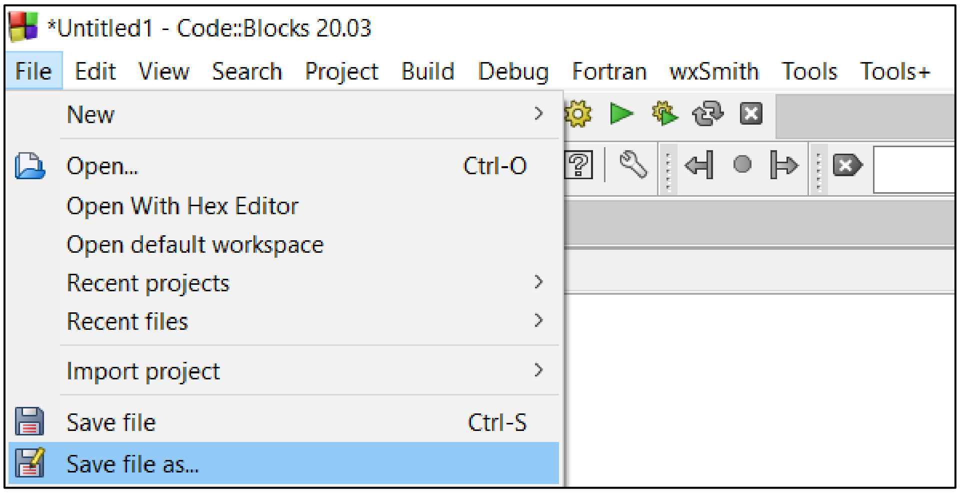
Release 17.12 rev 11256 (2017-12-28 10:44:41) gcc 5.1.0 Windows/unicode - 32 bit

How to Start First C Program.

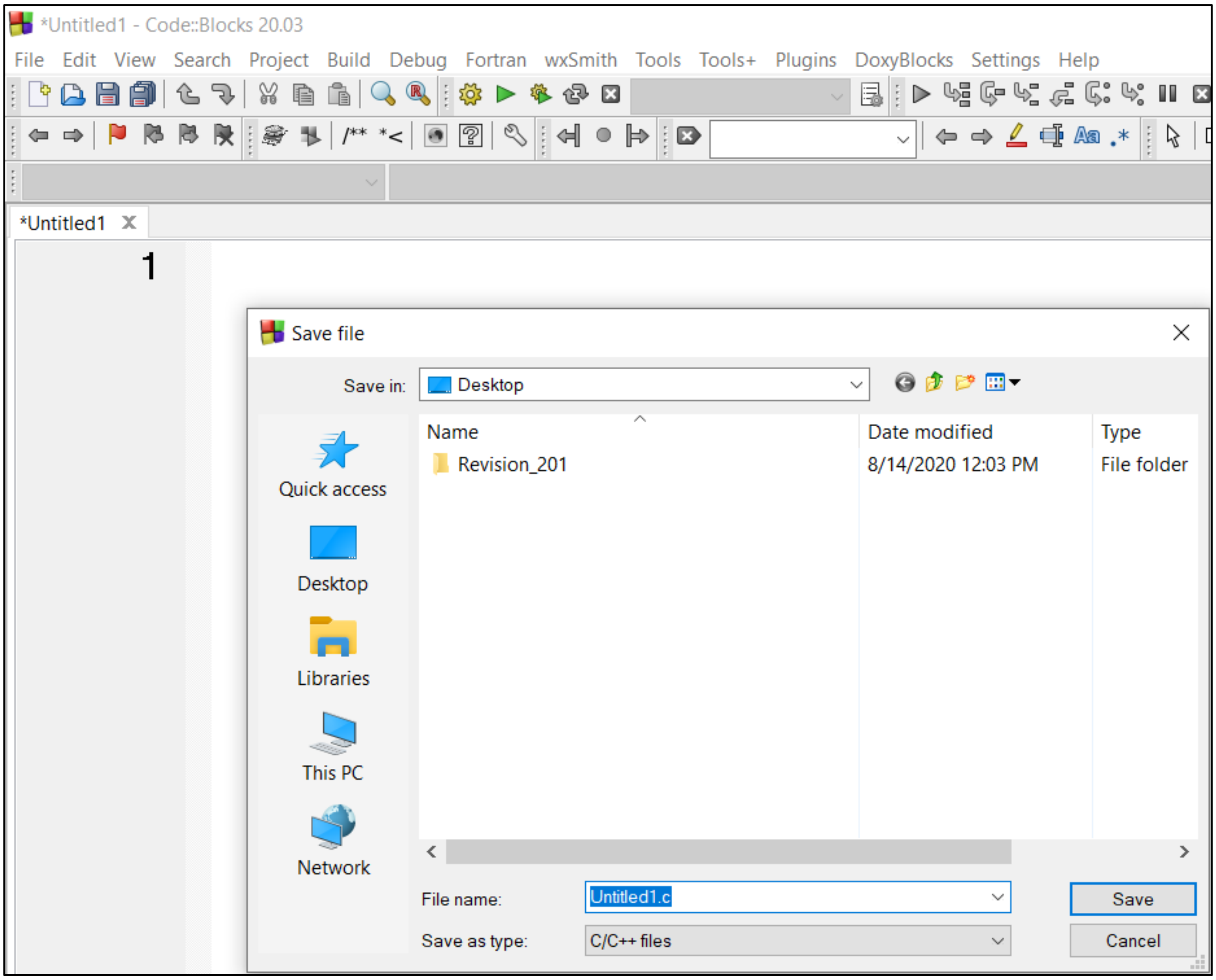
Step 1: Go to **File** on Menu. Then select **New** and **Empty File**.



Step 2: Go to File on Menu.



Step 3: Select the location to save the file. Then Save the file with file name and extension.


*Untitled1 - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
*Untitled1
Save file
Quick access
Desktop
Libraries
This PC
Network
Name
Date modified
Type
Save
Cancel

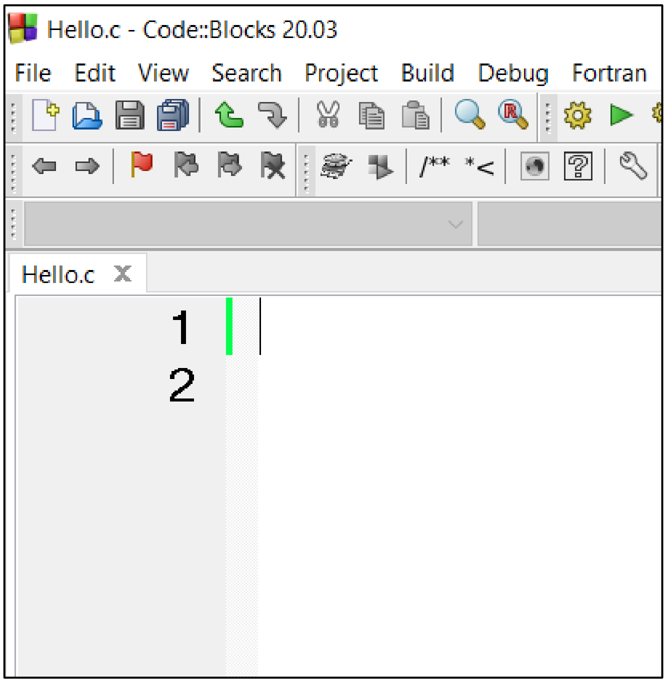
Save in: Desktop

Name	Date modified	Type
Revision_201	8/14/2020 12:03 PM	File folder

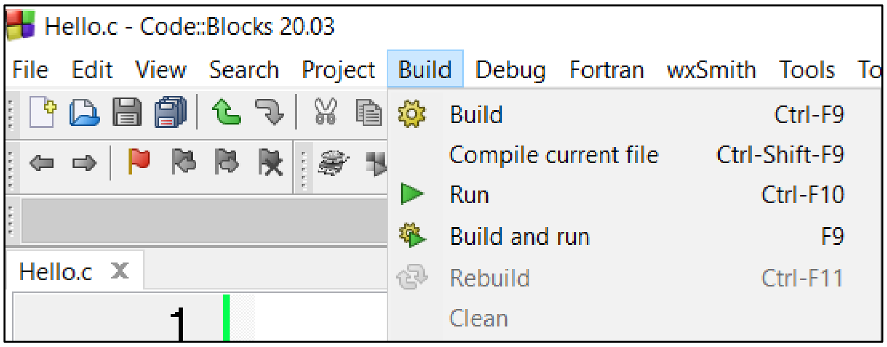
File name: Untitled1.c

Save as type: C/C++ files

Step 4: Type C program.


Hello.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran
Hello.c
1
2

Step 5: Compile and Run.


Hello.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools To
Build Ctrl-F9
Compile current file Ctrl-Shift-F9
Run Ctrl-F10
Build and run F9
Rebuild Ctrl-F11
Clean
Hello.c
1

Session 3.2

Experiment 1:

Step 1: Open an Empty File in CodeBlocks software.
Step 2: Key in the following C program.

```
#include <stdio.h>
int main(void)
{                printf("Diploma in Software Engineering
                 School of Computing
                 National Institute of Business Management");
        return 0;
}
```

Step 3: Save the above program as print.c in your working directory.
Step 4: Compile and run the above program
Step 5: Write your observations and conclusions

Experiment 2:

Repeat the experiment 1 with the following code.

```
#include <stdio.h>
int main(void)
{
        printf("Diploma in Computer System Design Management Information Systems
        Division National Institute of Business Management");

        return 0;
}
```

Experiment 3:

Repeat the experiment 1 with the following code.

```
#include <stdio.h>
int main(void)
{
        printf("Diploma in Computer System Design \nManagement Information Systems
        Division \nNational Institute of Business Management");
        return 0;
}
```

Experiment 4:

Repeat the experiment 1 with the following code.

```
#include <stdio.h>
int main(void)
{
        printf("Diploma in Computer System Design \t Management Information Systems
        Division \t National Institute of Business Management");
        return 0;
}
```

Experiment 5:

Step 1: Write two different programs to display the following table on your monitor.

```
                            <---------------- Marks ---------------->

        Student Number        Subject 1     Subject 2      Subject 3

        -------------------   -----------   ------------   -----------

        Stu-1                        40            50             60

        Stu-2                        50            70             80

        Stu-3                        50            40             30
```

Step 2: Compare the techniques used in the above programs and recommend one for practice.

Session 3.3

Experiment 1:

Step 1: Open an Empty File in CodeBlocks software.
Step 2: Key in the following C program.

```
#include <stdio.h>
int main(void)
{         char c;
          c = 'A';   // Single quotes
          printf("\nCharacter A : %c", c);

          return 0;
}
```

Step 3: Save the above program as charTest1.c, compile and run.
Step 4: Modify the above program to get the following code.

```
#include <stdio.h>
int main(void)
{         char c;
          char b;

          c = 'A'; // Single quotes
          b = “B”; // Double quotes

          printf("\nCharacter A : %c", c);
          printf("\nCharacter B : %c", b);
          return 0;
}
```

Step 5: Save, compile and run the above program.
Step 6: Write your observations and conclusion.

Experiment 2:

Step 1: Open an Empty File in CodeBlocks software.
Step 2: Key in the following C program.

```
#include <stdio.h>
int main(void)
{
        char c;
        c = 'AB';          // Single quotes
        printf("\nCharacter A : %c", c);

        return 0;
}
```

Step 3: Save the above program as charTest2.c, compile and run.
Step 4: Write your observations and conclusion.

Experiment 3:

Step 1: Open an Empty File in CodeBlocks software.
Step 2: Key in the following C program.

```
        #include <stdio.h>
        int main(void)
        {       int intSize, shortSize, longSize, charSize, floatSize,doubleSize;
                charSize = sizeof(char);
                shortSize = sizeof(short);
                intSize = sizeof(int);
                longSize = sizeof(long);
                floatSize = sizeof(float);
                doubleSize = sizeof(double);

                printf("\nSize of char data type : %d", charSize);
                printf("\nSize of short data type : %d", shortSize);
                printf("\nSize of int data type : %d", intSize);
                printf("\nSize of long data type : %d", longSize);
                printf("\nSize of float data type : %d", floatSize);
                printf("\nSize of double data type : %d", doubleSize);
        return 0;
          }
```

Step 3: Save the above program as data_types.c in your working directory.
Step 4: Compile and run the program.
Step 5: Write your observations and conclusion.

Experiment 4:

Step 1: Open an Empty File in CodeBlocks software.
Step 2: Key in the following C program.

```
#include <stdio.h>
int main(void)
{       char c1;
        int i;
        float x;
        double y;

        printf("\n%s\n%s", "Input a character", "an int, a float, and a double:");
        scanf("%c%d%f%lf",&c1,&i,&x,&y);
        printf("\nHere is the data that you typed in: \n");
        printf("%3c%5d%17f%17ebs \n",c1,i,x,y);
        printf("\n");

        return 0;
}
```

Step 3: Save the above program as input.c in your working directory.
Step 4: Compile and run the program.
Step 5: Write your observations and conclusion

Session 3.4

Experiment 1:

Step 1: Open an Empty File in CodeBlocks software.
Step 2: Key in the following C program.

```
        /* Converting kilometers to meters*/
        #include <stdio.h>
        int main()
        {        float  kilometers = 10.57625;
                 float   meters = 0;

                 meters = kilometers * 1000;

                 /*        Printing value stored in ‘meters’
                 %.2f print as floating point, 2 digits   after decimal point */

                 printf("10.5 km in meters :%.2f \n", meters);
        return 0;
}
```

Step 3: Save the above program as convert.c in your working directory.
Step 4: Compile and run the program.
Step 5: Write your observations and conclusion
Step 6: Change the above program so that it can convert any distance from kilometers to meters.

Experiment 2:

Write a program that asks user to enter two numbers, obtain the two numbers from the user, and print the sum, product, difference, quotient, and modulus of the two numbers.

Experiment 4:

Write a program to input three integer numbers into an array and display each.

Experiment 5:

Write a program to input three integer numbers into an array and display result of multiplication.

Session 3.5

Experiment 1:

Step 1: Open an Empty File in CodeBlocks software.
Step 2: Key in the following C program.

```
#include <stdio.h>
int main()
{         if (1==1)
                    printf("1 is equal to 1");
          return 0;
 }
```

Step 3: Save the above program as if.c, compile and run.
Step 4: Modify the above program to get the following code.

```
#include <stdio.h>
int main()
{         int num;

          printf("Please enter number : ");
          scanf("%d", &num);
          if (num == 1)
                    printf("%d is equal to 1\n",num);
           else
                    printf("%d is not equal to 1\n",num);
return 0;
}
```

Step 5: Save, compile and run the above program.
Step 6: Write your observations and conclusion.

Experiment 2:

Do the followings for the given programming exercises:

Step 1: Prepare a flowchart to visualize your solution (programming logic)
Step 2: Do a paper-based simulation to verify the flowchart
Step 3: Convert the flowchart developed in step 1 into a C program
Step 4: Create a source file using the program developed in step 3, save, compile, run, and test it.

1. Write a program to determine whether a given number is positive, negative or equal to 0. Print necessary messages. After displaying the appropriate word, the program should skip a line and display the words "To try another number, run this program again."
2. Write a program to get mark for a subject and print the message “You Passed!!” if the mark is greater than or equal to 60 or to print the message “You Failed!!” if the mark is less than 60.
3. Write a program to print the grade depending on the mark entered by user according to following criteria

Mark	Grade
<40	F
<55	C
<70	B
<80	A

Step 5: Test your program with following inputs
Mark = -45
Mark = 55
Mark = 120
Step 6: Change your program if it is not producing the correct output

4. Write a program to determine whether a user entered number is odd or even. If the number is odd print the square of the number as the output. If the number is even print the cube of the number as the output.
5. The bonus is determined by the salesperson’s code and, in some cases, by the sales amount. If the code is 1 and the salesperson sold at least $10,000, then the bonus is $500; otherwise these salespeople receive $200. If the code is 2 and the salesperson sold at least $20,000, then the bonus is $600; otherwise these salespeople receive $550. All others receive $150. Write a program to print the bonus when a salesperson enters his code and sales amount.

Experiment 3:

Step 1: Open an Empty File in CodeBlocks software.
Step 2: Key in the following C program.

```
#include <stdio.h>
int main()
{         int i;
          for( i=0; i<=10 ; ++i){
                    printf("%d \t",i);
          }
 return 0;
}
```

Step 3: Save the above program as numbers.c, compile and run.
Step 4: Write your observations and conclusion.

Complete following Exercises:

1. Write a program to print even numbers from 1 to 100
2. Write a program to find total and average of 10 numbers entered by user.
3. Print following patterns, get the number of lines to be printed from the user

```
*

**

***
```

```
1

12

123
```

```
@

**

@@@
```

```
*****

****

***
```

```
  *

 ***

*****
```

```
***** *****

****   ****

***     ***
```

Experiment 4:

Step 1: Open an Empty File in CodeBlocks software.
Step 2: Key in the following C program.

```
#include <stdio.h>
int main()
{  int i=0;
   while( i <= 10){
         printf("%d \n",i);
         ++i;
   }
 return 0;
}
```

Step 3: Save the above program as numbers.c, compile and run.
Step 4: Write your observations and conclusion.

Complete following Exercises:

1. Write a program to add all numbers entered by user until user enter zero.
2. Write a simple program to print all the numbers that are less than the number user has entered (up to 0) Eg:- Input : 5
   Output : 4,3,2,1,0
3. Write a C program to find the factorial of a number, where the number is entered by user.
   (Hints: factorial of n = 1*2*3*...*n)
   Step 5: Test your program with the user inputs: 1, 0 and -1.
   Step 6: Change your program if it is not producing the correct output.

Experiment 5:

Step 1: Open an Empty File in CodeBlocks software.
Step 2: Key in the following C program.

```
#include <stdio.h>
int main()
{
int num,sum=0;
do {
        printf("Hello user, Enter a number: \n");
        scanf("%d",&num);
        sum +=num;
}while(sum < 100);
printf("Summation of values you have entered is %d", sum);
return 0;
}
```

Step 3: Save the above program as sum.c, compile and run.
Step 4: Write your observations and conclusion.

Complete following Exercises:

1. Write a C program to find average of maximum of n positive numbers entered by user. But, if the input is negative, display the average (excluding the average of negative input) and end the program
2. Write a C program to find the product of 4 integers entered by a user. If user enters 0 skip it.
3. Write a program in C to find the addition of the first 10 elements of the following series.
   1 + 1/2 + 1/3 + 1/4+ ………………
4. Write a program to check whether a number is prime or not.

Session 3.6

Experiment 1:

Step 1: Open an Empty File in CodeBlocks software.
Step 2: Key in the following C program.

```
# include <stdio.h>
int  integer_addition(int num1, int num2){
        int sum;
        sum = num1 + num2;
return sum;
}
int main() {
         int x,y,sum;
 printf("Please enter two values : ");
 scanf("%d %d", &x,&y);

 sum =  integer_addition(x,y); // function call
 printf("Summation of %d & %d is : %d", x,y,sum);

        printf("\n\nPlease enter two values : ");
        scanf("%d %d", &x,&y);

        // function call within printf
        printf("Summation of %d & %d is: %d",x,y,integer_addition(x,y));

return 0;
}// end of main function
```

Step 3: Save the above program as integer_addition.c, compile and run.
Step 4: Write your observations and conclusion.

Step 5: Change your program as follows.

```
# include <stdio.h>
int  integer_addition(int x, int y); // function prototype
int main() {
        int x,y,sum;
        printf("Please enter two values : ");
        scanf("%d %d", &x,&y);

        sum=  integer_addition(x,y); // function call
        printf("Summation of %d & %d is : %d", x,y,sum);

return 0;

} // end of main function

  /* Declaration of function integer_addition
        after main function  */

int  integer_addition(int num1, int num2){
        int sum;
        sum = num1 + num2;

return sum;
}
```

Step 5: Save, compile and run the above program.
Step 6: Write your observations and conclusion.

Complete following Exercises:

1. Write a program to return the addition, multiplication, minimum, maximum and average of given three integers. (Write separate functions for each operation)
2. Write a program to raise a given integer to its given power and print using a separate function E.g. (a^b)
3. Write a program to get three numbers from the user and duplicate(eg: x*2,y*2,z*2) those numbers using a separate function and print the results.
4. Write a program in C to find the addition of the first 10 elements of the following series (hyperbolic functions - hyperbolic tangent)

$$\tanh^{-1} x = x + \frac{1}{3}x^3 + \frac{1}{5}x^5 + \frac{1}{7}x^7 + \frac{1}{9}x^9 + \ldots\ldots\ldots$$

Experiment 2:

```
# include <stdio.h>
int sum(int a);
int main()
{         int n;

 printf("Up to which number do you want to add? ");
 scanf("%d", &n);

 printf("\nSummation of numbers 0 to %d is : %d",n,sum(n));

return 0;
} // end of main

int sum(int a)   {
          if(a<=1) {
                    printf("\n 1 ");
                    return 1;
                    }
          else {
                    printf("\n %d + sum(%d)", a, a-1);
                    return ( a + sum(a-1));
                    }
}
```

Complete following Exercises:

5. Write a recursive function to find the factorial of a number, where the number is entered by user.
6. Write a function to find given number of values in Fibonacci series.

    1. $f_1 = 1$, $f_2 = 1$, $f_n = f_{n-1} + f_{n-2}$