Modified: 05 09 2019

FLORIDA INTERNATIONAL UNIVERSITY



***Tutorial on C compiler capability from Code::Blocks***

**EEL 2880 Software Engineering Techniques Summer 2019**

Professor: Dr. Subbarao V Wunnava  Ph.D.,P.E.

Course Coordinator: Dr. Herman Watson Ph.D

Teaching Assistant: Mr. Midam Islam

Voluntary TAs: Mr. Ms. Apoorva; Ms Jamie Sanchez

*[ Download Code::Blocks from* <http://www.codeblocks.org/downloads>*]*

***C compiler capability from Code::Blocks 05 09 2019***

1) To start the Code blocks compiler, click the **Start** button on the task bar to open the start menu. Open the Programs menu ( All Programs in Windows XP/7) and select Code Blocks . In the menu bar, click **File >>>New >>Project .** Select  **OK.**

2) Select  **Console Application** . Then **GO.** You can select **“view as large icons”** in the downside of screen. Then click **NEXT.**

3) Now select C++ for our case . Then Click **“Finish”**

Write **Project title,Folder name,Project File name,Resulting file Name**, Then **Next .**

As shown, Code Blocks will create a directory called First Program (Project Title) and returns yourselected directory in Folder to create project in. Inside that directory will be the Project filename (FirstProgram) and a resulting filename, which contains a Code Block Project file (.cbp) named FirstProgram.cbp. The project title and project filename in this case are the same. However, they need not bethe same and these names can be altered.

Click on the Click Next after finished .

 Start by filling in the Project Title. You will notice that the Project Filename automatically becomes the same name. If you wish, you can change the filename, but for simplicity leave it as is. To specify thelocation of the folder to contain the project, click on the “...” button (selected in the picture above)

 browse to your desired folder on your drive to store the project.

 Press Ok after selection.

4) The next window to pop up will be the Compiler screen. Keep **Debug and Release** compiled versions of your program , Then Click **Next** . A screen will pop up to write program code.

The system will then return to the First Program window and you are ready to write your program.

 It should be noted that the Build target is Debug, which will allow you to use the debugger to find errors.

In the Management area of the screen (Shift-F2 toggles the Management display), you will see the files that are part of the project in the Projects tab.

To see the source files, click on the plus [+]’s to expand the Workspace and its subdirectories.

5) Under Sources, there is a file called main.cpp, which is automatically created for you when you build a console application.

**Adding Files To Your Project :**

If you have a project with additional existing files, go to the **Project menu** and select **“Add Files.”** This will bring in the files associated with your program. You also have the option to **Remove files.**

6) If you are creating a **New file**, you can use the pull-down **File menu** and open an empty file.

Instantly a new file will open to write Program. You will be asked to add this project in the Project. Then click **“YES”.** Save the file to your desired directory. Slect **“ DEBUG ” & “RELEASE ”.** Then **“OK”**

You can work with a debug target, which will allow you to testthe program using a debugger.

 A debug target will be large in size, because it has extra information in itto allow you to test for errors.

 A release target is smaller in size, because it does not have the debugging information.

7) New file will open as sample.cpp as for Example.

 The Sources now has sample.cpp as a source file in addition to the main.cpp file.

 You can remove main.cpp file if you don’t need it from Remove option.

8) To edit a file from your project, double click on it's name from Sources and it will appear in the window with line numbers. You can now edit the file and prepare your program.

9) After writing program Code you can Now click Project, Build options to check debugging is ongoing or not.

For Debug make sure “producedebugging symbols[-g]” are checked then click OK

10) Now click Build , Compile current file (ctrl-shift-F9) . Now click Build and Run (F9) from Build Menu.

11) If your program code is OK,no Error in Debugging Then you will get output screen with your desired result.

Before closing Save all files in your particular project . Also Save Project.

 **## One Worked Out Example :**

 ## **Write a program code in where you will enter 4 number ,Result will be sum12, sum34, dif12, dif34, prod12, prod34, idiv12, idiv34** **float fdiv12, fdiv34.**

#include <stdio.h>

int main (void)

// declare variables both integer and floating point

{ int num1, num2, num3, num4;

 int sum12, sum34, dif12, dif34, prod12, prod34, idiv12, idiv34;

 float fdiv12, fdiv34; // floating point decimal values

 // read 4 numbers num1, num2, num3, num4

 printf (" Enter 4 numbers num1, num2, num3, num4 \a\a \n");

 scanf("%d %d %d %d", &num1, &num2, &num3, &num4);

//perform arithmetic operations

 sum12 = (num1 + num2);

 sum34 = (num3 + num4);

 dif12 = (num1 - num2);

 dif34 = (num3 - num4);

 prod12 = (num1\*num2);

 prod34 = (num3\*num4);

 idiv12 = (num1/num2);

 idiv34 = (num3/num4);

 fdiv12 = (float) num1/num2;

 fdiv34 = (float) num3/num4;

 printf (" numbers num1, num2, num3, num4 are:\a\a %d %d %d %d \n\n", num1, num2, num3, num4);

 printf (" sum12, sum34 dif12, dif34 are:\a\a %d %d %d %d \t\n\n", sum12, sum34, dif12, dif34);

 printf (" prod12, prod34, idiv12, idiv34 are: %d %d %d %d \t\n\n", prod12, prod34, idiv12, idiv34);

 printf ("\a\a fdiv12 and fdiv34 are: %.2f %.4f \t\t \n ", fdiv12, fdiv34);

 printf ("\a\a\n");

 if (sum12 >= sum34)

 {

 printf (" sum12 is larger than or equal to sum34 \a\n");

 }

 if (sum34 > sum12)

 {

 printf (" sum34 is larger than sum12 \a\n");

 }

//end of the program

 return 0;

}

Result Display :



All these compilers have built in functions and headers. They need to be used for conducting any

Software development.

It is always a good practice to use at least 2 or possibly 3 different C compilers and C++ compilers to

Develop C and C++ software schemes. Also, the functions developed for C can be exported to C++.

Functionality should remain similar for C, or C++ or Java and for any other software platforms