

Code::Block manual

for CS101 course



Department of Computer Science and Engineering
Indian Institute of Technology - Bombay
Mumbai - 400076.

July 19, 2014

Contents

1	Introduction	1
1.1	Code::Blocks	1
1.2	Code::Block IDE	1
2	Working with Code::Block on Ubuntu	6
2.1	Writing a new c/c++ program	6
2.2	Building the Project	8
2.3	Opening Existing Program/Project	9

List of Figures

1	Code::Block IDE	1
2	Menu Bar	1
3	Main Toolbar	2
4	Debugger Toolbar	2
5	Compiler Toolbar	3
6	Manager	3
7	Editor, Start/Home Page	4
8	Shortcut to Create New Project or Open Existing Project	4
9	Shortcut to History of Projects Opened Using Code::Blocks	5
10	Logs	5
11	Starting a new project	6
12	Selecting the language for project	6
13	Title for Project	7
14	Selecting Compiler to Compile the Program	7
15	Project Node when Expanded	8
16	Project Node when Expanded for helloworld.c (with code shown in editor)	8
17	Output for helloworld.c	8
18	Select file with .cbp extension to open an existing project	9

1 Introduction

1.1 Code::Blocks

“Code::Blocks is a free C++ IDE built to meet the most demanding needs of its users.” [1]. Developed by ‘The Code::Blocks Team’, Code::Block is a free, open-source [2] and cross-platform IDE, which supports various free compilers. It is built around plugin framework, which allows functionality of Code::Block to be extended by installing appropriate plugins. Plugins required for compiling and debugging are already provided by default. This manual is prepared after installing and testing Code::Block on Ubuntu 12.04.

1.2 Code::Block IDE

Code::Block IDE is shown in figure 1 (Ubuntu 12.04). The main parts of Code::Block along with figures are discussed below

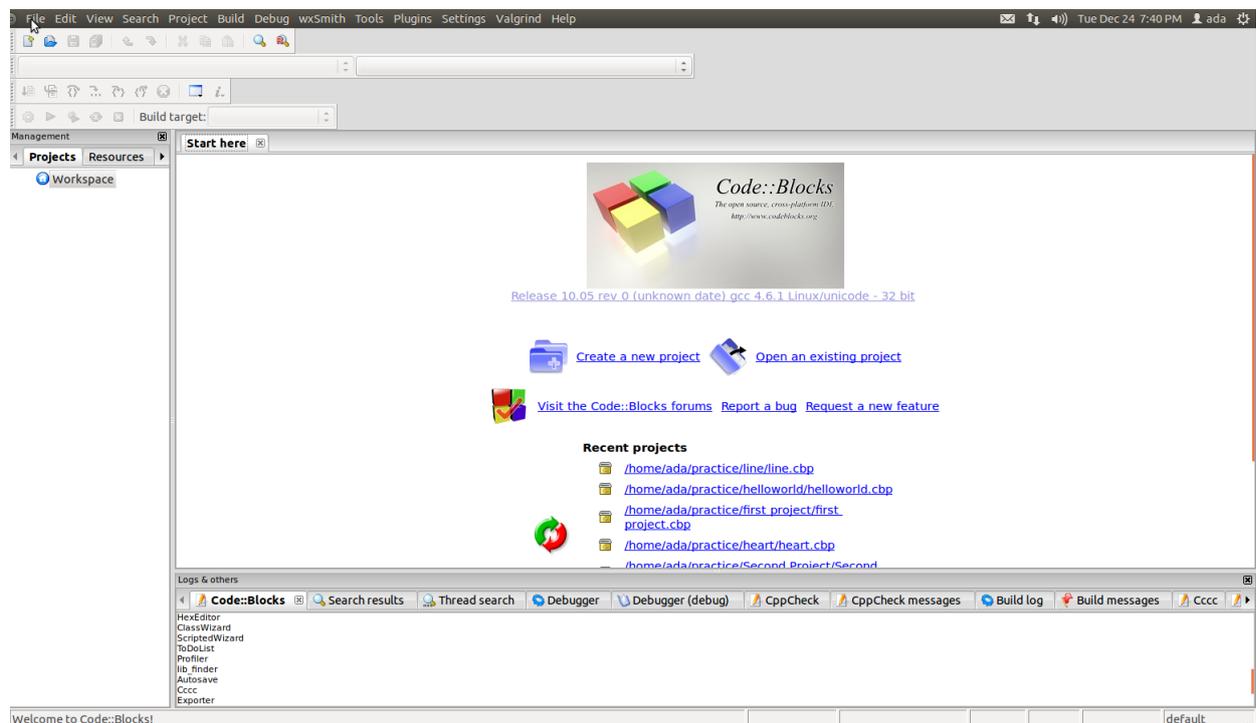


Figure 1: Code::Block IDE

1. Menu bar:

Menu bar is shown in figure 2. Menu bar can be toggled using F10. Few important link in menu bar are described below (described from left to right):

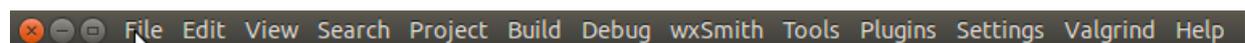


Figure 2: Menu Bar

- (a) File: File menu link contains options to create a new project, open an already existing project, save file, save project, save workspace and save everything. It also contains options for closing a single file, closing a project or closing entire workspace. Other options in File are to print, export and quit the Code::Block

- (b) Edit: All the editing options required for editor are provided in Edit.
- (c) View: This menu link contains link for various perspectives and toolbars along with manager, logs, script console, status bar, full screen.
- (d) Project: Options related to the project is provided in this link which includes configuring build options along with options for adding files, removing files and autoversioning of project.
- (e) Build: Options for building the project, compiling a single file, running the project, building and running the project, rebuilding the project and cleaning the project is provided in build. Options for Building, rebuilding and cleaning the entire workspace is also provided along with options to select target (debug/release) and analysing error one by one.
- (f) Debug: Various Debugging options are provided in this link.
- (g) Plugins: Various plugins can be executed using this link. The link to manage the plugins is also provided here.
- (h) Settings: This contains link for various settings, setting related to *Environment...*, *Editor...*, *Compiler and debugger...*, *Global Variables...* and *Scripting...*. Script to be executed during Code::Block start-up can also be edited here.
- (i) Help: It contains information about Code::Block version, tips which can be toggled to be displayed at start-up and information about various plugins.

2. **Main tool bar:**

Main tool bar is shown in figure 3. The buttons in Main toolbar are (from left to right):



Figure 3: Main Toolbar

- (a) New File: For creating a new project.
- (b) Open: For opening an already created project.
- (c) Save : To save the file open in active editor (active editor means the editor tab in focus).
- (d) Save all files: To save all the files for the current/selected project.
- (e) Undo: To undo the executed action.
- (f) Redo: To redo the undone action.
- (g) Cut: To cut the selected/highlighted part in editor.
- (h) Copy: To copy the selected/highlighted part in editor.
- (i) Paste: To paste the cut/copy message in editor.
- (j) Find: To find required text in the file in active editor.
- (k) Replace: To replace required text in the file in active editor by some alternate text.

3. **Debugger tool bar:**

Debugger tool bar is shown in figure 4. Debugger tool bar is used to debug the current/selected project. The buttons in debugger toolbar are (from left to right) *Debug/Continue*, *Run to cursor*, *Next line*, *Next instruction*, *Step into*, *Step out*, *Stop debugger*, *Debugging Windows* and *Various info*. You will be able to understand the use of this buttons by rigorous practise of debugging various projects.



Figure 4: Debugger Toolbar

4. **Compiler tool bar:**

Compiler tool bar is shown in figure 5 and is used in building/compiling/running the current/selected project. The buttons in Compiler toolbar are (from left to right):



Figure 5: Compiler Toolbar

- (a) Build: For building the current/selected project.
- (b) Run: For running the current/selected project.
- (c) Build and run: For building and running the current/selected project.
- (d) Rebuild: For rebuilding the current/selected project.
- (e) Abort: For aborting the build process for the current/selected project.
- (f) Build target: For defining the type of build target for current/selected project, either debug or release.

5. **Manager:**

Manager is shown in figure 6. It is labelled as Management. This window provides the list of all the open projects and files for easy access to any required file of any project.

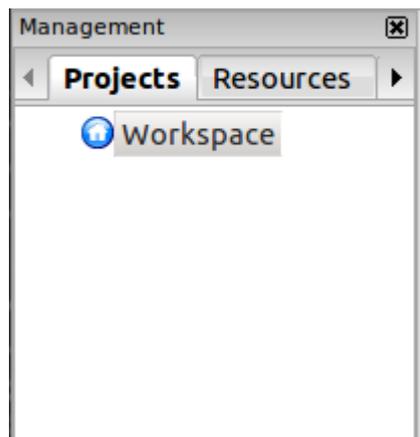


Figure 6: Manager

6. **Editor:**

Editor is shown in figure 7. Here, all the coding work will take place. It is provided in tabbed fashion to work with many files at once. When no project is open, the start page or home page is displayed in editor. The links given in start page is divided into two parts and explained below

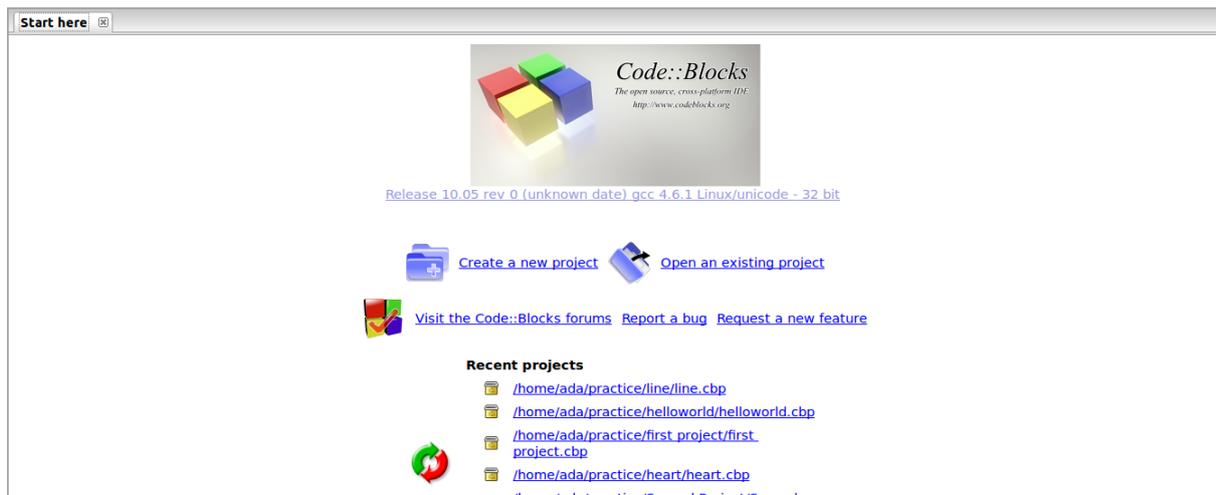


Figure 7: Editor, Start/Home Page

7. Figure 8 is short-cut on Starting page of IDE for creating a new project and opening an already created project. It also contains link for Code::Block forum where many useful resources can be found along with other useful discussions. The link points to url <http://forums.codeblocks.org/>. The second and third link points to BerLiOS Developer Site aims at enriching the Open Source community by providing a centralized place for Open Source Developers to control and manage Open Source Software Development.



Figure 8: Shortcut to Create New Project or Open Existing Project

8. Figure 9 is short-cut to list of projects and files already opened in the IDE. It is link to few projects and files from history of IDE.

Recent projects

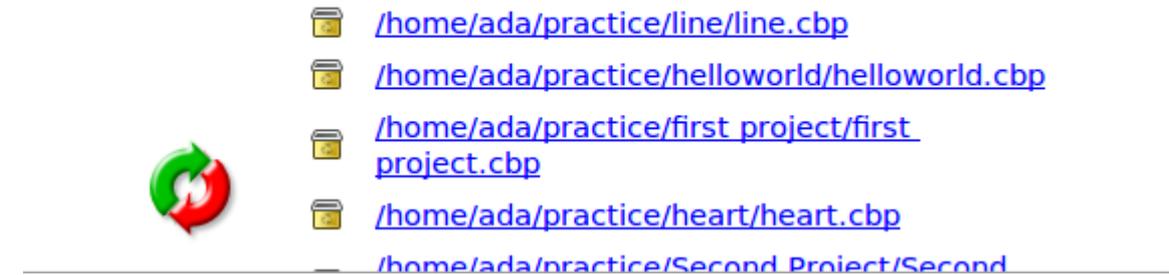


Figure 9: Shortcut to History of Projects Opened Using Code::Blocks

9. Logs:

Log window is shown in figure 6. It is labelled as 'Logs & others'. This window acts as log for various actions performed in IDE. All logs related to various activities can be checked at appropriate windows.

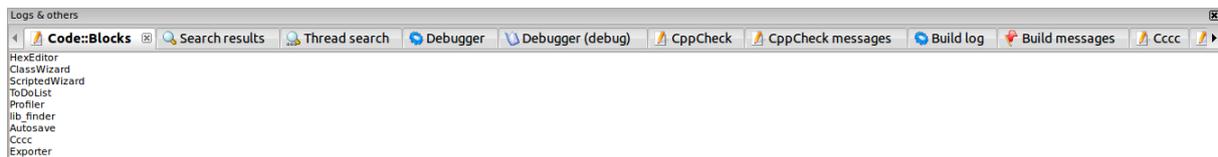


Figure 10: Logs

2 Working with Code::Block on Ubuntu

2.1 Writing a new c/c++ program

1. Click *File* → *New* → *Project*. The ‘New from template’ window opens as shown in figure 11. Select ‘Console application’ → on ‘Go’.

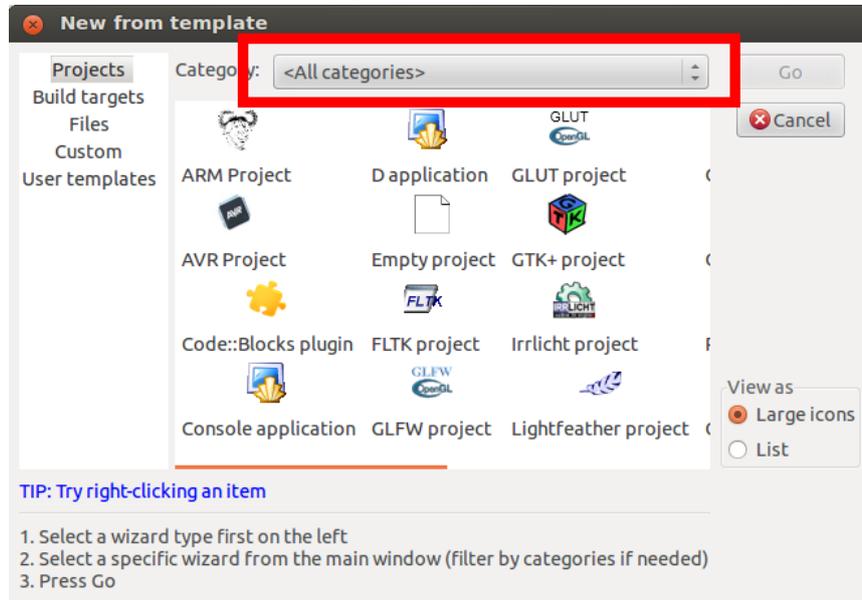


Figure 11: Starting a new project

2. When *Go* button is clicked, a new window opens as shown in figure 12. This window enables the user to select the required language to proceed. Select the language and click *Next*.

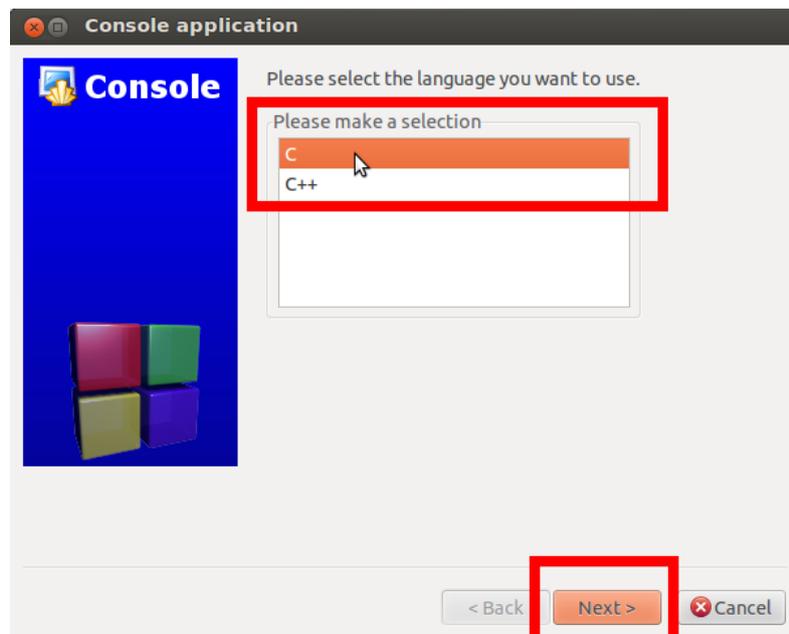


Figure 12: Selecting the language for project

- The subsequent windows enable the user to provide title for the project and the folder where user wishes to create the project in. This is shown in figure 13. After filling in the details click 'Next'.

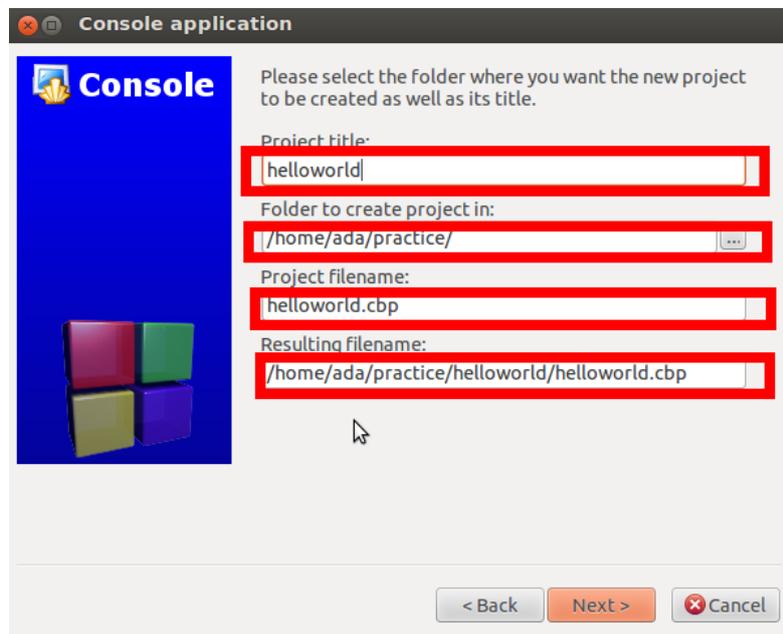


Figure 13: Title for Project

- Next window is to select the compiler. By default 'GNU GCC Compiler' is selected. Click *Finish*

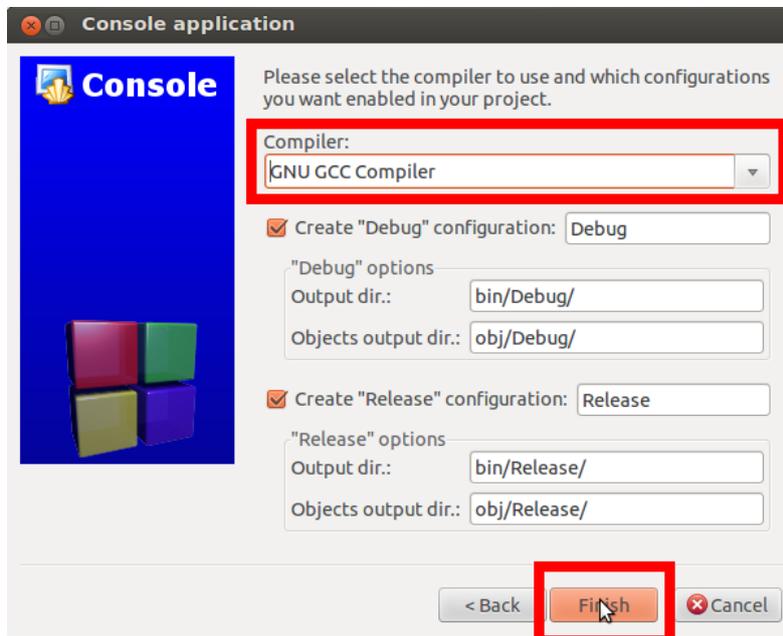


Figure 14: Selecting Compiler to Compile the Program

- Now, the project node opens in manager window as shown in figure 15. Project node can be expanded to see the main.c file.

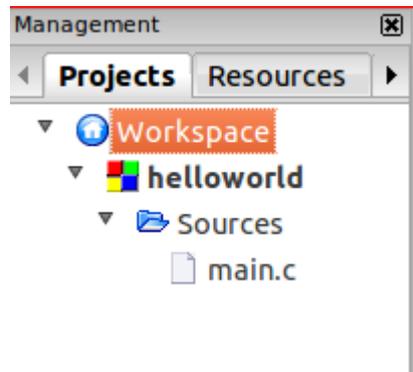


Figure 15: Project Node when Expanded

When main.c file is clicked, it opens in editor as shown in figure 16 for hello world project.

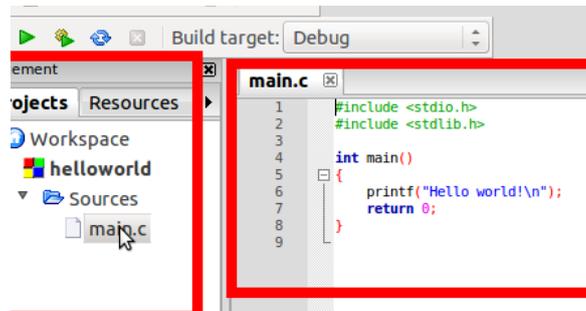


Figure 16: Project Node when Expanded for helloworld.c (with code shown in editor)

- While using Code::Blocks for the first time, some extra windows will be displayed. In this manual only the important windows are shown.

2.2 Building the Project

After the code is written, project needs to be built. Click 'Build' → 'Build and run'. Output is as shown in figure 17.

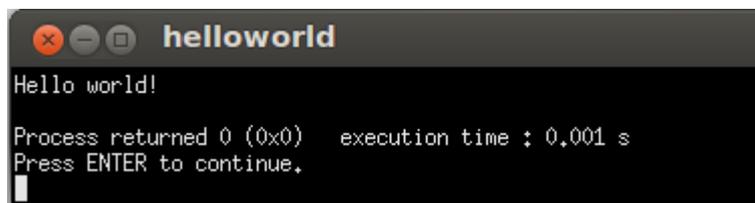


Figure 17: Output for helloworld.c

2.3 Opening Existing Program/Project

Click *File* → ‘Open’. Browse to desired directory and select the file with .cbp extension as shown in figure 18 and click *Open*.

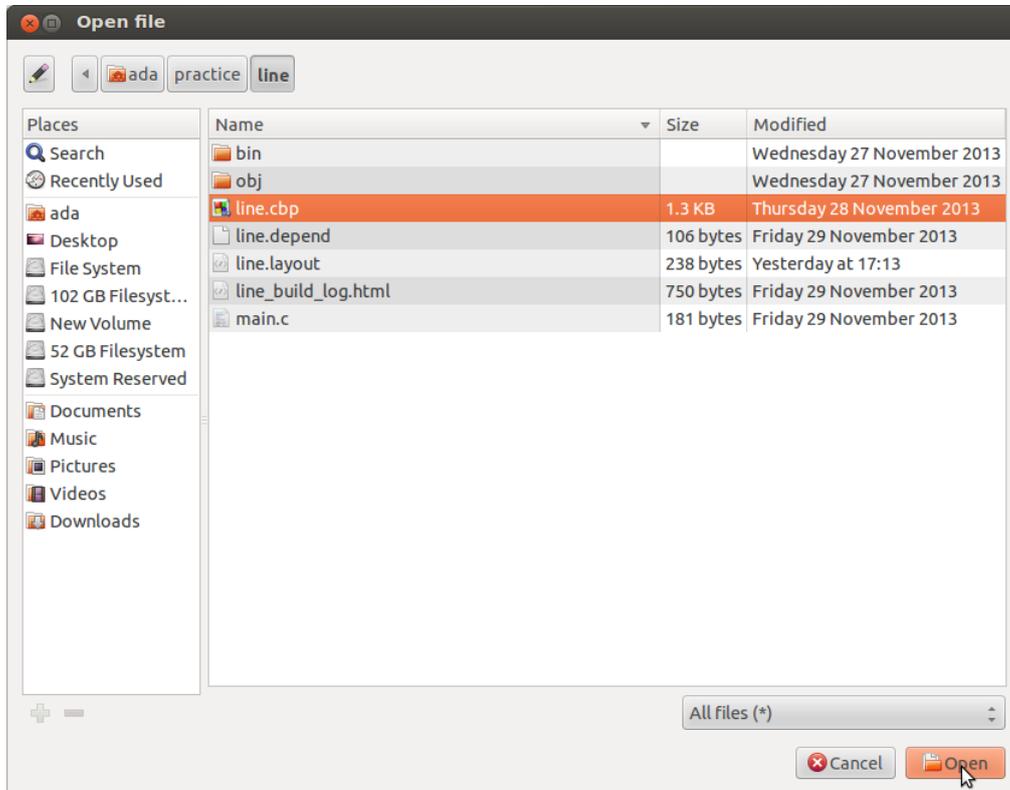


Figure 18: Select file with .cbp extension to open an existing project

References

- [1] The Code::Block Team. Code::block homepage
<http://www.codeblocks.org/>.
- [2] The Code::Block Team. Gpl v3.0 license
<http://www.codeblocks.org/license>.

Code::Block Tutorial For Running Your First C++ Program

By: Rahul Dev Parashar

Running Sample C++ Program In Code Blocks IDE

A Checklist

To write your first C++ program in Code Blocks IDE (Integrated Development Environment) you'll need:

- ✓ The Code::Blocks IDE installed on your system
 - To check that you have Code::Blocks installed or not, press 'window' button  on your keyboard and type 'code block' in search bar. The icon  will appear in Applications section. If the icon is there, it means Code::Blocks is installed.
 - If it is not installed then follow instructions given in Appendix-A at the end of document.

Creating Your First C++ Program

A sample program 'first_cpp_program_noofstudents.cpp' is taken as an example for demonstration. Now follow the steps given below:

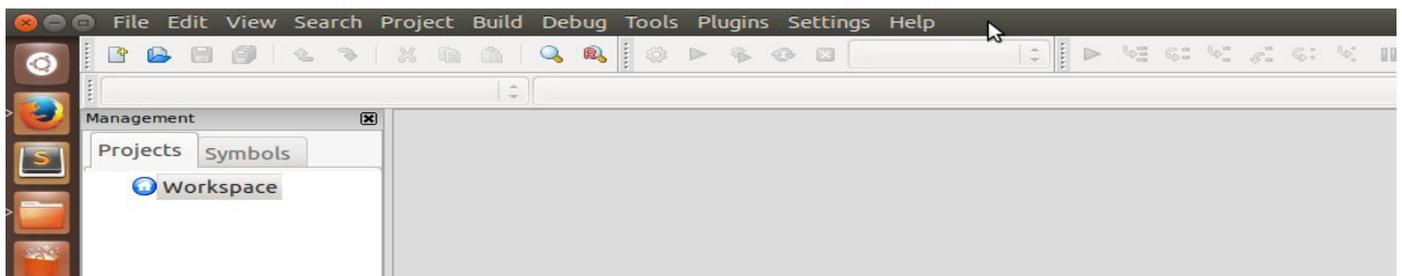
- ✓ Open the Code::Blocks IDE
 - Click 'Windows' button  on your keyboard and type 'code block' in search bar. Icon  will appear in Applications sections. Click on the icon and Code::Block window will appear on screen.

** Explanation for different components of IDE is given here:

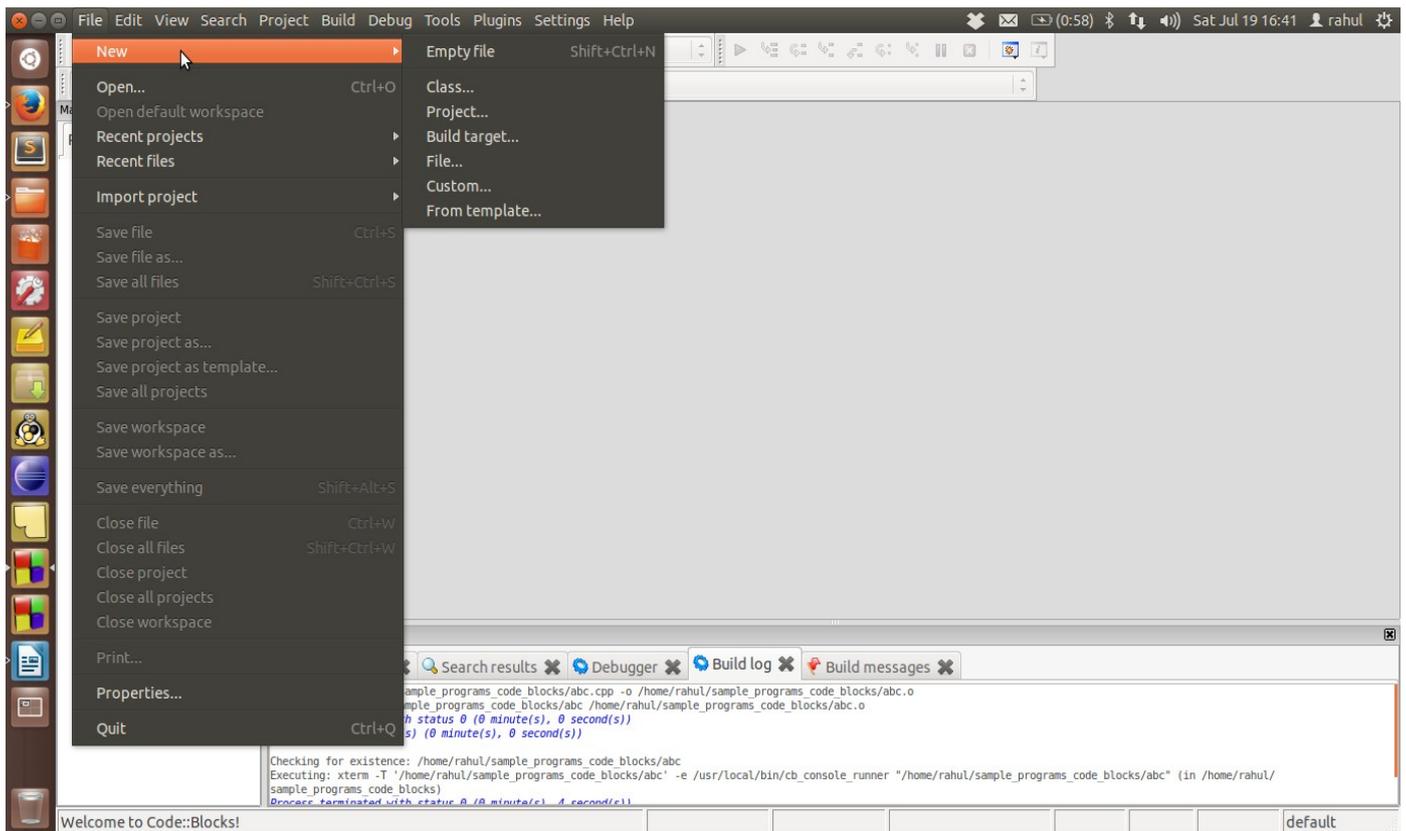
<http://www.it.iitb.ac.in/frg/wiki/images/e/e8/CodeBlockManual.pdf>

- ✓ Now follow the instructions given below in screenshots to write your first C++ program:

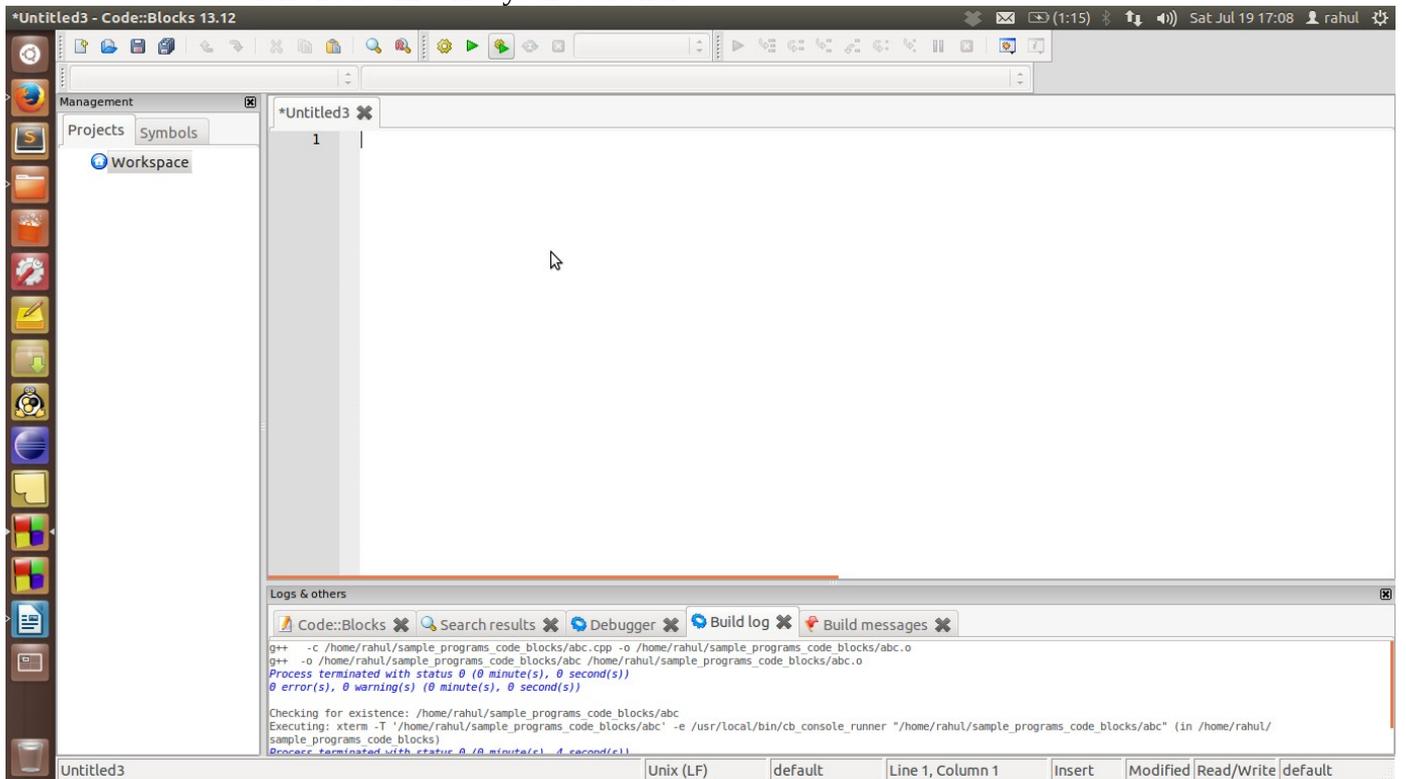
Step-1: Locate the menu toolbar by taking your mouse cursor to the top of the window. It is shown at top of the image where mouse cursor is.



Step-2: Now click on 'File' from menu toolbar and take your cursor on 'New'.



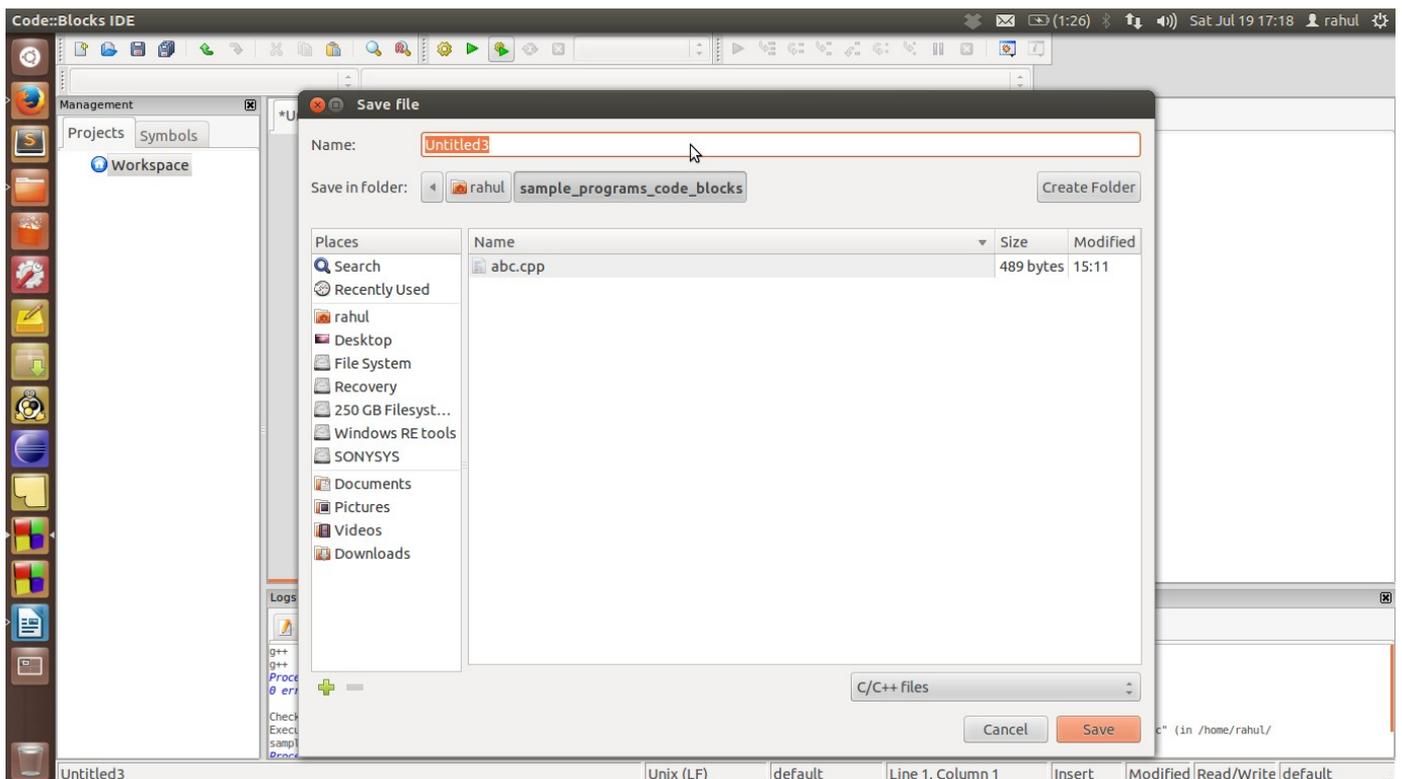
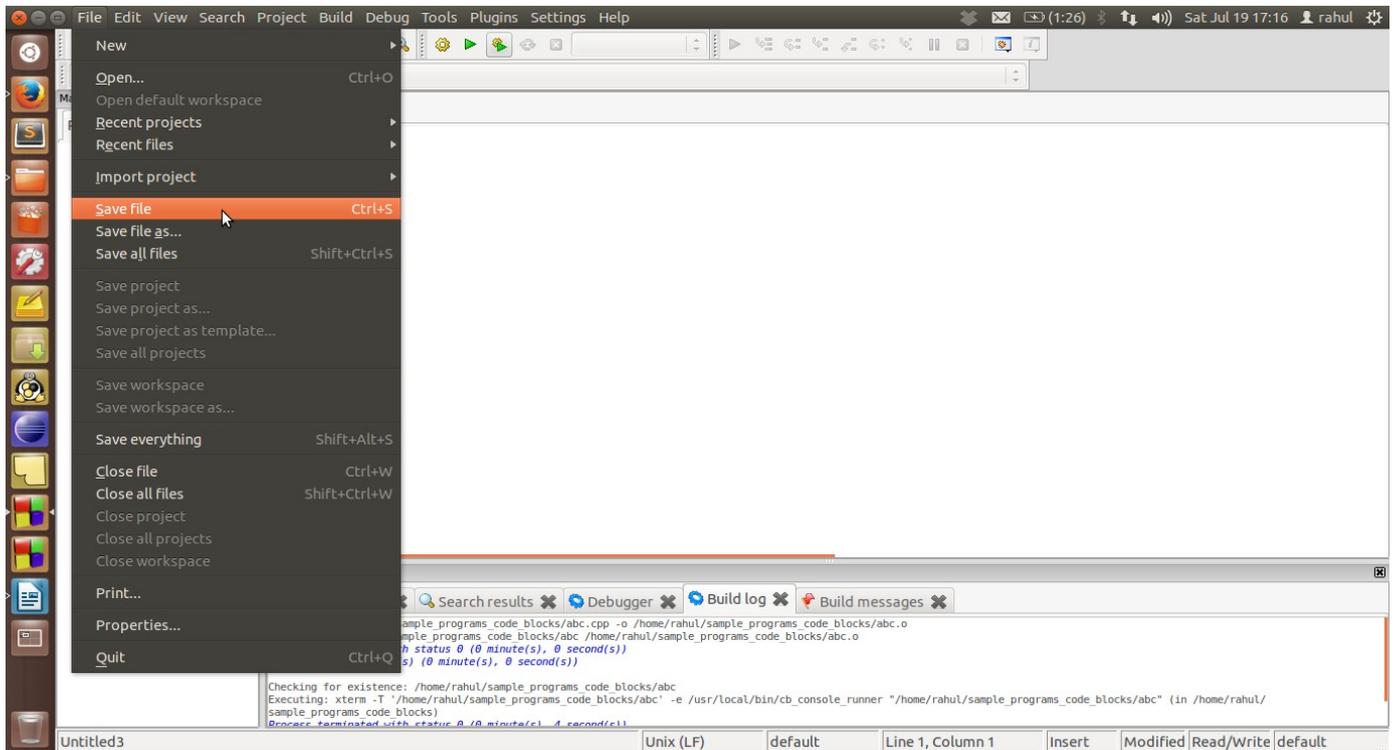
Step-3: Now click on 'Empty file'. A new file (Untitled_SOME_NUMBER) will be opened in editor section. You will write your code here.



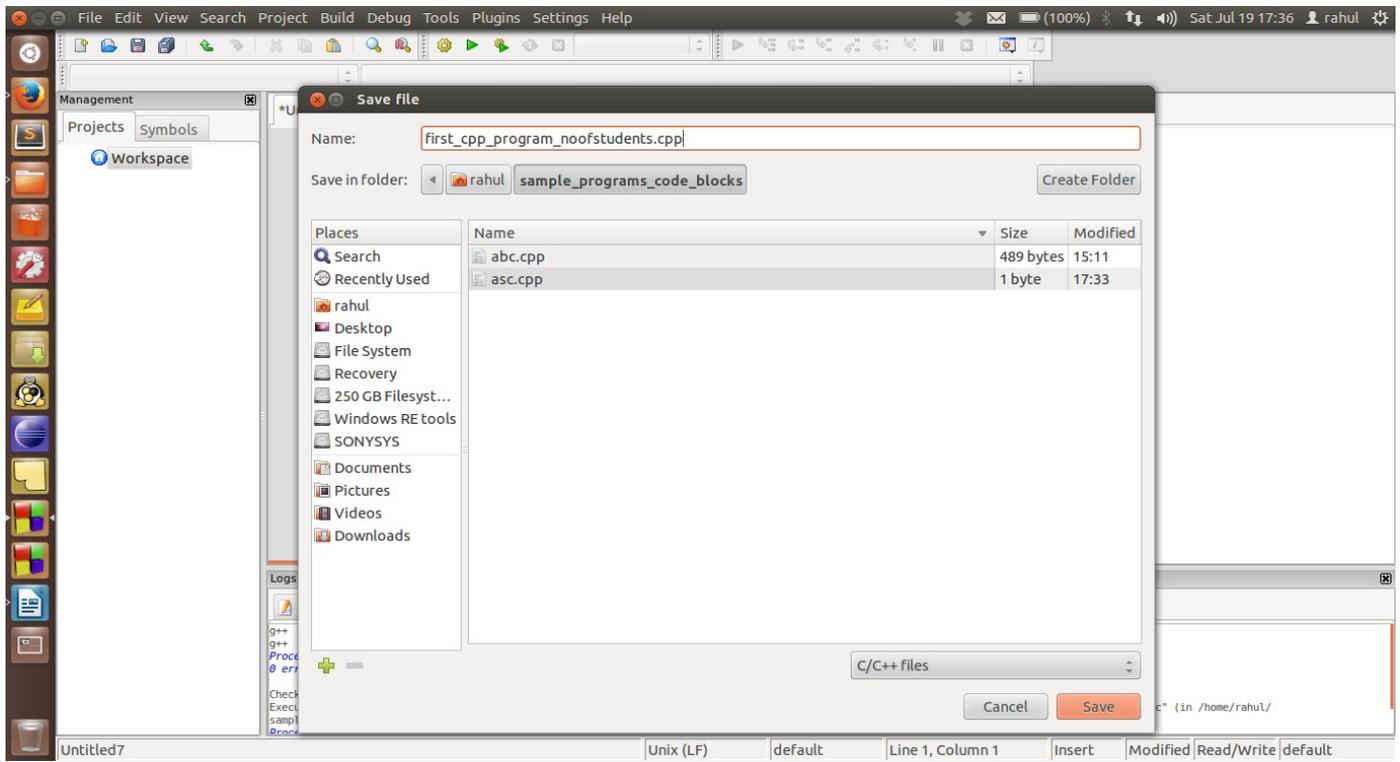
Shortcut: The steps 1 to 3 can be done directly by the keyboard shortcut Ctrl+Shift+N.

This 'Untitled 3' file in above image is in temporary buffer. So the code you will write in it will not be saved without saving this file on disk first. So first save this file on disk.

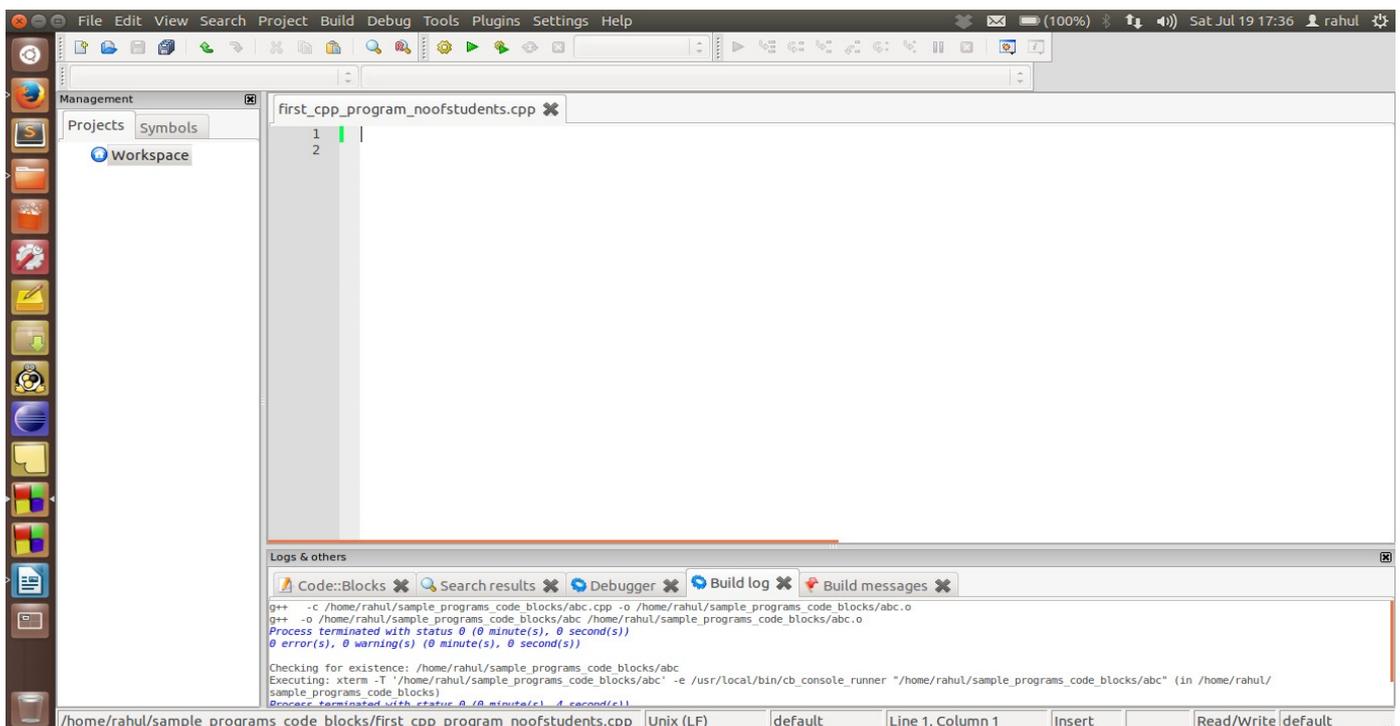
Step-4: To save this file go to 'File' in menu toolbar and click on 'Save file'. This can also be done using the keyboard shortcut Ctrl+S.



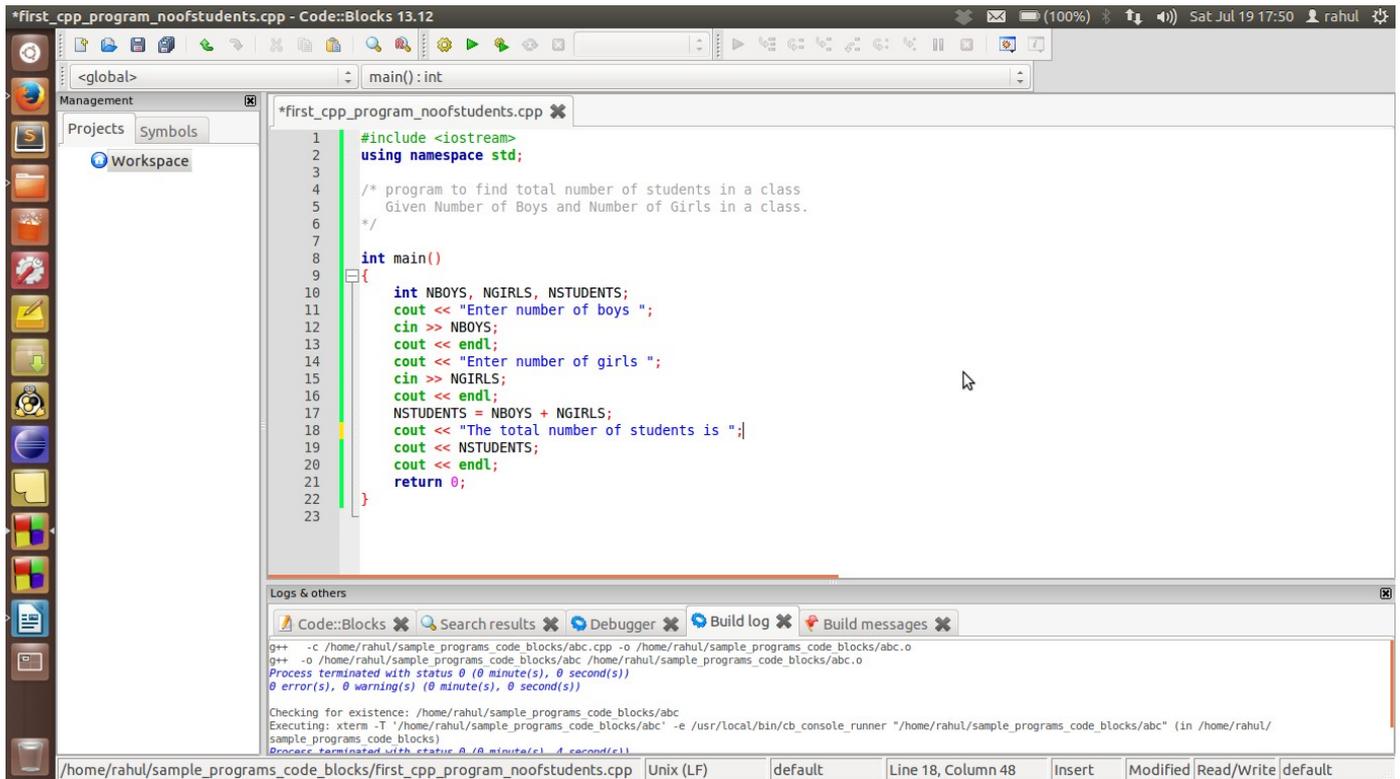
Step-5: Now select directory where you want to save this program and give it an appropriate name. It is recommended that if you are giving long names then separate them out using underscore and not by spaces. After the name give extension as .cpp. After selecting proper directory and name for program, click on save button from dialog box. Do not forget extension '.cpp' after file name.



If there is a '*' before file name in editor, it means the content in editor is not saved. If there is not a '*' before it, means the file's content is safe(saved) on disk.



Step-6: Now write your program in editor's space. Then save it either using the keyboard shortcut Ctrl+s or by going to 'File' from menu toolbar and then by clicking on 'Save file'.

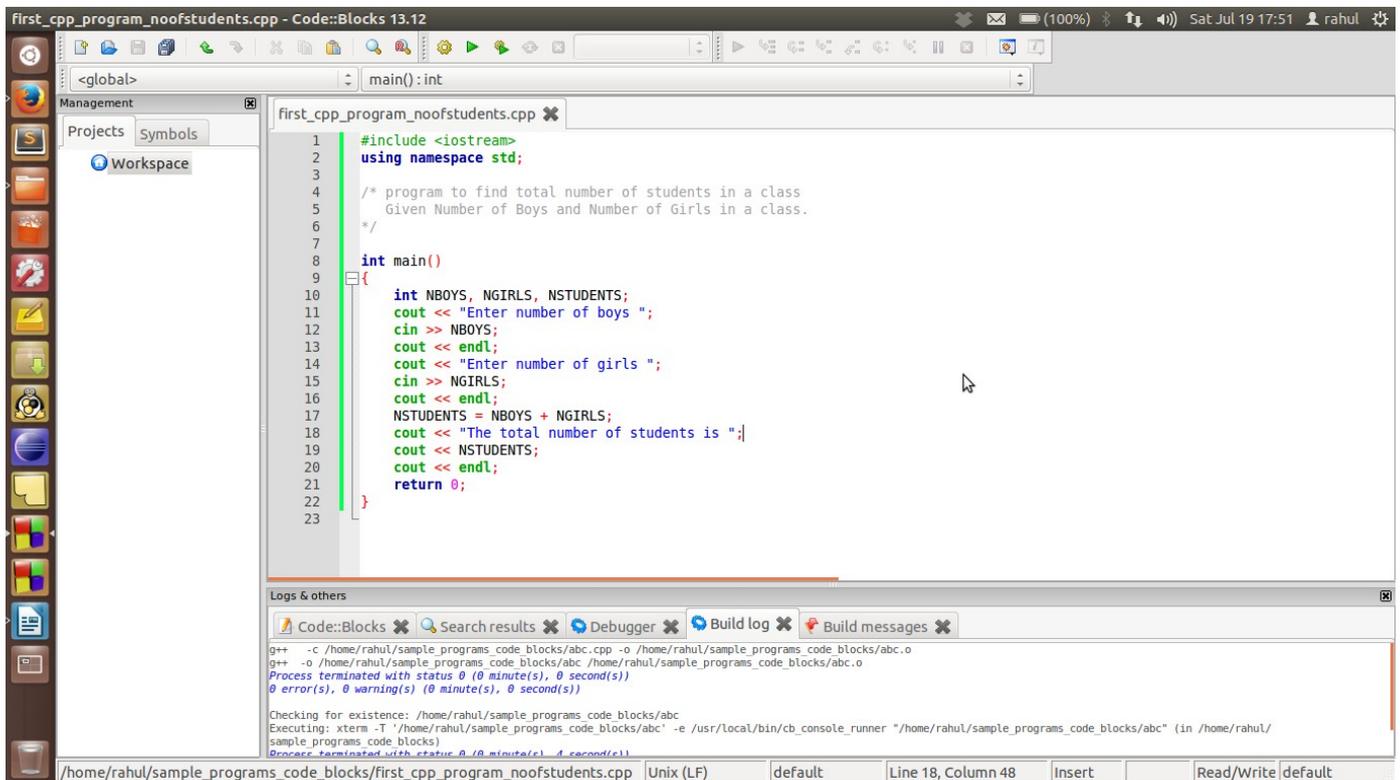


```
1 #include <iostream>
2 using namespace std;
3
4 /* program to find total number of students in a class
5    Given Number of Boys and Number of Girls in a class.
6 */
7
8 int main()
9 {
10     int NBOYS, NGIRLS, NSTUDENTS;
11     cout << "Enter number of boys ";
12     cin >> NBOYS;
13     cout << endl;
14     cout << "Enter number of girls ";
15     cin >> NGIRLS;
16     cout << endl;
17     NSTUDENTS = NBOYS + NGIRLS;
18     cout << "The total number of students is ";
19     cout << NSTUDENTS;
20     cout << endl;
21     return 0;
22 }
23
```

Logs & others

```
Code::Blocks Search results Debugger Build log Build messages
g++ -c /home/rahu/sample_programs_code_blocks/abc.cpp -o /home/rahu/sample_programs_code_blocks/abc.o
g++ -o /home/rahu/sample_programs_code_blocks/abc /home/rahu/sample_programs_code_blocks/abc.o
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
Checking for existence: /home/rahu/sample_programs_code_blocks/abc
Executing: xterm -T /home/rahu/sample_programs_code_blocks/abc -e /usr/local/bin/cb_console_runner "/home/rahu/sample_programs_code_blocks/abc" (in /home/rahu/sample_programs_code_blocks)
Process terminated with status 0 (0 minute(s), 1 second(s))
```

After pressing Ctrl+s, '*' before the file name will go away. Means your file is successfully saved on the disk.



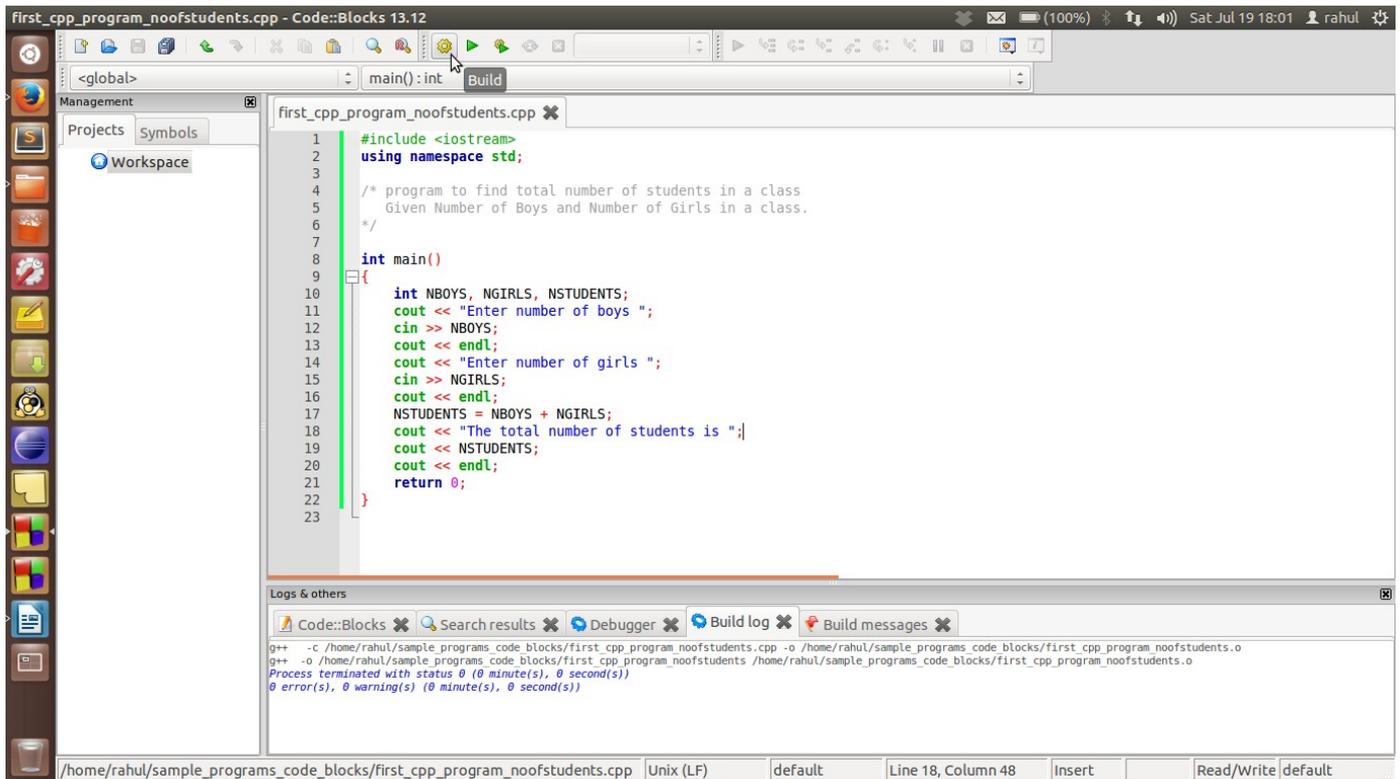
```
1 #include <iostream>
2 using namespace std;
3
4 /* program to find total number of students in a class
5    Given Number of Boys and Number of Girls in a class.
6 */
7
8 int main()
9 {
10     int NBOYS, NGIRLS, NSTUDENTS;
11     cout << "Enter number of boys ";
12     cin >> NBOYS;
13     cout << endl;
14     cout << "Enter number of girls ";
15     cin >> NGIRLS;
16     cout << endl;
17     NSTUDENTS = NBOYS + NGIRLS;
18     cout << "The total number of students is ";
19     cout << NSTUDENTS;
20     cout << endl;
21     return 0;
22 }
23
```

Logs & others

```
Code::Blocks Search results Debugger Build log Build messages
g++ -c /home/rahu/sample_programs_code_blocks/abc.cpp -o /home/rahu/sample_programs_code_blocks/abc.o
g++ -o /home/rahu/sample_programs_code_blocks/abc /home/rahu/sample_programs_code_blocks/abc.o
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
Checking for existence: /home/rahu/sample_programs_code_blocks/abc
Executing: xterm -T /home/rahu/sample_programs_code_blocks/abc -e /usr/local/bin/cb_console_runner "/home/rahu/sample_programs_code_blocks/abc" (in /home/rahu/sample_programs_code_blocks)
Process terminated with status 0 (0 minute(s), 1 second(s))
```

- ✓ After your program is successfully written on disk, you can run and see output by following these steps:
 - From here on file name is considered as 'first_cpp_program_noofstudents' and extension is '.cpp'.

Step-7: Now build your program using 'build' icon given in IDE. Locate the mouse cursor in given screenshot below, and click on build.



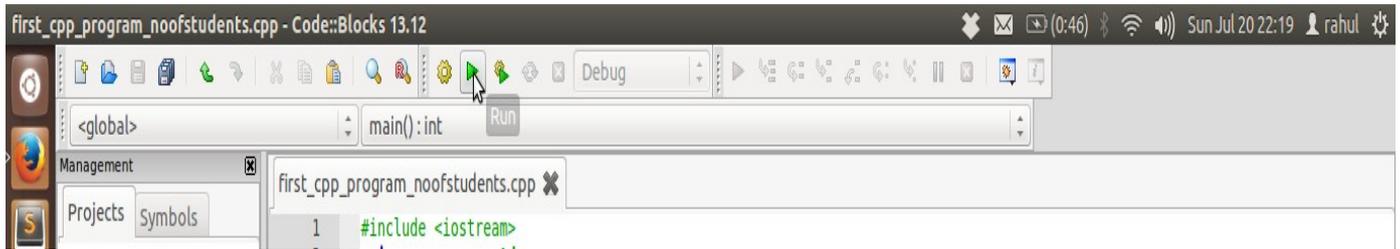
If there are no error in your program. Then at the bottom of IDE in 'Build Log' it will show “ Process terminated with status 0” -- It means your program is error free.

It will be followed by number of errors and warning in the program.
“0 error(s), 0 warning(s)”

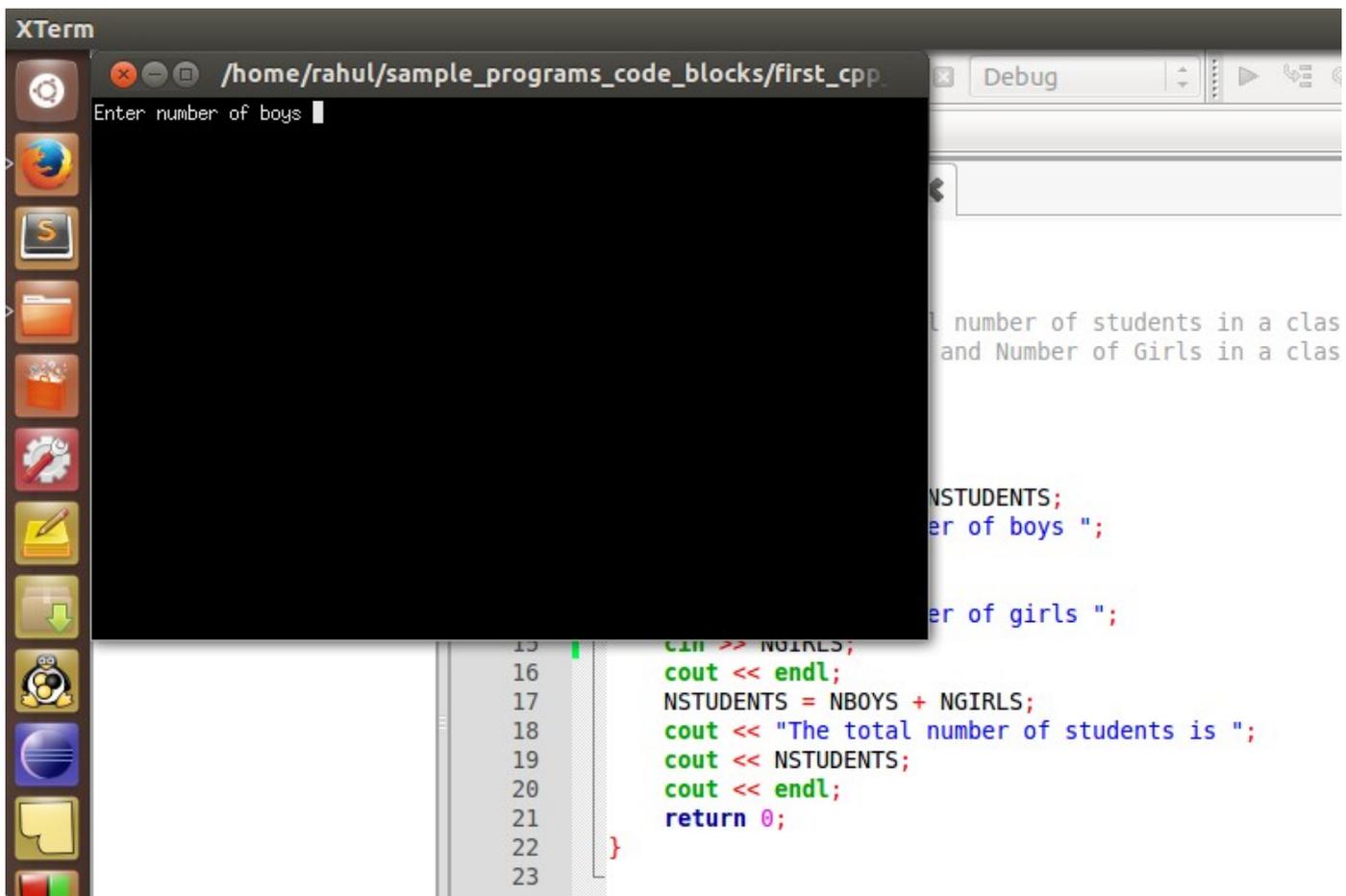
-- If there is an error (or 'are errors') in your program then it will be shown in 'Build log' with line number. After correcting the error(s), follow the step-7 until there is no error in your program.

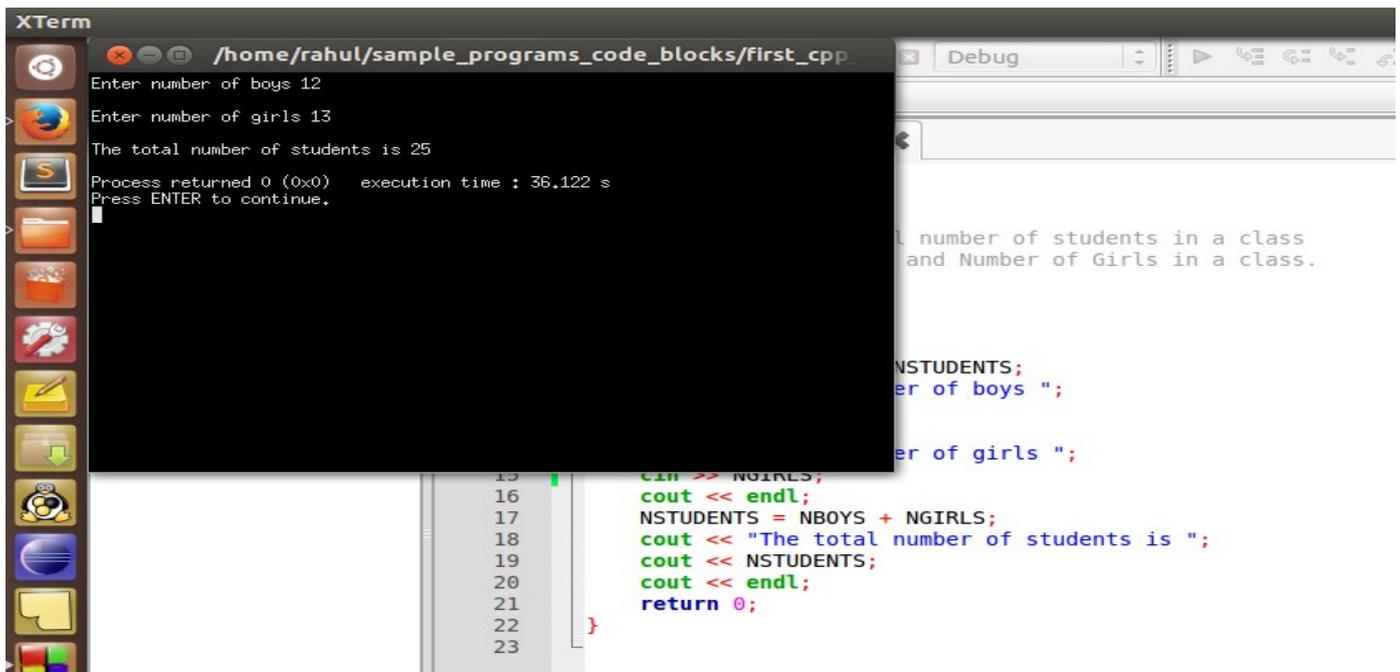
-- Read the error message carefully to understand the error.

Step-8: If your program is error free then you run it by clicking on 'Run' icon as shown below. Locate the mouse pointer to find the icon.



Step-9: After clicking on 'Run', a new terminal window will open. Where you can give input and output will be shown back on the screen. Now put number of boys and press 'Enter'. Again put number of girls and click 'Enter'. Now the total number of students will be shown back on screen. Click 'Enter' again to exit from terminal.

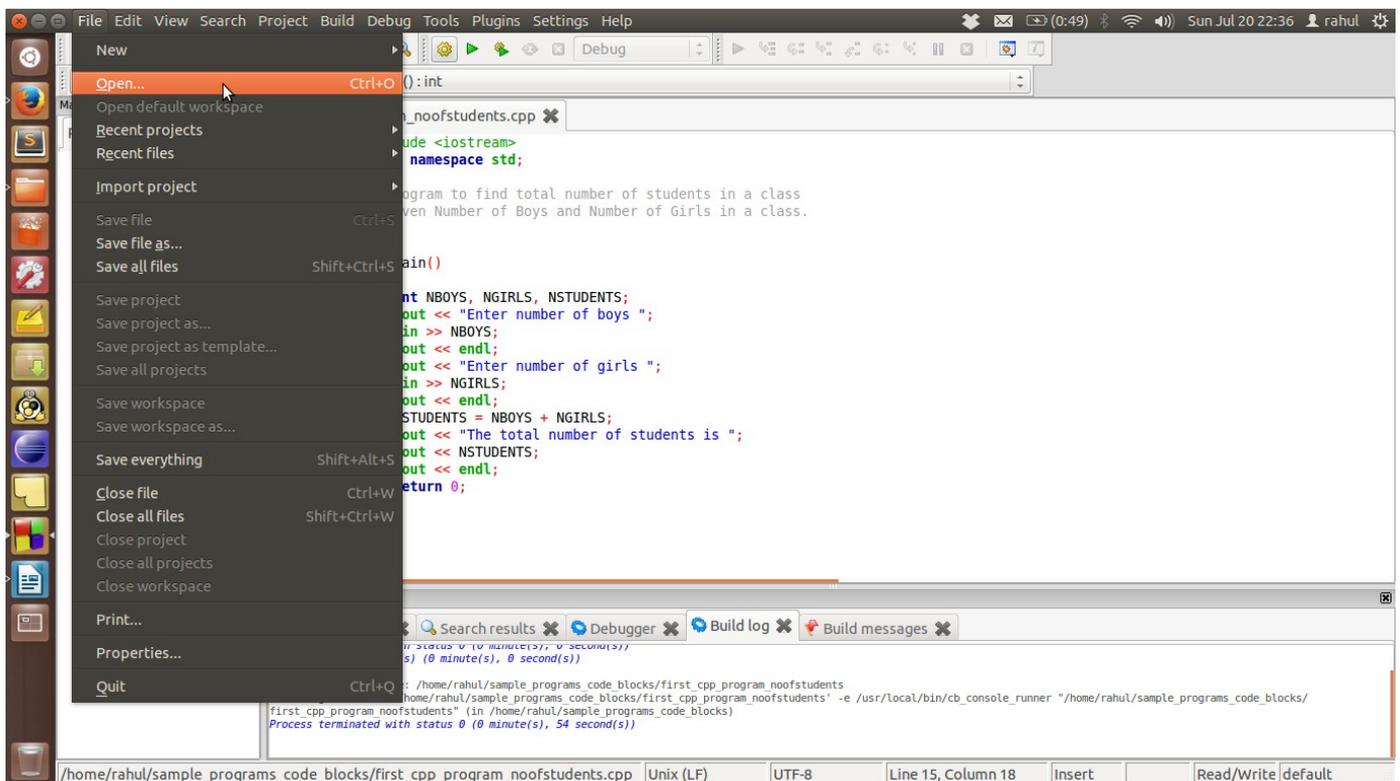




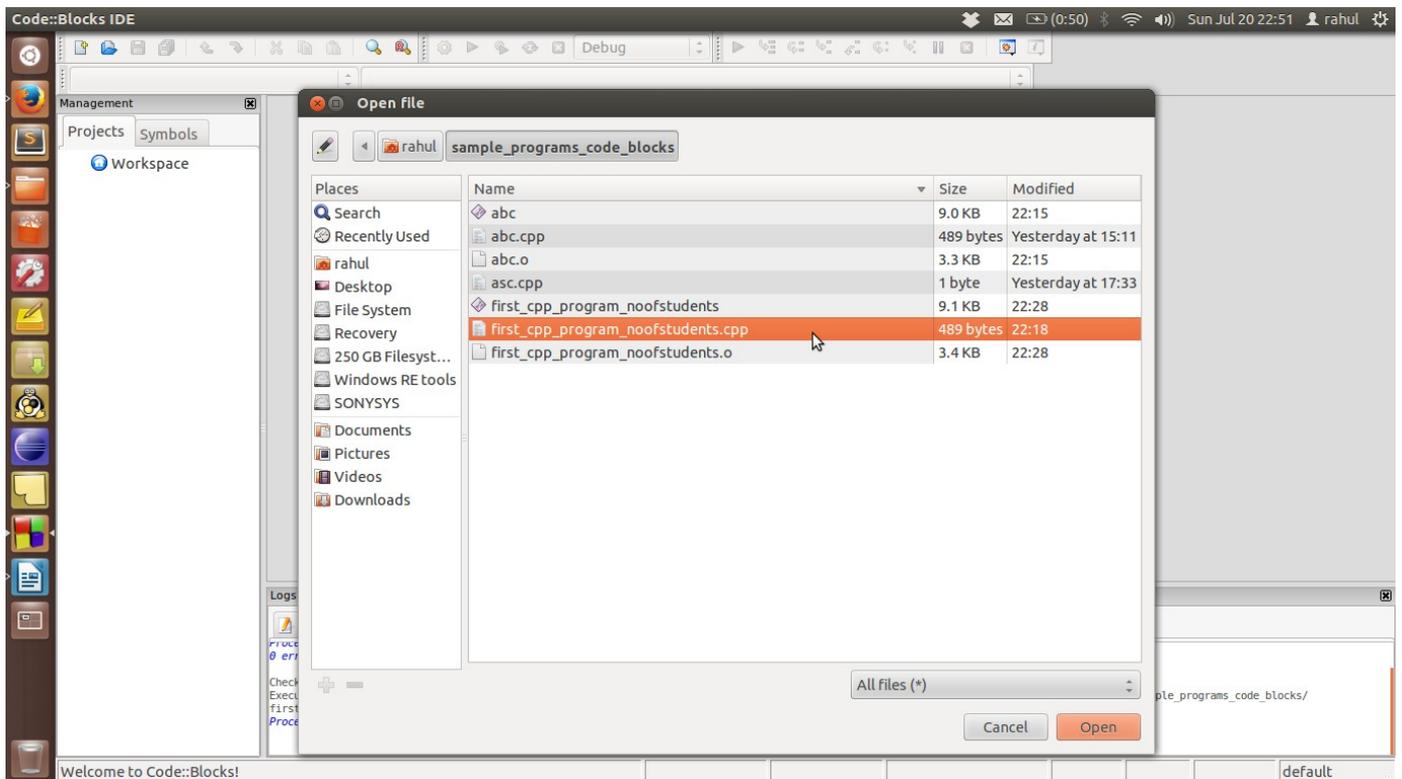
That's it. Your first cpp program is successfully completed.

- ✓ To open an existing program follow simple steps given below.

Step-A: To open an existing file go to 'File' from menu toolbar and click 'Open'. As a shortcut you can also directly use the keyboard shortcut Ctrl+o.



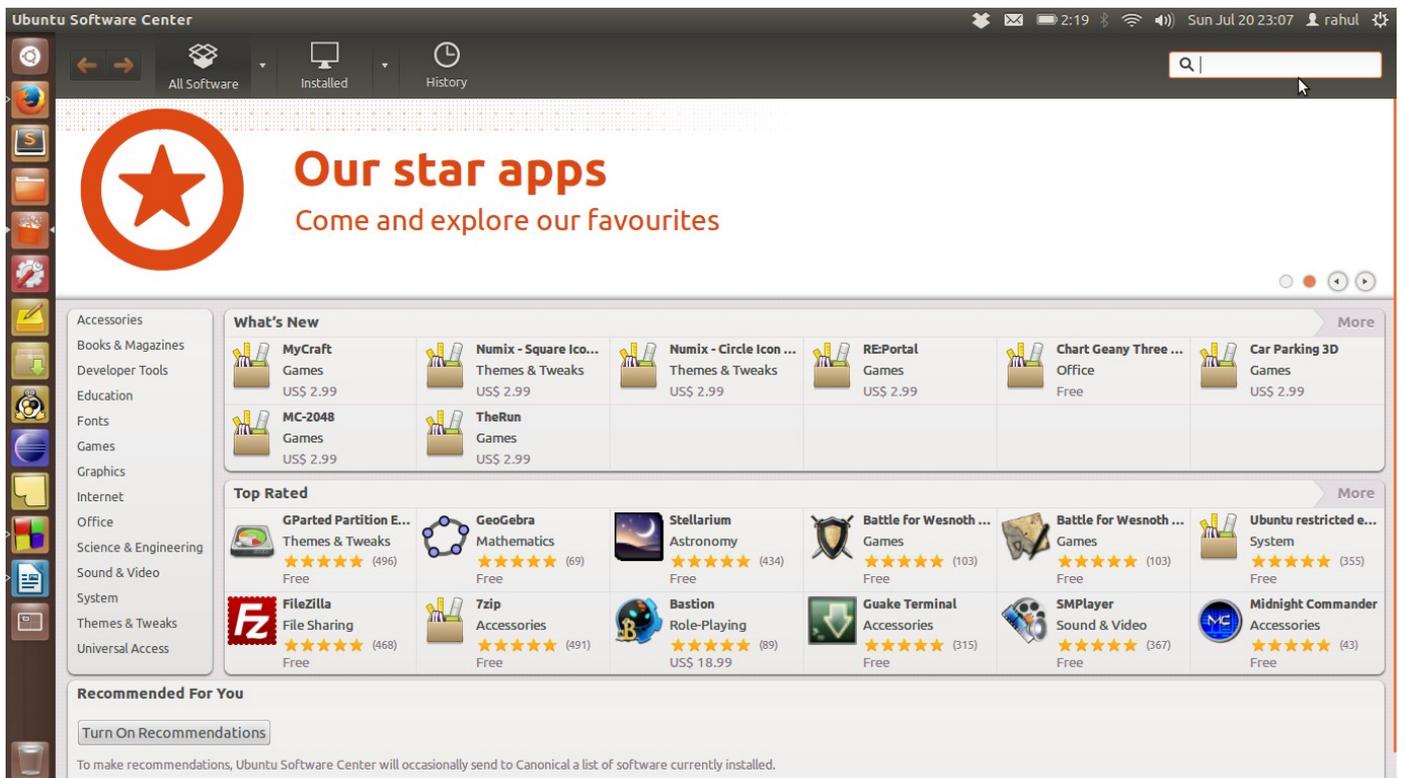
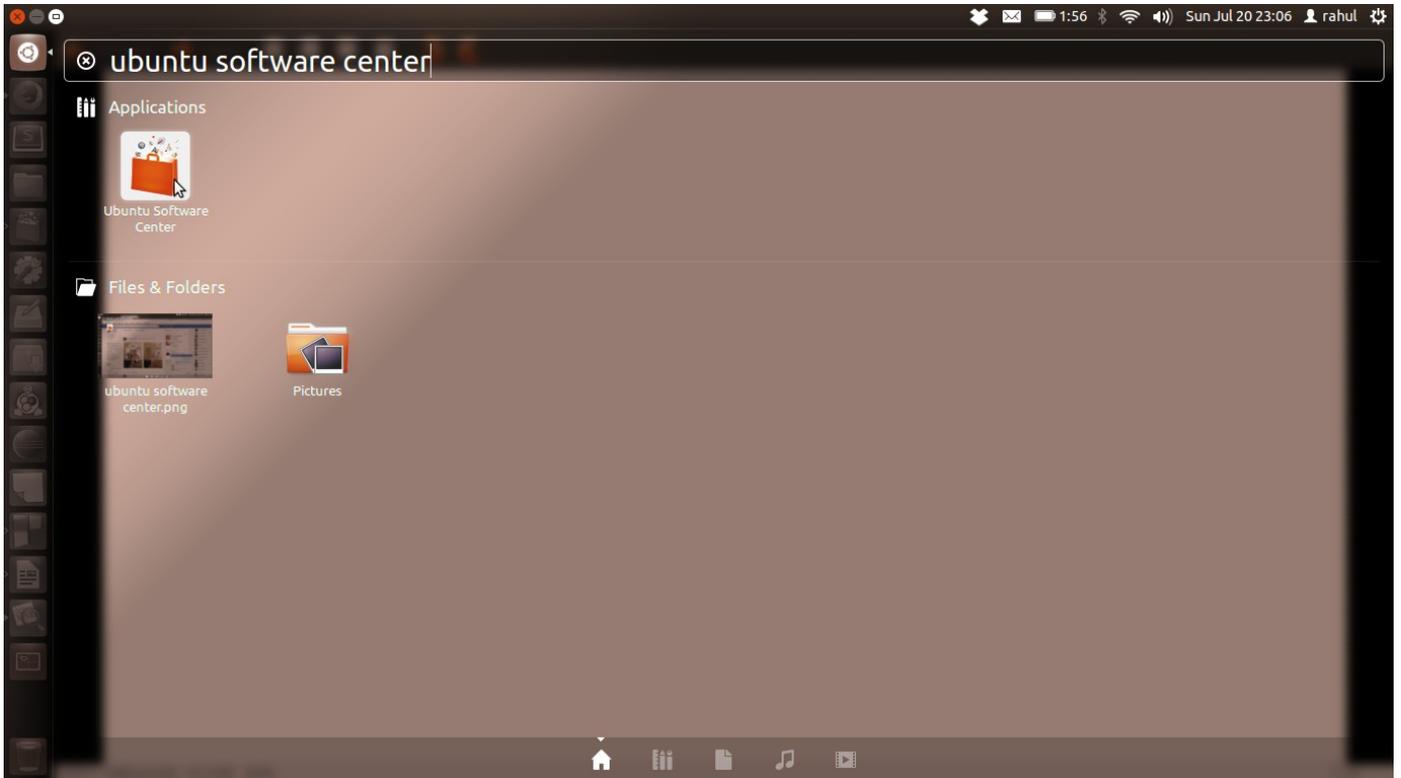
Step-B: It will open a dialog box, where you can choose path where your file actually resides. Now select the file name from dialog box, which will open your file in editor. For building and running your file, follow the steps 7-9 as given above.



Appendix-A -- Guidelines for Installation of Code::Block IDE

✓ Press 'Windows' button  on your keyboard. Now type 'ubuntu software center' in search bar.

Click on icon of 'ubuntu software center'. It will open separate window for 'ubuntu software center'.



- ✓ Now at top of 'ubuntu software center' window type 'code block' in search bar, and press 'Enter'. It will show 'Code::Block IDE' in 'All Software' section. Click on it and then click on 'Install'. It will install Code::Block on your system.

