Name: Jawad Ahmed
24 BS AI 018

Practical # 01

Introduction to DEV C++ IDE

Objective: ***To understand the DEV C++ IDE(Integrated Development Environment) and implement a simple C Program.***

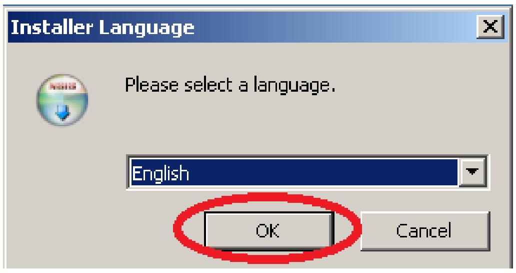
Theory:

The Integrated Development Environment (IDE)

Dev-C++, developed by Bloodshed Software, is a fully featured graphical IDE (Integrated Development Environment), which is able to create Windows or console-based C/C++ programs using the MinGW compiler system. MinGW (Minimalist GNU* for Windows) uses GCC (the GNU g++ compiler collection), which is essentially the same compiler system that is in Cygwin (the unix environment program for Windows) and most versions of Linux.

Installation Steps:

1. Download the installer from the internet. Follow the instructions and install the program. The following screenshots will help you install and run the product:

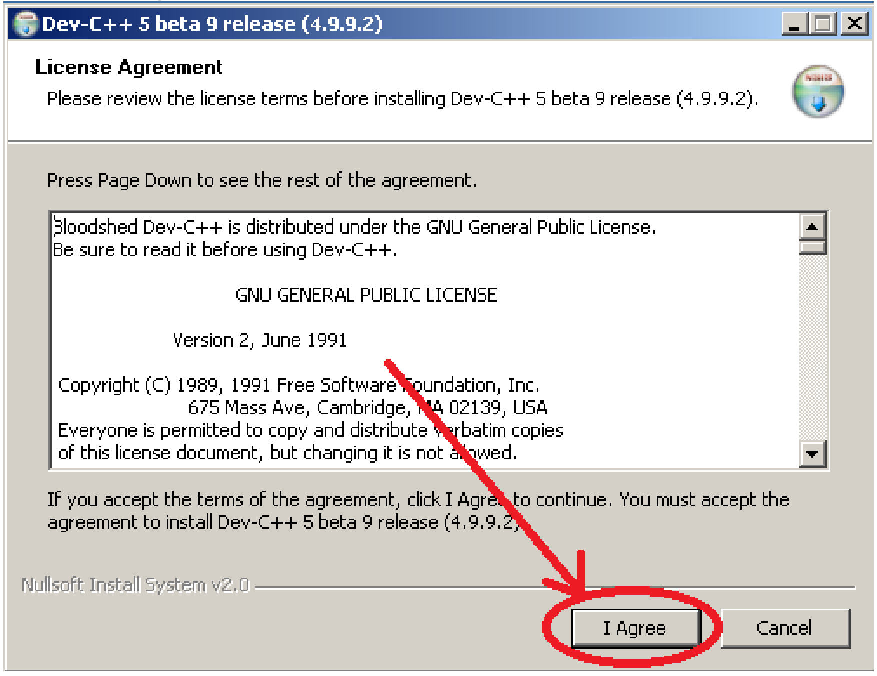


Installer Language

Please select a language.

English

2. License Agreement
   Click on the "I Agree" button to continue



Dev-C++ 5 beta 9 release (4.9.9.2)

License Agreement

Please review the license terms before installing Dev-C++ 5 beta 9 release (4.9.9.2).

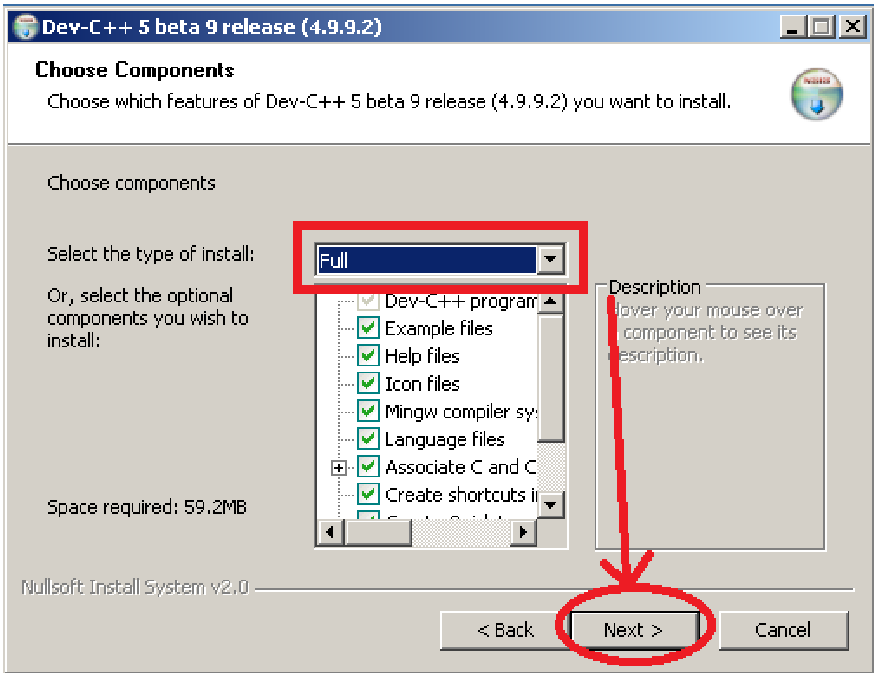
Press Page Down to see the rest of the agreement.

```
Bloodshed Dev-C++ is distributed under the GNU General Public License.
Be sure to read it before using Dev-C++.

                    GNU GENERAL PUBLIC LICENSE

             Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
              675 Mass Ave, Cambridge, MA 02139, USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Dev-C++ 5 beta 9 release (4.9.9.2).

Nullsoft Install System v2.0

3. Choose Components
   Make sure that the type of install is Full and click the Next button to continue



Dev-C++ 5 beta 9 release (4.9.9.2)

Choose Components

Choose which features of Dev-C++ 5 beta 9 release (4.9.9.2) you want to install.

Choose components

Select the type of install: Full

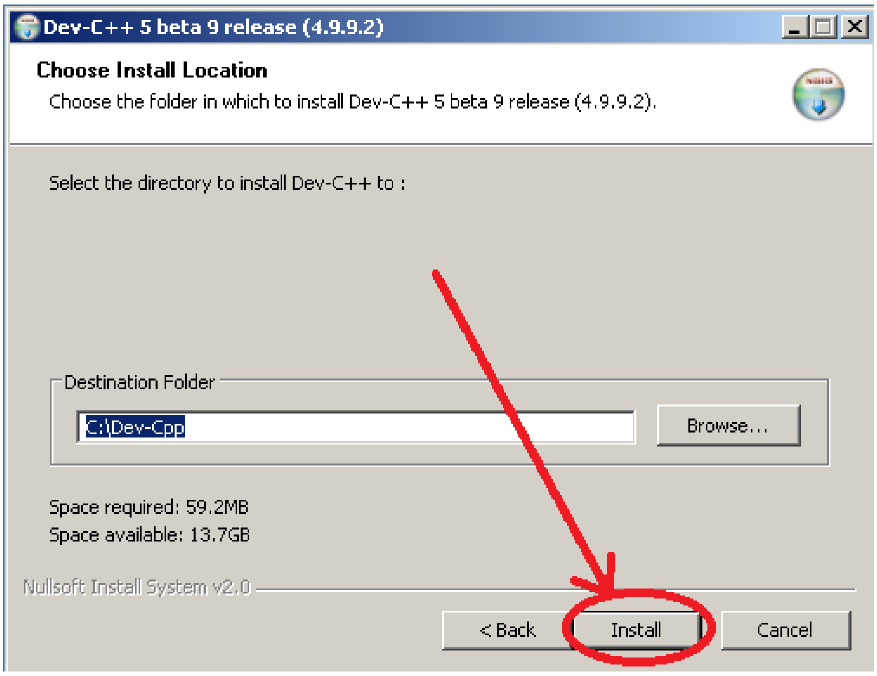
Or, select the optional components you wish to install:

- ☑ Dev-C++ program
- ☑ Example files
- ☑ Help files
- ☑ Icon files
- ☑ Mingw compiler sy:
- ☑ Language files
- ☑ Associate C and C
- ☑ Create shortcuts i

Description
Hover your mouse over a component to see its description.

Space required: 59.2MB

4. Choose Install Location
   Click the Install button to continue



Dev-C++ 5 beta 9 release (4.9.9.2)

Choose Install Location

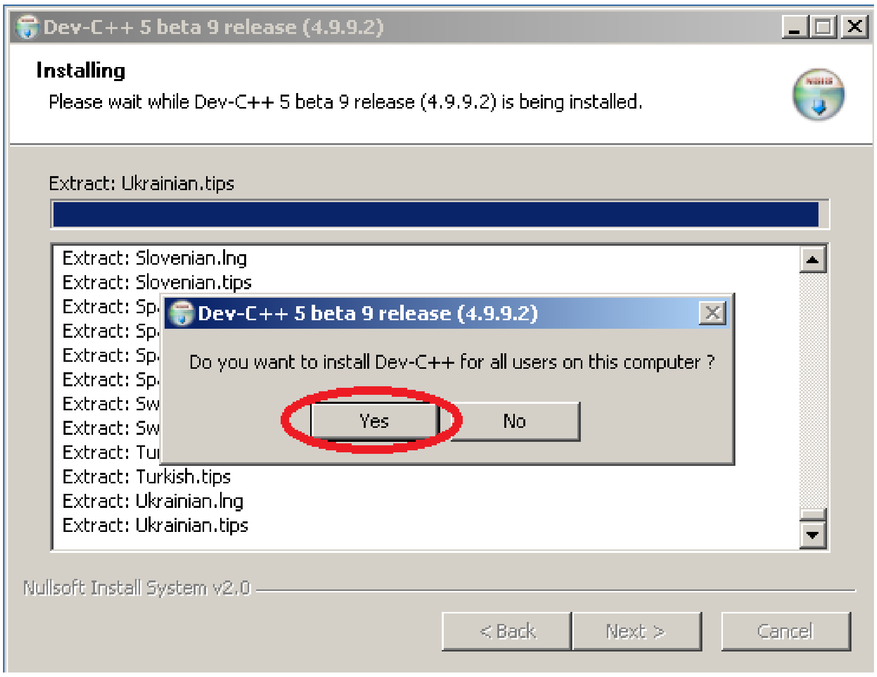
Choose the folder in which to install Dev-C++ 5 beta 9 release (4.9.9.2).

Select the directory to install Dev-C++ to :

Destination Folder: C:\Dev-Cpp

Space required: 59.2MB
Space available: 13.7GB

5. Installing
   Click the Yes button



Dev-C++ 5 beta 9 release (4.9.9.2)

Installing

Please wait while Dev-C++ 5 beta 9 release (4.9.9.2) is being installed.

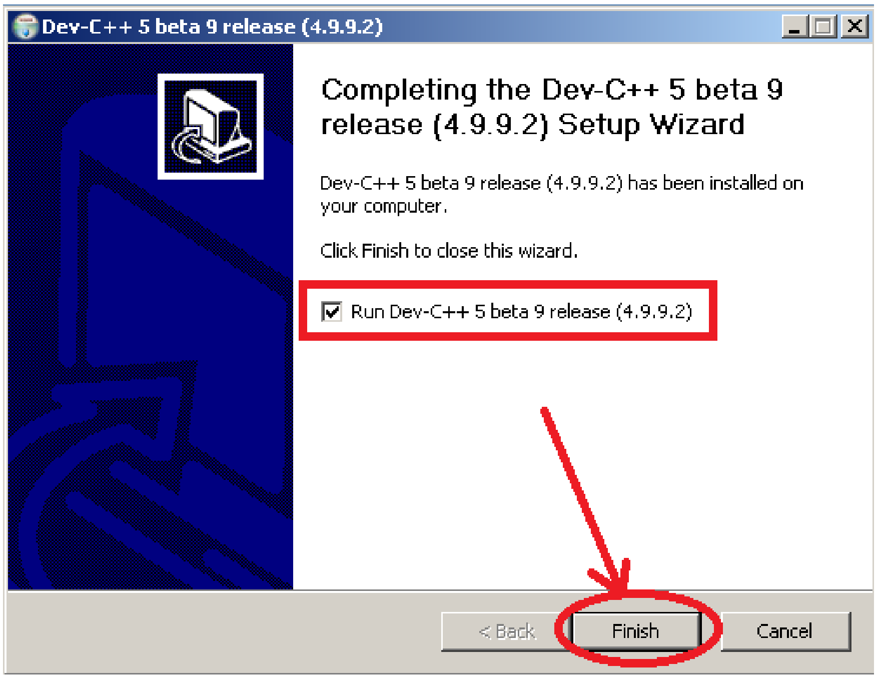
Extract: Ukrainian.tips

```
Extract: Slovenian.lng
Extract: Slovenian.tips
Extract: Sp
Extract: Sp
Extract: Sp
Extract: Sp
Extract: Sw
Extract: Sw
Extract: Tu
Extract: Turkish.tips
Extract: Ukrainian.lng
Extract: Ukrainian.tips
```

Dev-C++ 5 beta 9 release (4.9.9.2)

Do you want to install Dev-C++ for all users on this computer ?

6. Finished
   Click the Finish button to finalize the installation and run the program.



Dev-C++ 5 beta 9 release (4.9.9.2)

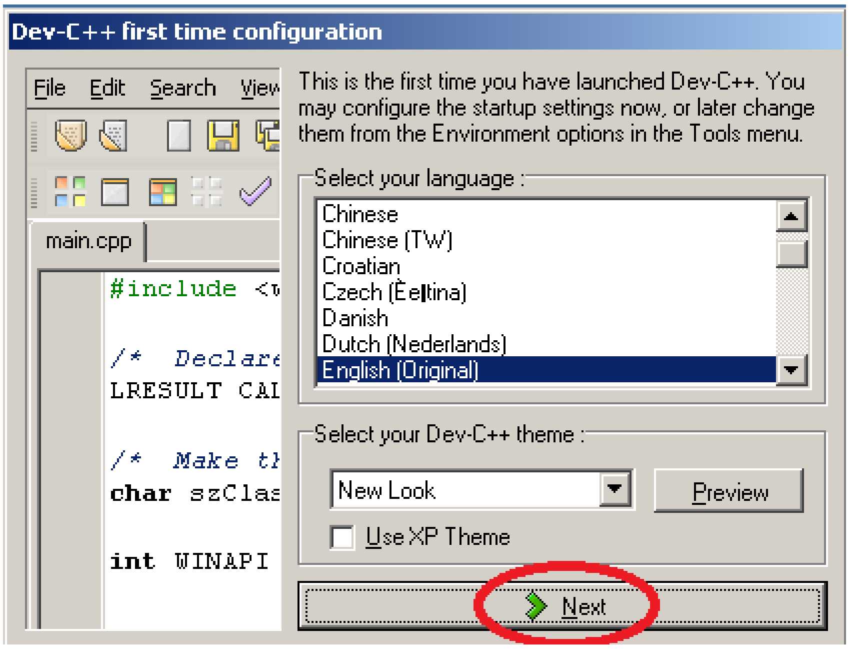
Completing the Dev-C++ 5 beta 9 release (4.9.9.2) Setup Wizard

Dev-C++ 5 beta 9 release (4.9.9.2) has been installed on your computer.

Click Finish to close this wizard.

☑ Run Dev-C++ 5 beta 9 release (4.9.9.2)

7. First Time Configuration
   Click the Next button to continue



```
#include <w

/*  Declare
LRESULT CAL

/*  Make th
char szClas

int WINAPI
```

This is the first time you have launched Dev-C++. You may configure the startup settings now, or later change them from the Environment options in the Tools menu.

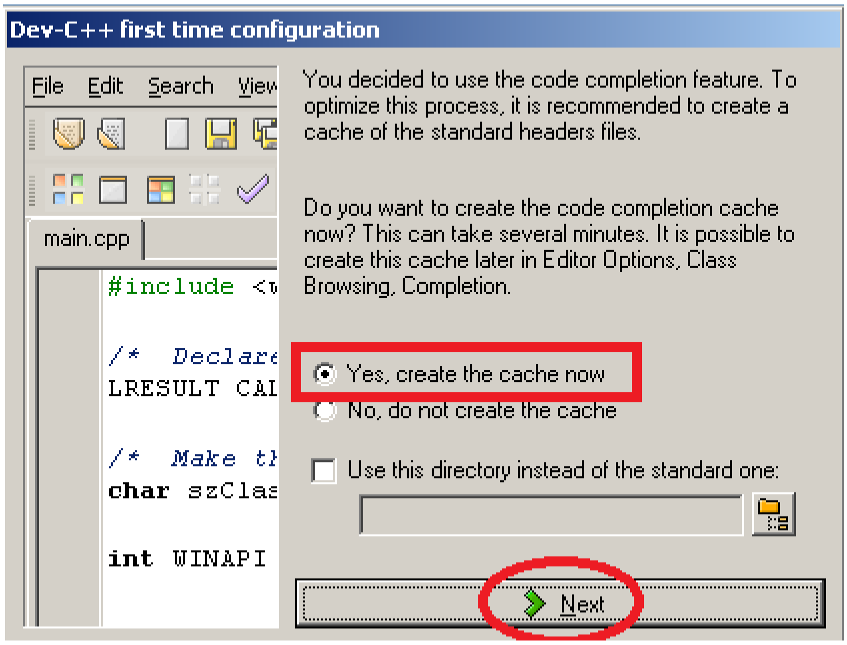
Select your language :

- Chinese
- Chinese (TW)
- Croatian
- Czech (Èeltina)
- Danish
- Dutch (Nederlands)
- English (Original)

Select your Dev-C++ theme : New Look

☐ Use XP Theme

8. First Time Configuration
   Click the Next button to continue



```
#include <w

/*  Declare
LRESULT CAL

/*  Make th
char szClas

int WINAPI
```

You decided to use the code completion feature. To optimize this process, it is recommended to create a cache of the standard headers files.

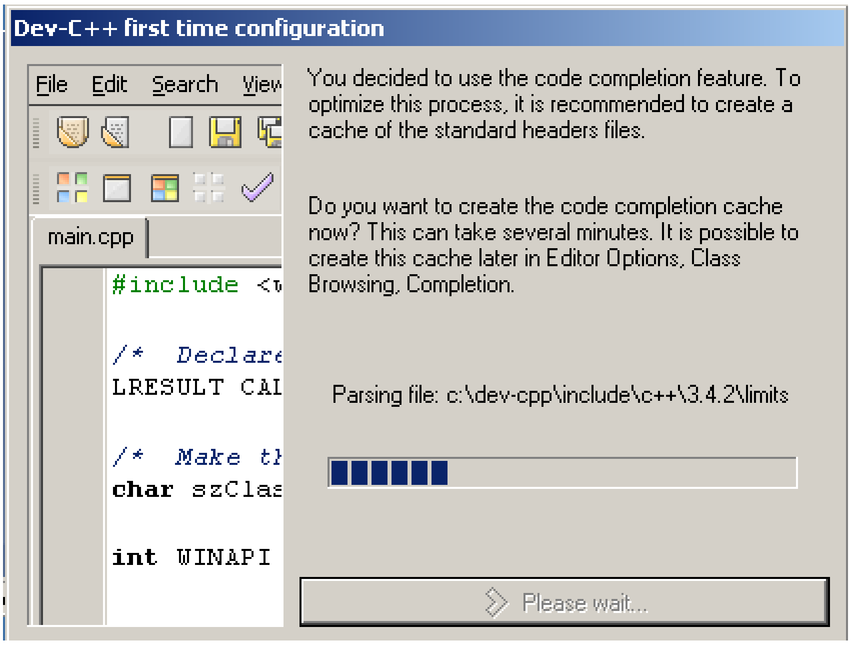
Do you want to create the code completion cache now? This can take several minutes. It is possible to create this cache later in Editor Options, Class Browsing, Completion.

● Yes, create the cache now

○ No, do not create the cache

☐ Use this directory instead of the standard one:

9. First Time Configuration
   Wait for the Progress Bar to Complete



Dev-C++ first time configuration

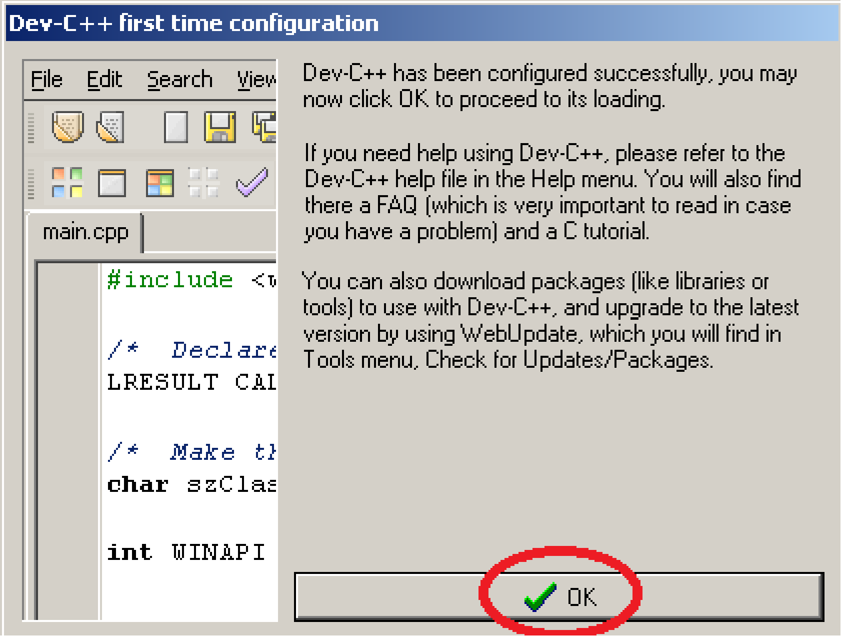
You decided to use the code completion feature. To optimize this process, it is recommended to create a cache of the standard headers files.

Do you want to create the code completion cache now? This can take several minutes. It is possible to create this cache later in Editor Options, Class Browsing, Completion.

Parsing file: c:\dev-cpp\include\c++\3.4.2\limits

```
#include <w

/*  Declare
LRESULT CAL

/*  Make th
char szClas

int  WINAPI
```

10. First Time Configuration
    Click the OK button to Finalize



Dev-C++ first time configuration

Dev-C++ has been configured successfully, you may now click OK to proceed to its loading.

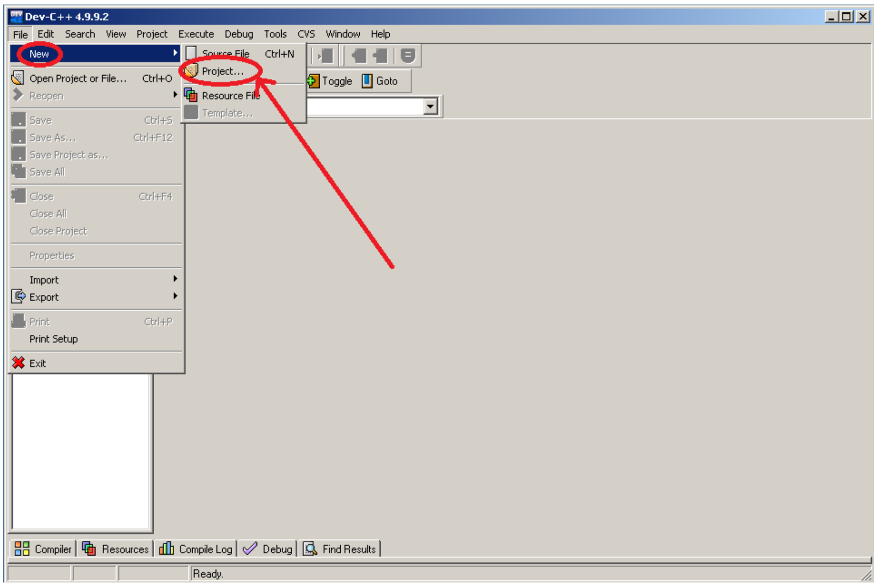
If you need help using Dev-C++, please refer to the Dev-C++ help file in the Help menu. You will also find there a FAQ (which is very important to read in case you have a problem) and a C tutorial.

You can also download packages (like libraries or tools) to use with Dev-C++, and upgrade to the latest version by using WebUpdate, which you will find in Tools menu, Check for Updates/Packages.

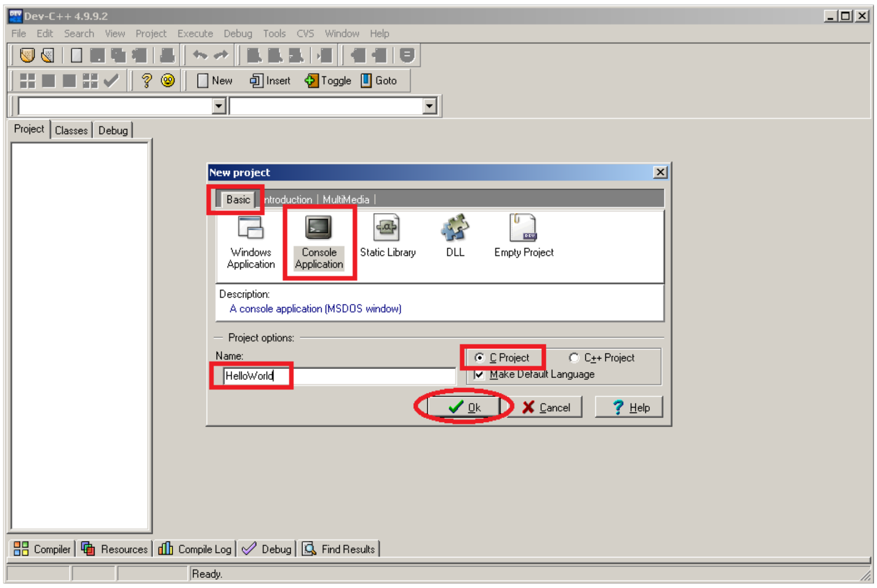
```
#include <w

/*  Declare
LRESULT CAL

/*  Make th
char szClas

int  WINAPI
```

New Project Menu

Click the File menu, then select the New menu item and click the Project menu item.



New Project
On the top, make sure that Basic tab is selected and under the Basic tab, select "Console Application"
Give a name to your project using the Name text box, For instance, "Hello World".
Important: Choose "C Project" under "Project Options", on the left
Click the OK button to create your project

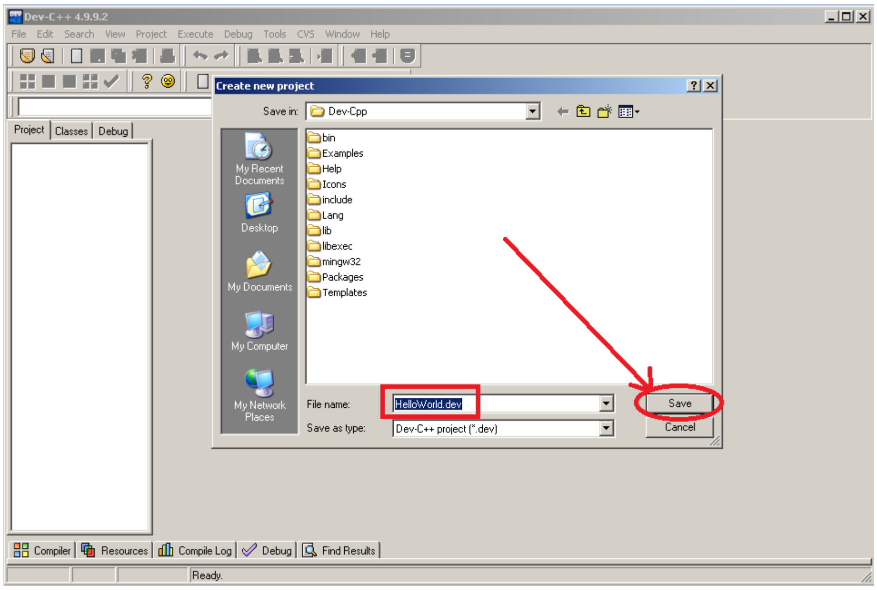


Description:
A console application (MSDOS window)

Project options:
Name: HelloWorld
● C Project ○ C++ Project
☑ Make Default Language

Create New Project

Give a name to your project file and click the Save button to continue



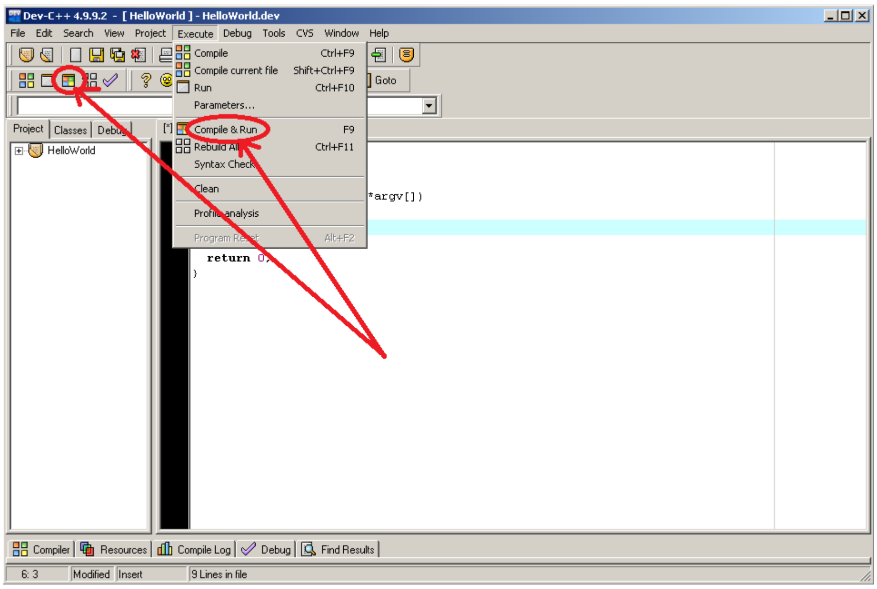
Save in: Dev-Cpp

File name: HelloWorld.dev

Save as type: Dev-C++ project (*.dev)

Compile & Run:

Click “Compile & Run” menu item or the icon displayed in the below screenshot or just Press F9 to compile and run your program.

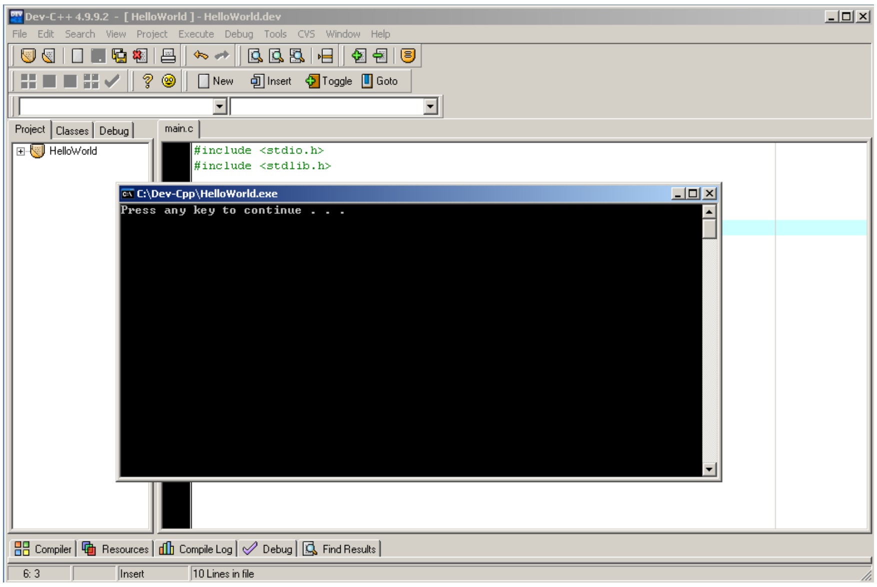


Dev-C++ 4.9.9.2 - [HelloWorld] - HelloWorld.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
Compile Ctrl+F9
Compile current file Shift+Ctrl+F9
Run Ctrl+F10
Parameters...
Compile & Run F9
Rebuild All Ctrl+F11
Syntax Check
Clean
Profile analysis
Program Reset Alt+F2
Goto
Project Classes Debug
HelloWorld
Compiler Resources Compile Log Debug Find Results
6: 3
Modified
Insert
9 Lines in file

```
*argv[])

   return 0;
}
```

Running:

Assuming you did not make any syntax errors on your code, you should see a similar output window running your program.

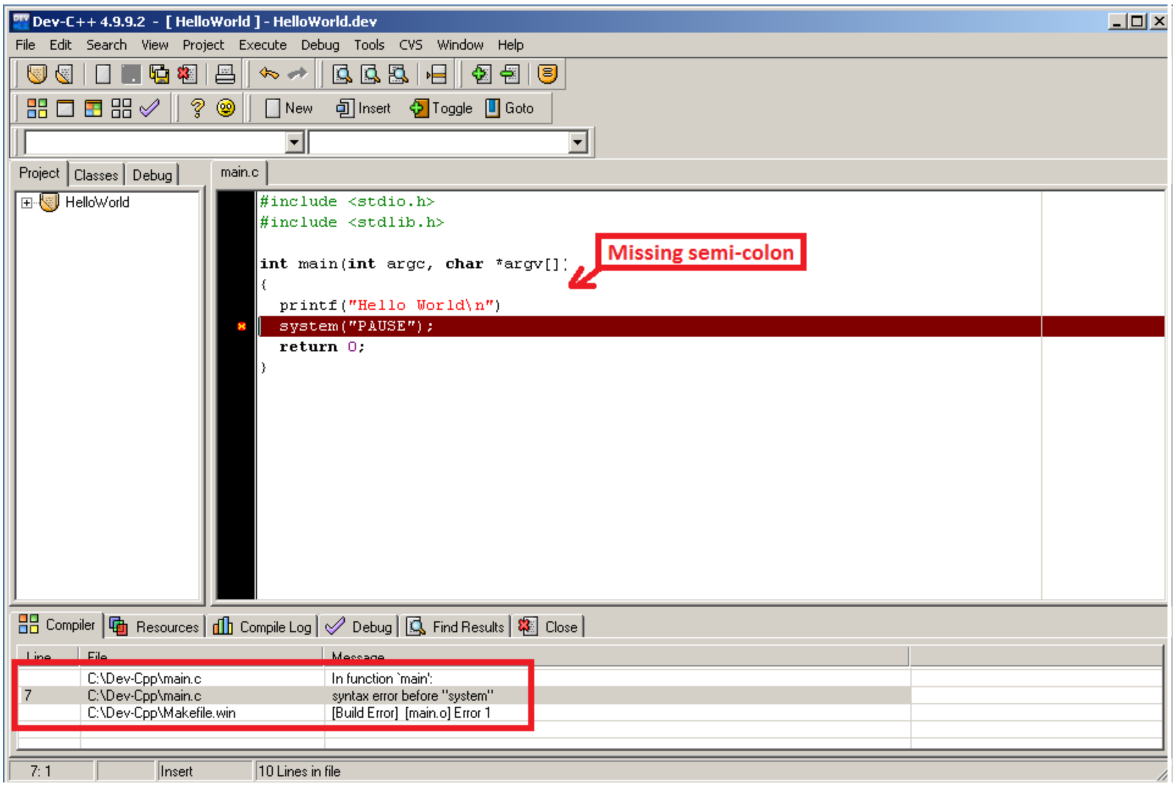

Dev-C++ 4.9.9.2 - [HelloWorld] - HelloWorld.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Project Classes Debug
main.c
HelloWorld
C:\Dev-Cpp\HelloWorld.exe
Compiler Resources Compile Log Debug Find Results
6: 3 Insert 10 Lines in file

```
#include <stdio.h>
#include <stdlib.h>
```

```
Press any key to continue . . .
```

Compile Failed

If you try to compile a code which has syntax errors, Compiler window lists the errors with their line numbers. You can double click the error and see the error highlighted in the code.


Dev-C++ 4.9.9.2 - [HelloWorld] - HelloWorld.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Project Classes Debug
main.c
HelloWorld
Missing semi-colon
Compiler Resources Compile Log Debug Find Results Close
7: 1 Insert 10 Lines in file

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  printf("Hello World\n")
  system("PAUSE");
  return 0;
}
```

Line	File	Message
	C:\Dev-Cpp\main.c	In function `main':
7	C:\Dev-Cpp\main.c	syntax error before "system"
	C:\Dev-Cpp\Makefile.win	[Build Error] [main.o] Error 1

Review Questions/ Exercise:

1. Discuss the steps necessary to produce executable file?

First, go to the File menu and click on "New." A screen will appear, where you should select "Basic Console Application" as the project type. Next, name your file and choose a save location. Once done, your project or executable file will be ready for use. You can then run and debug the file as needed.

2. Discuss the purpose of Compiler & the file needed by compiler?

A compiler translates code written in languages like Java, C, or C++ into machine code. Its primary purpose is to enable developers to write code in a human-readable or human-friendly format. The compiler requires various files as inputs, including source code files (.c, .cpp, .java, .py), header files (.h, .hpp), and library files (.lib). Together, these files provide the essential components needed for the compiler to produce executable code.

3. Discuss the linker & the file needed by the linker?

A linker is a program that combines multiple compiled files, generated by a compiler, into a single executable file. Its primary role is to resolve external references between object files (compiled code) and libraries, enabling a program to run on a computer. Essentially, the linker makes it possible to use functions and variables defined in other files. The linker typically requires the following types of files:

1. *Object File*: This file contains machine code, created by compiling source code written in languages like C or C++. It serves as the compiled version of the original source.

2. *Library File*: Libraries include pre-compiled code that provides commonly used functions, such as standard library functions like printf and scanf.

3. *Header File*: Header files, usually specified at the beginning of a code file, provide declarations of functions and variables. They help the linker identify and link the functions and variables in external files (e.g., files with .h or .hpp extensions).