COMP345:Tutorial 1

Winter 2017

Typical C++ Development Environment

- C++ systems generally consist of three parts: a program development environment, the language and the C++ Standard Library.
- C++ programs typically go through six phases: edit, preprocess, compile, link, load and execute.

Typical C++ Development Environment (Cont.)

- ✓ Phase 1 consists of editing a file with an *editor* program, normally known simply as an editor.

  -Type a C++ program (source code) using the editor such as Visual studio or eclipse.

  -Make any necessary corrections.

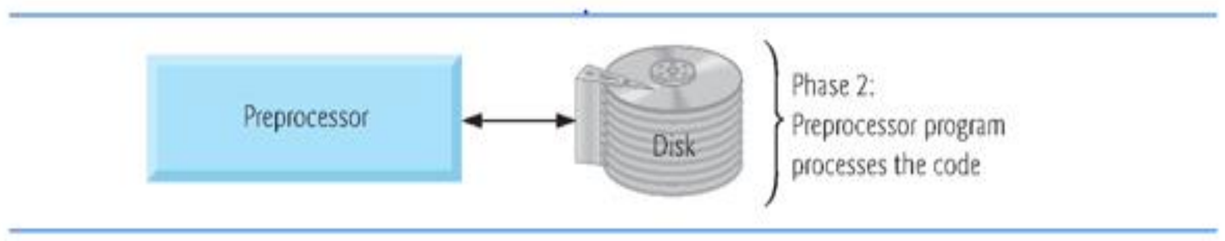
  -Save the program.

  -C++ source code filenames often end with the .cpp, .cxx, .cc or .C extensions (note that C is in uppercase) which indicate that a file contains C++ source code.

Typical C++ Development Environment (Cont.)

- ✓ In phase 2, you give the command to compile the program.
  - -A preprocessor program executes automatically before the compiler's translation phase begins (so we call preprocessing Phase 2 and compiling Phase 3).
  - -The C++ preprocessor obeys commands called preprocessing directives, which indicate that certain manipulations are to be performed on the program before compilation.

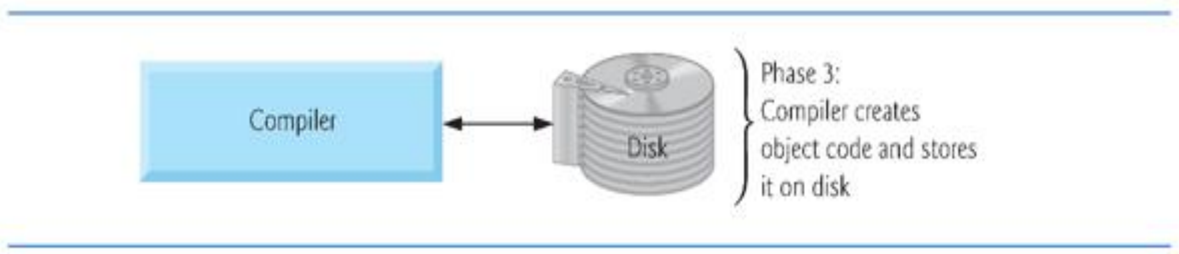
Typical C++ Development Environment- preprocessor phase



Typical C++ Development Environment (Cont.)

- ✓ In Phase 3, the compiler translates the C++ program into machine-language code—also referred to as object code.

Typical C++ Development Environment- compilation phase



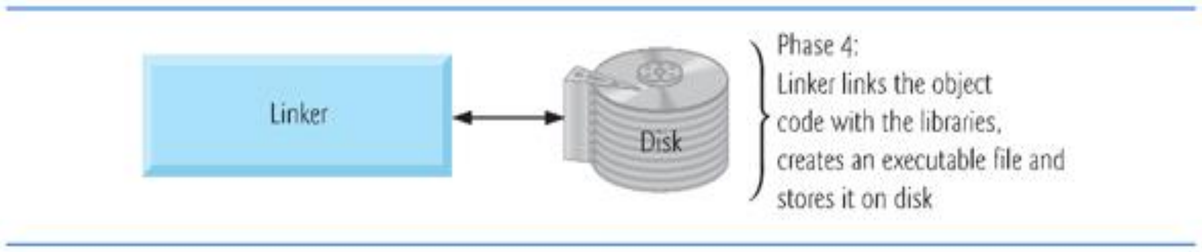
Typical C++ Development Environment (Cont.)

✓ Phase 4 is called linking.

-A linker links the object code with the code for the missing functions to produce an executable program.

-If the program compiles and links correctly, an executable image is produced.

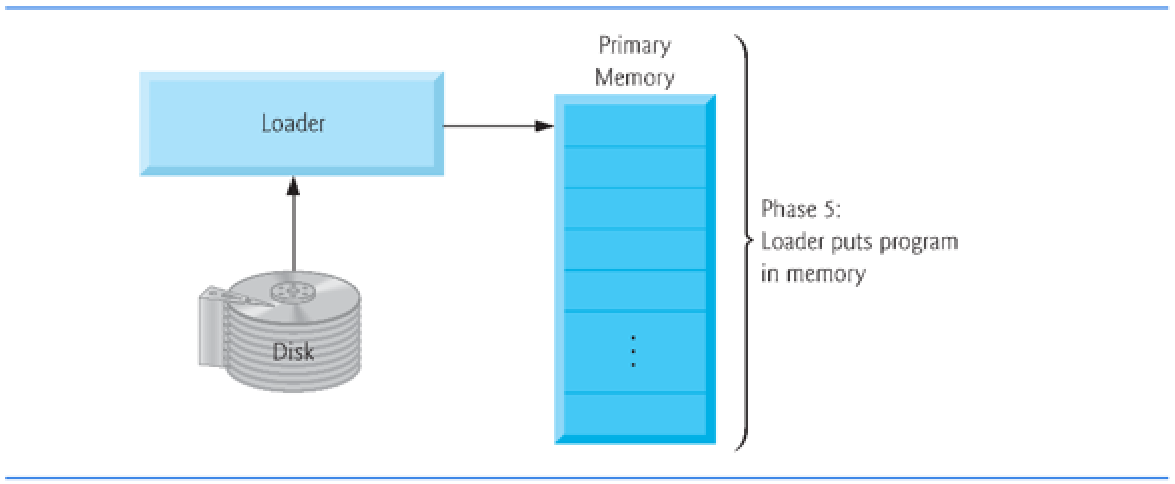
Typical C++ Development Environment- linking phase


Linker
Disk
Phase 4:
Linker links the object code with the libraries, creates an executable file and stores it on disk

Typical C++ Development Environment (Cont.)

- ✓ Phase 5 is called loading.
  - -Before a program can be executed, it must first be placed in memory.
  - -This is done by the loader, which takes the executable image from disk and transfers it to memory.
  - -Additional components from shared libraries that support the program are also loaded.

Typical C++ Development Environment- loading phase



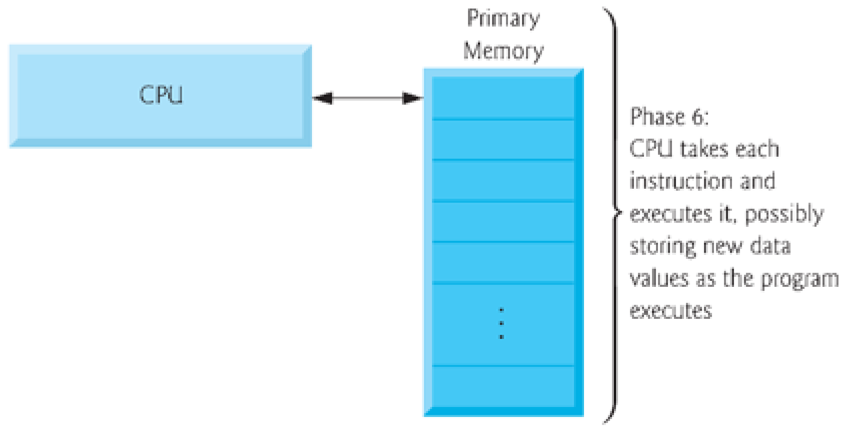
Typical C++ Development Environment (Cont.)

- ✓ Phase 6: Execution

  -Finally, the computer, under the control of its CPU, executes the program one instruction at a time.

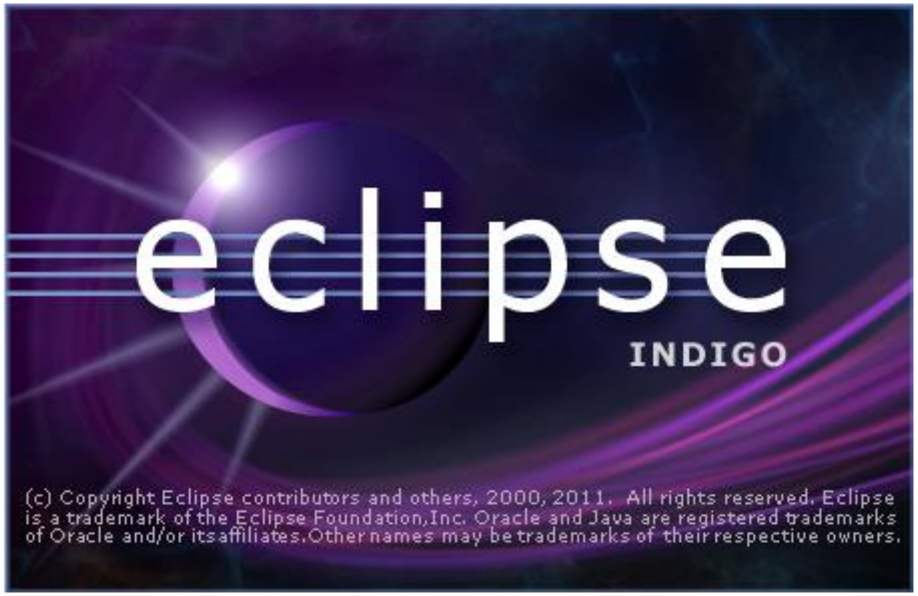
  -Some modern computer architectures can execute several instructions in parallel.

Typical C++ Development Environment- execution phase

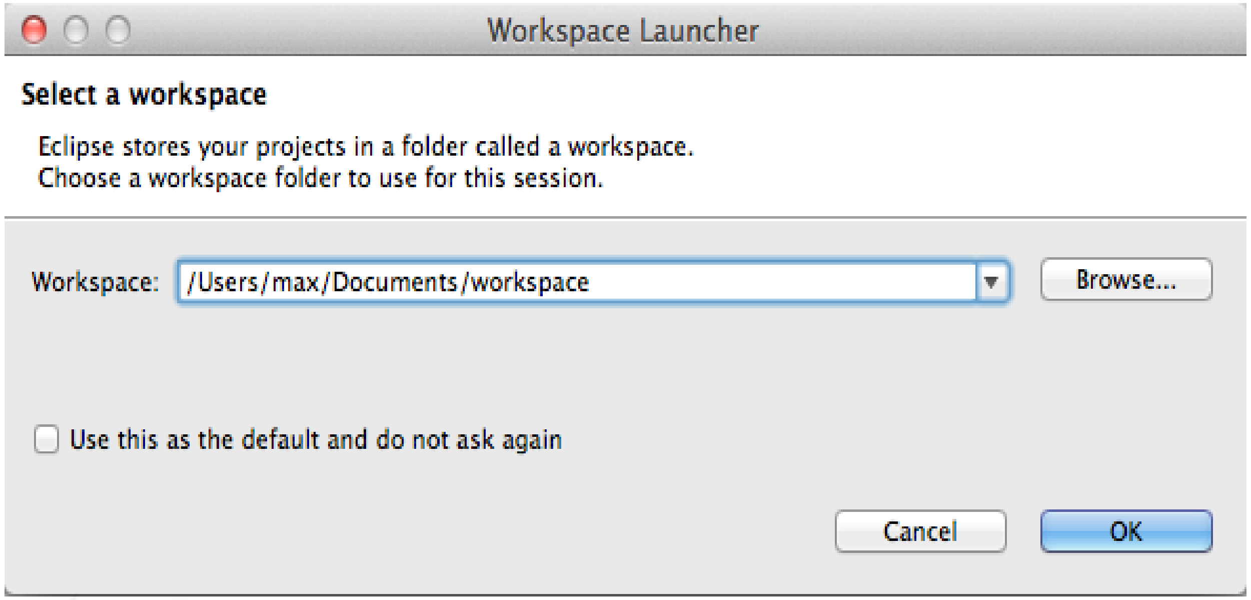

Primary Memory
CPU
Phase 6:
CPU takes each instruction and executes it, possibly storing new data values as the program executes

First Run of Eclipse

After installing eclipse, you should be able to run it.


eclipse
INDIGO
(c) Copyright Eclipse contributors and others, 2000, 2011. All rights reserved. Eclipse is a trademark of the Eclipse Foundation,Inc. Oracle and Java are registered trademarks of Oracle and/or its affiliates.Other names may be trademarks of their respective owners.

Immediately after that, Eclipse will ask you for your workspace location. It defaults to a place within your personal settings. It is a good idea to use the default workspace, you may want to note where it is located.

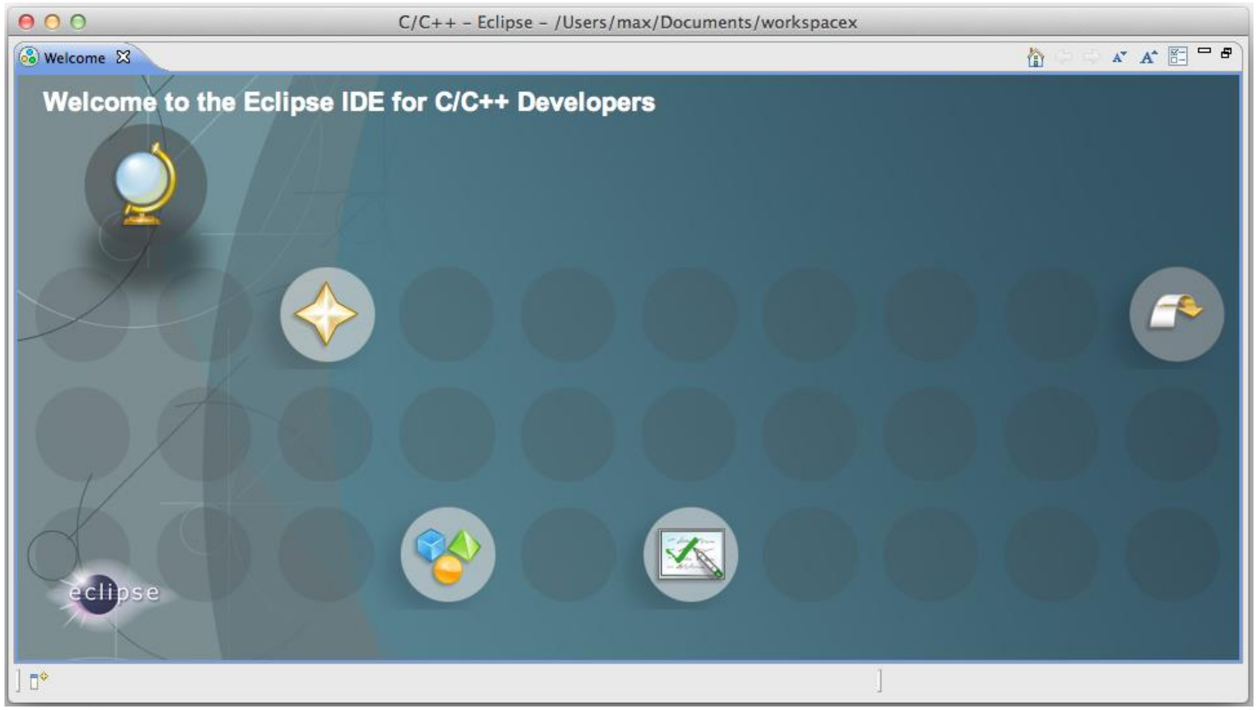


Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: /Users/max/Documents/workspace

If you always want to use the same workspace, you may select the Use this as the default... and you'll never have to worry about workspaces again. This is usually a good idea once you've used Eclipse for a while. Finally Eclipse starts up with the welcome screen:


C/C++ – Eclipse – /Users/max/Documents/workspacex
Welcome
Welcome to the Eclipse IDE for C/C++ Developers
eclipse

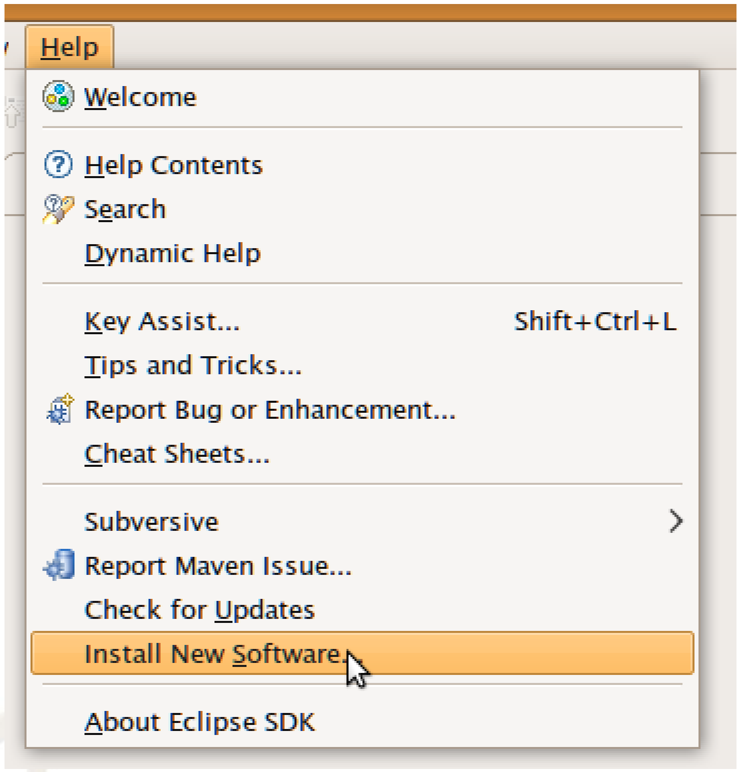
And if you select the "Go to the workbench" (the backward arrow) on the right, then you are inside Eclipse.

- If you downloaded the "Eclipse IDE for C/C++ Developers" you can skip the next section and go straight to the section called “Configuring the CDT”.

- If you have downloaded a different package, there are additional tools needed to start programming. We want to develop in C++, so we will continue with the next section.

Installing the CDT

In the "Help" menu select "Install New Software..."



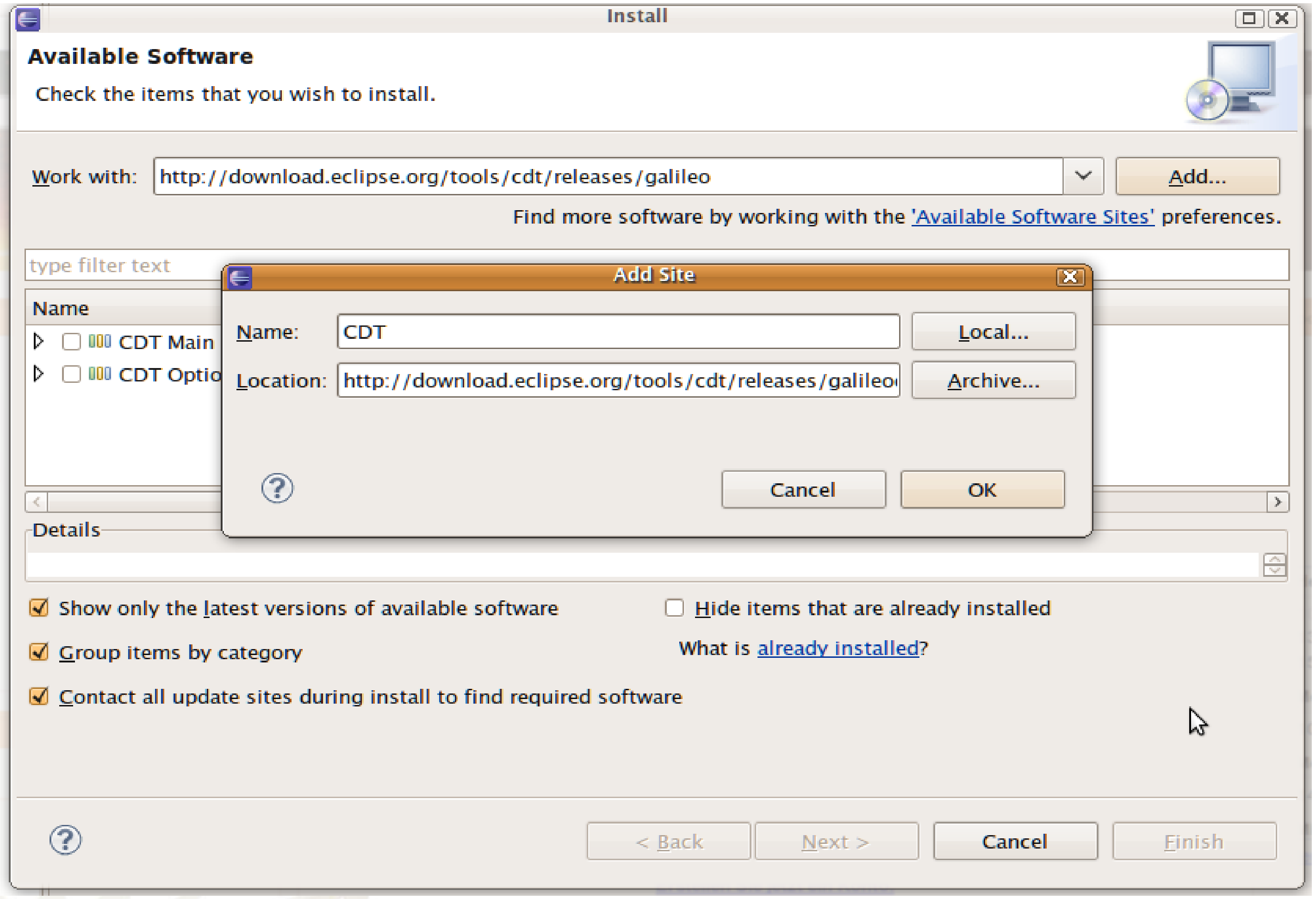
This will show you the list of available software update sites. CDT is part of the standard release, so you can select an update site matching your eclipse version.

Please note: It is important to install the right Version of CDT for your version of eclipse; for example:

-For Eclipse 3.7 (Indigo): Use http://download.eclipse.org/tools/cdt/releases/indigo

-For Eclipse 3.6 (Helios): Use http://download.eclipse.org/tools/cdt/releases/helios

-For Eclipse 3.5 (Galileo): Use http://download.eclipse.org/tools/cdt/releases/galileo



Install

Available Software

Check the items that you wish to install.

Work with: http://download.eclipse.org/tools/cdt/releases/galileo

Find more software by working with the 'Available Software Sites' preferences.

Add Site

Name: CDT

Location: http://download.eclipse.org/tools/cdt/releases/galileo

☑ Show only the latest versions of available software

☐ Hide items that are already installed

☑ Group items by category

What is already installed?

☑ Contact all update sites during install to find required software

When you have found the CDT site, it will give you two entries: CDT Main Features and CDT Optional Features. Expand both, and find the latest version of the CDT. Make sure you select at least the following:

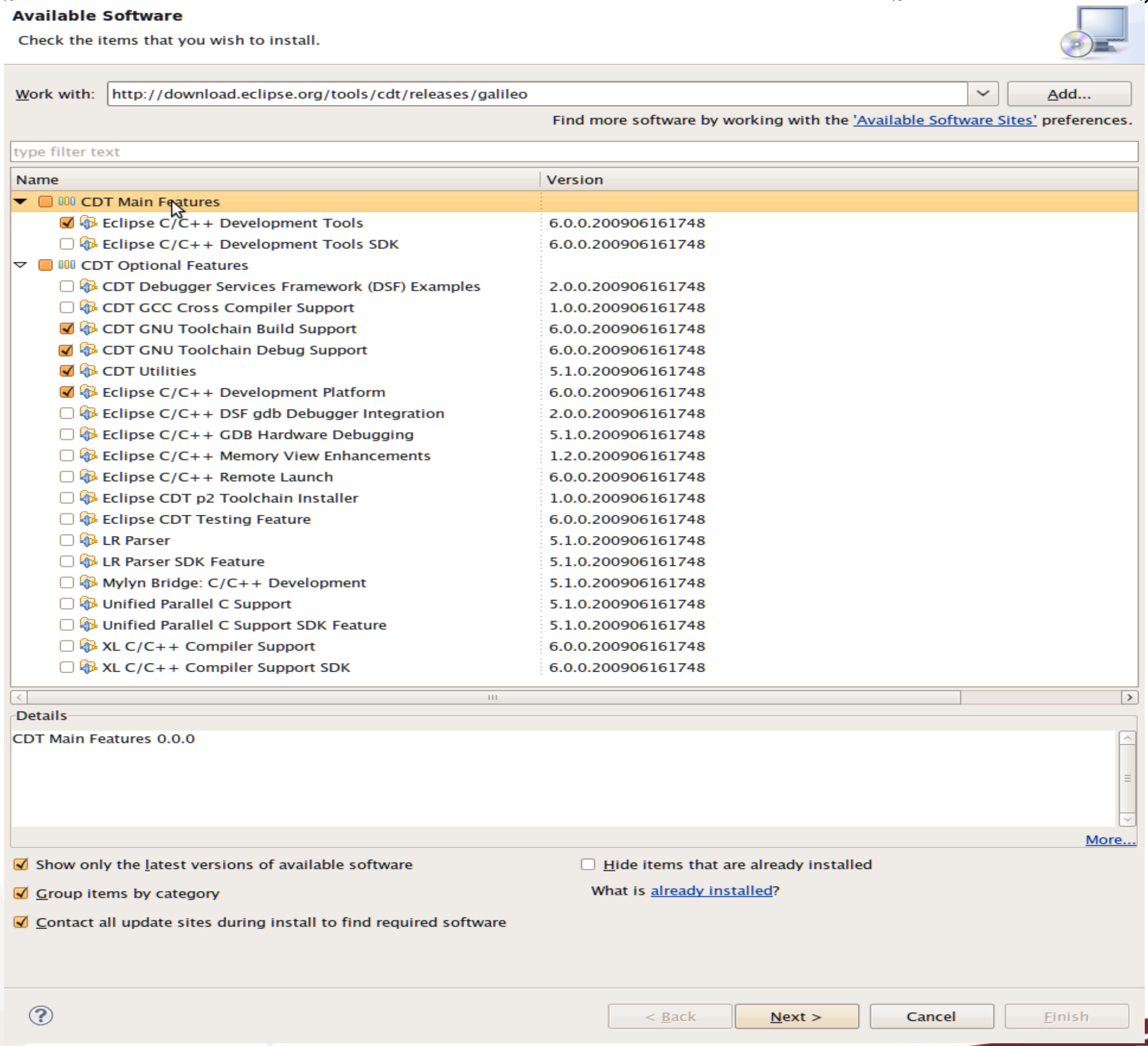
C/C++ Development Tools

C/C++ GNU Toolchain Build Support

C/C++ GNU Toolchain Debug Support

C/C++ Development Platform

Do not select all items! Some of these require dependencies from other projects, which may not be installed and thus fail to install. Select only the features you need!



Available Software

Check the items that you wish to install.

Work with: http://download.eclipse.org/tools/cdt/releases/galileo

Find more software by working with the 'Available Software Sites' preferences.

type filter text

Name	Version
CDT Main Features	
✓ Eclipse C/C++ Development Tools	6.0.0.200906161748
Eclipse C/C++ Development Tools SDK	6.0.0.200906161748
CDT Optional Features	
CDT Debugger Services Framework (DSF) Examples	2.0.0.200906161748
CDT GCC Cross Compiler Support	1.0.0.200906161748
✓ CDT GNU Toolchain Build Support	6.0.0.200906161748
✓ CDT GNU Toolchain Debug Support	6.0.0.200906161748
✓ CDT Utilities	5.1.0.200906161748
✓ Eclipse C/C++ Development Platform	6.0.0.200906161748
Eclipse C/C++ DSF gdb Debugger Integration	2.0.0.200906161748
Eclipse C/C++ GDB Hardware Debugging	5.1.0.200906161748
Eclipse C/C++ Memory View Enhancements	1.2.0.200906161748
Eclipse C/C++ Remote Launch	6.0.0.200906161748
Eclipse CDT p2 Toolchain Installer	1.0.0.200906161748
Eclipse CDT Testing Feature	6.0.0.200906161748
LR Parser	5.1.0.200906161748
LR Parser SDK Feature	5.1.0.200906161748
Mylyn Bridge: C/C++ Development	5.1.0.200906161748
Unified Parallel C Support	5.1.0.200906161748
Unified Parallel C Support SDK Feature	5.1.0.200906161748
XL C/C++ Compiler Support	6.0.0.200906161748
XL C/C++ Compiler Support SDK	6.0.0.200906161748

Details

CDT Main Features 0.0.0

More...

✓ Show only the latest versions of available software

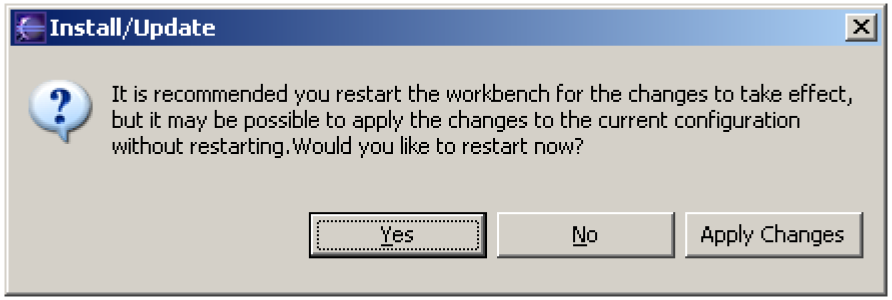
✓ Group items by category

✓ Contact all update sites during install to find required software

Hide items that are already installed

What is already installed?

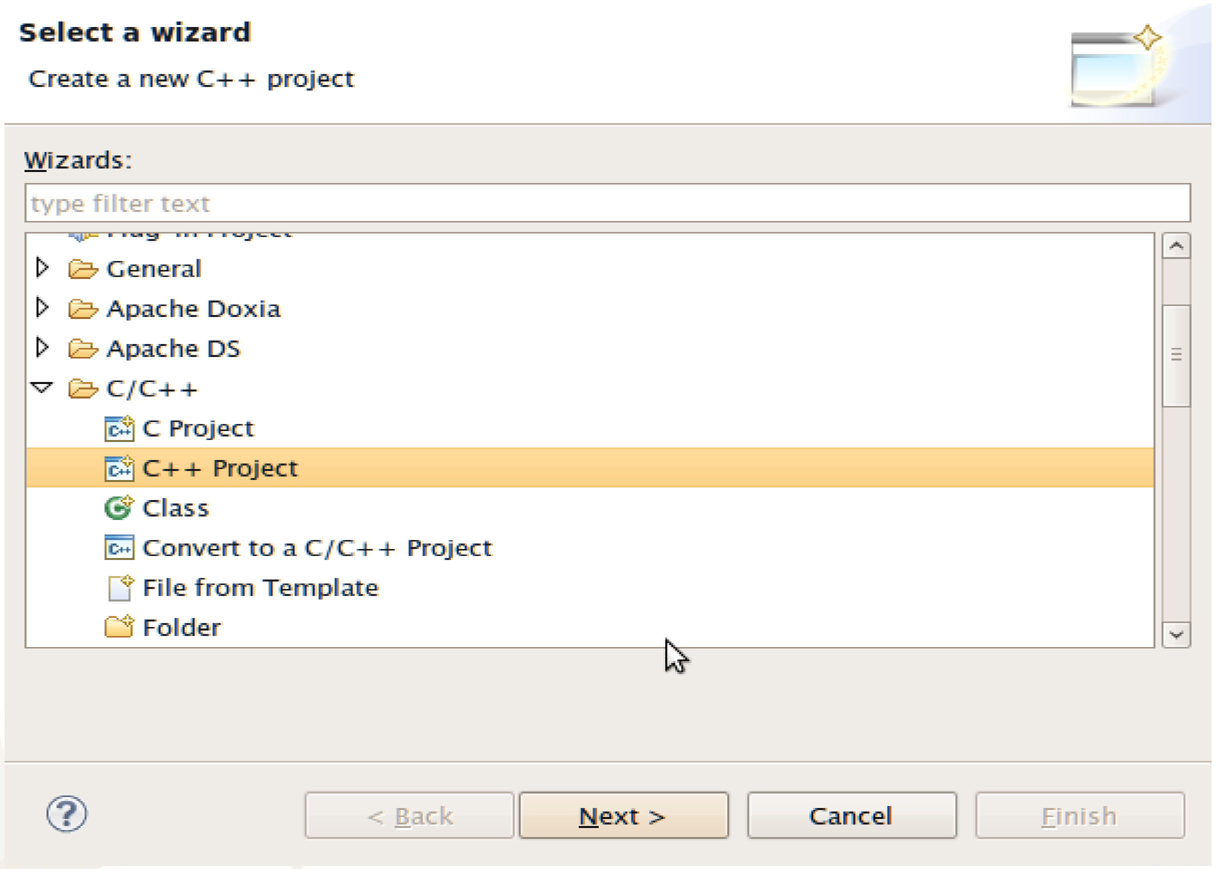
The select "Next..."
And accept the license agreement with "Finish".
Downloading and installing will take a while.
Once its done it will ask you to restart Eclipse.
This is a good idea, so select "Yes".


It is recommended you restart the workbench for the changes to take effect, but it may be possible to apply the changes to the current configuration without restarting.Would you like to restart now?

Once Eclipse has restarted you may need to configure it for your computer.

Hello, World!

Once you are in Eclipse, you are given an empty workspace. You now have to start a new project. To do so, select "File" / "New" / "Project...". Expand the section "C++" and select "C++ Project", then click "Next >".



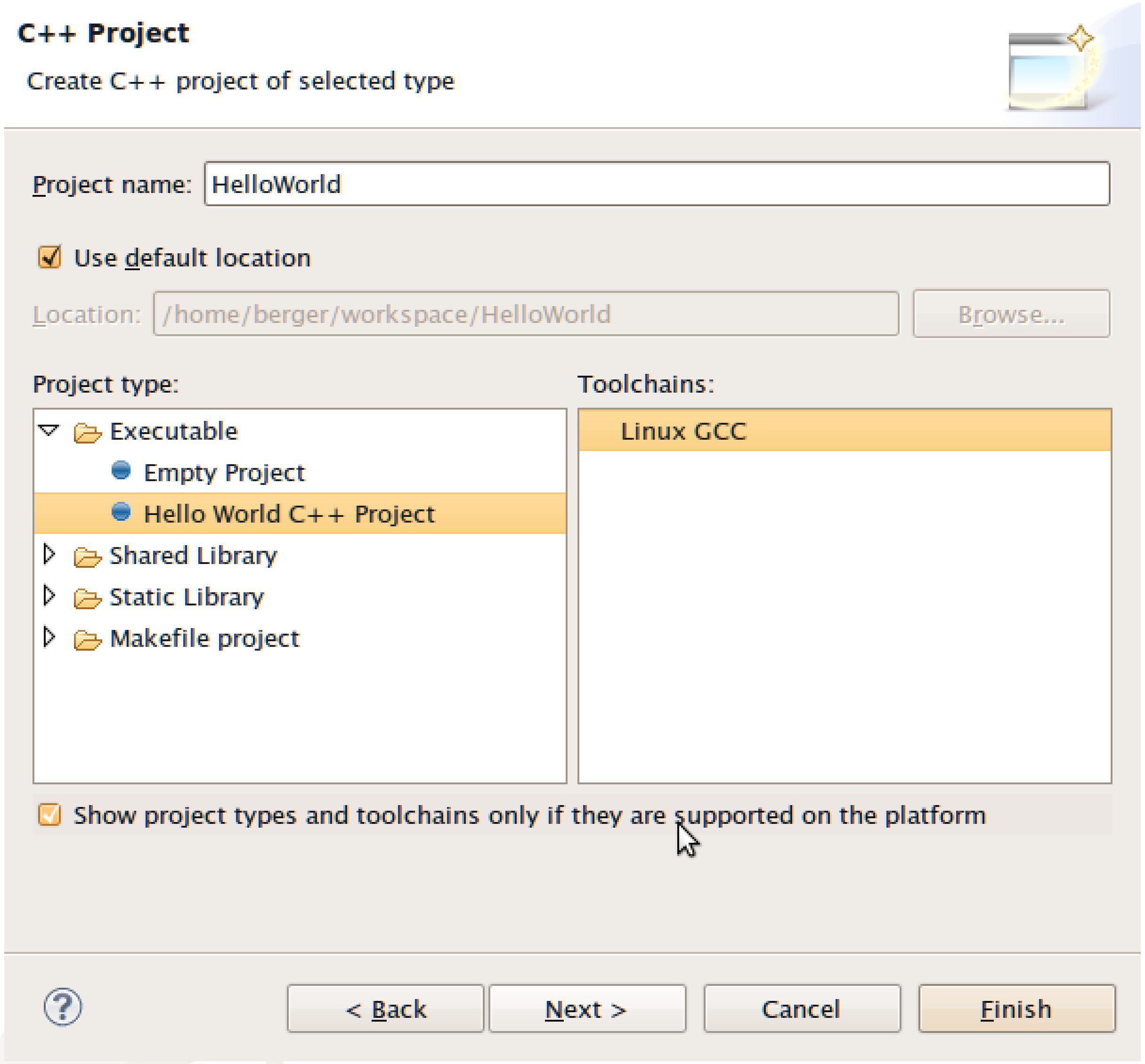
On the next screen, you have to give your project a name. In this case, it will be "HelloWorld", however, you may use any name you like. Also, you have to select a toolchain.

For Windows, select "MinGW GCC"

For Linux, select "Linux GCC"

For Mac OS X, select "MacOSX GCC“

Do not use "Cross GCC"! Unfortunately this option may be selected by default if you create an empty project!



C++ Project

Create C++ project of selected type

Project name: HelloWorld

[✓] Use default location

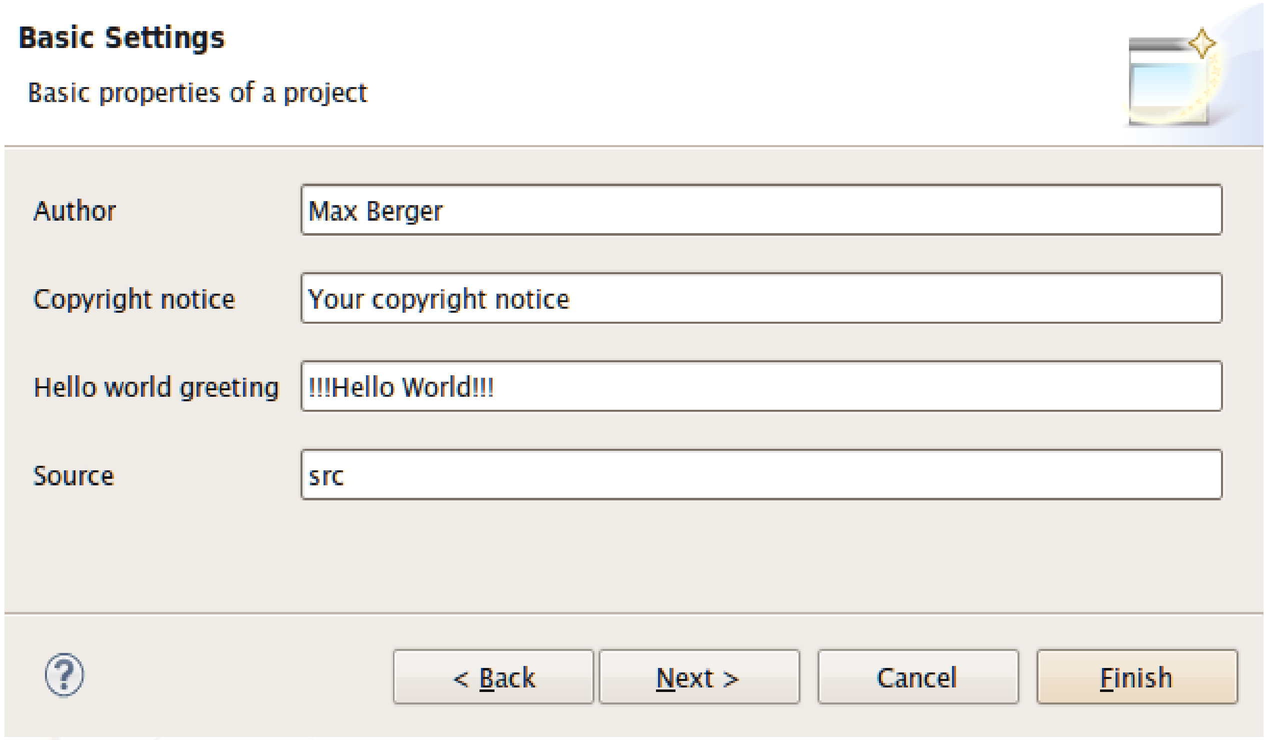
Location: /home/berger/workspace/HelloWorld

Project type:
- Executable
  - Empty Project
  - Hello World C++ Project [✓]
- Shared Library
- Static Library
- Makefile project

Toolchains:
- Linux GCC [✓]

[✓] Show project types and toolchains only if they are supported on the platform

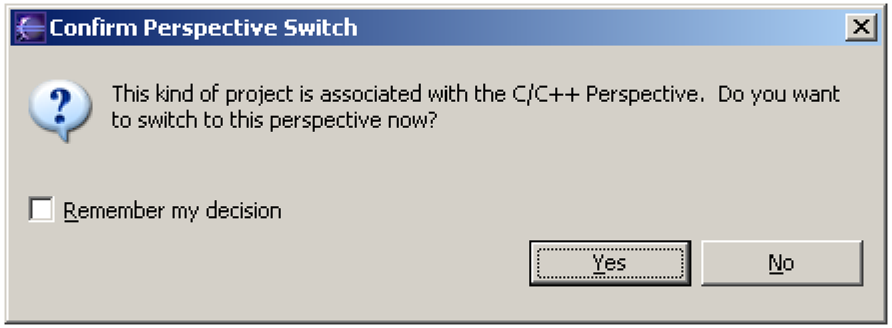
The next screen contains some Basic settings. Fill in what you like:


Basic Settings

Basic properties of a project

Author	Max Berger
Copyright notice	Your copyright notice
Hello world greeting	!!!Hello World!!!
Source	src

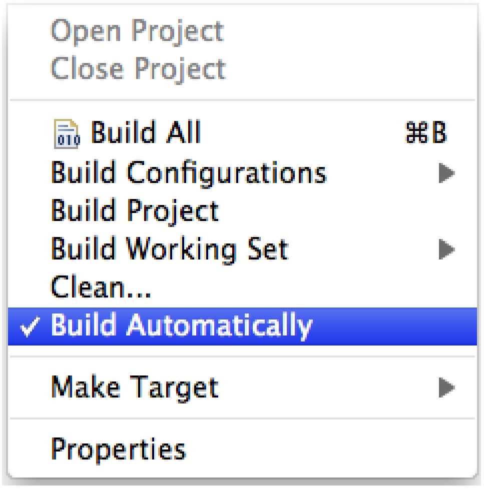
Eclipse will now generate a few things, and then ask you if you want to switch to the C/C++ Perspective. This is a good idea, so say yes.



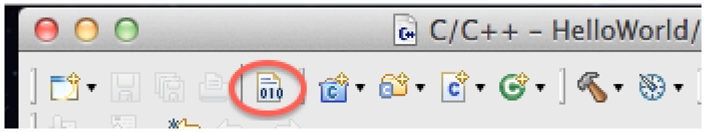
This kind of project is associated with the C/C++ Perspective. Do you want to switch to this perspective now?

Great. You have a project now, and it does contain some sample code! You will immediately get an editor window for your project. Eclipse will also auto-build your project every time you save.

If eclipse does not auto-build, you have to turn on "build automatically" in the Project Menu, or click the "Build All" button after every change.

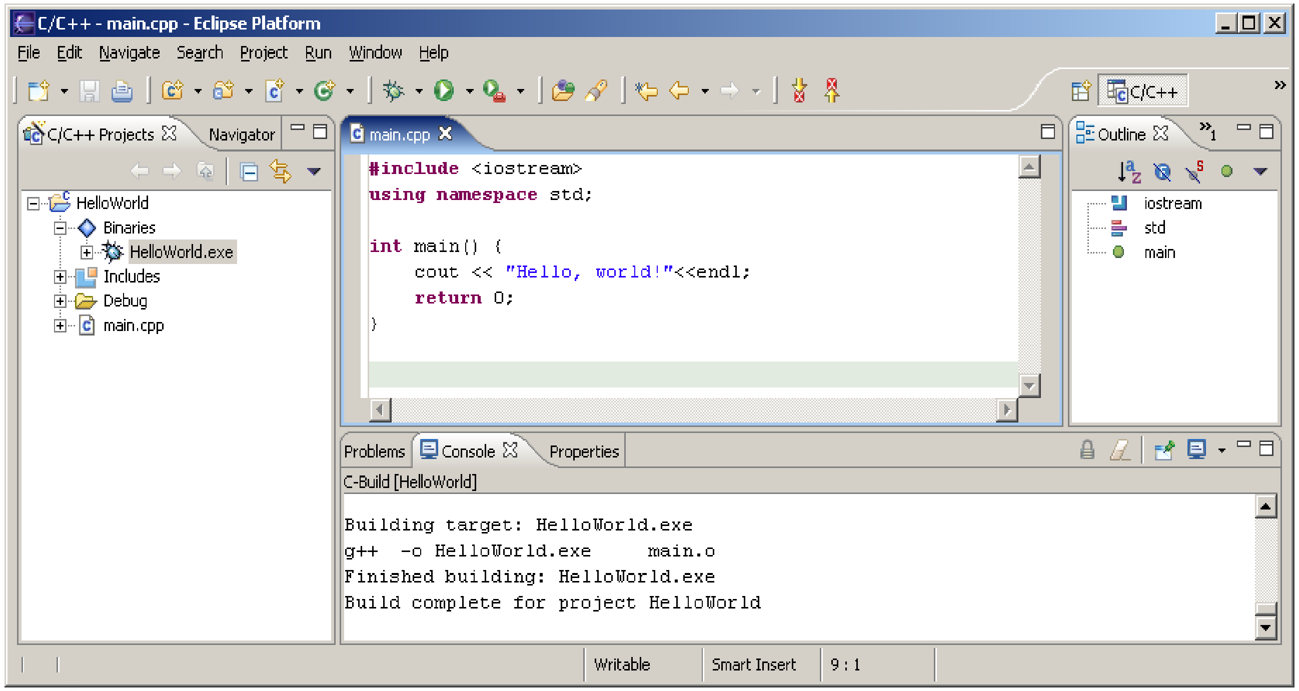


Make sure "Build Automatically" is enabled



If you turn off autobuild, you have to click the "build" button on the toolbar

Example Hello World application

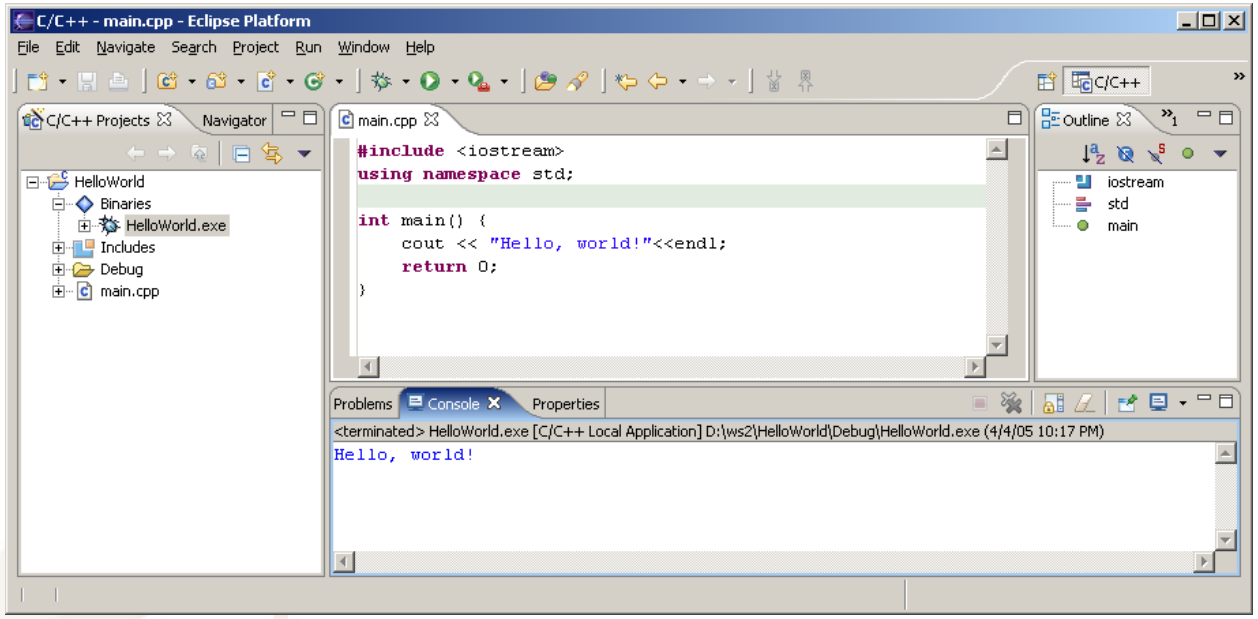


```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!"<<endl;
    return 0;
}
```

```
Building target: HelloWorld.exe
g++  -o HelloWorld.exe     main.o
Finished building: HelloWorld.exe
Build complete for project HelloWorld
```

Running the example Hello World application

Now here comes the tricky part: On the left pane, select "C/C++ Projects", expand "Binaries" and you should see and executable (HelloWorld.exe on Windows). Now right-click that executable, and select "Run" / "Run Local C/C++ Application". If everything goes well your output will be in the bottom right window in the "Console" tab and it should say "Hello, World".



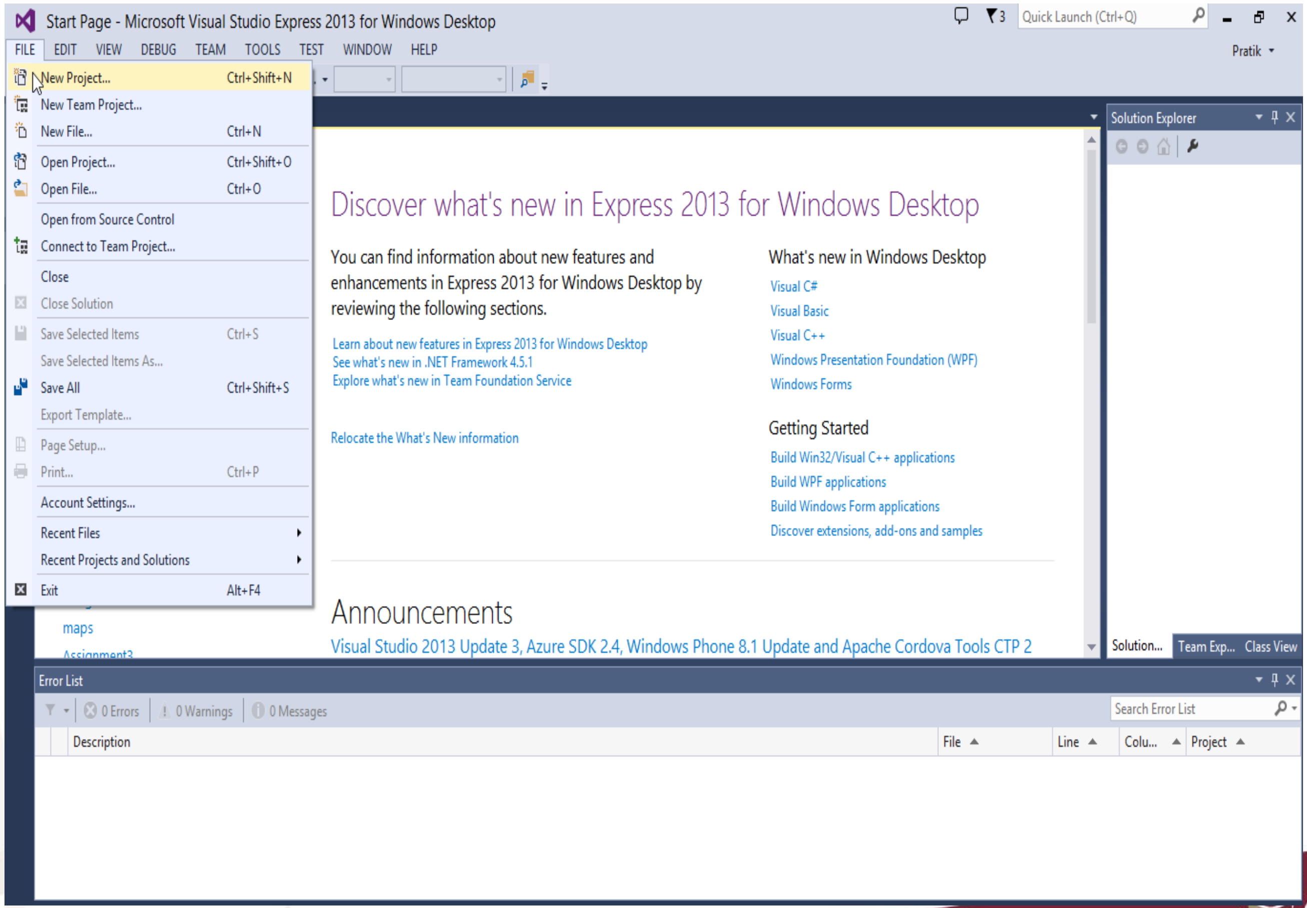
```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!"<<endl;
    return 0;
}
```

```
<terminated> HelloWorld.exe [C/C++ Local Application] D:\ws2\HelloWorld\Debug\HelloWorld.exe (4/4/05 10:17 PM)
Hello, world!
```

Visual Studio

- ✓ Open Microsoft Visual Studio.
- ✓ To create a new project, you can either click the menu item:

File >> New >> Project or press Ctrl + Shift + N.



Discover what's new in Express 2013 for Windows Desktop

You can find information about new features and enhancements in Express 2013 for Windows Desktop by reviewing the following sections.

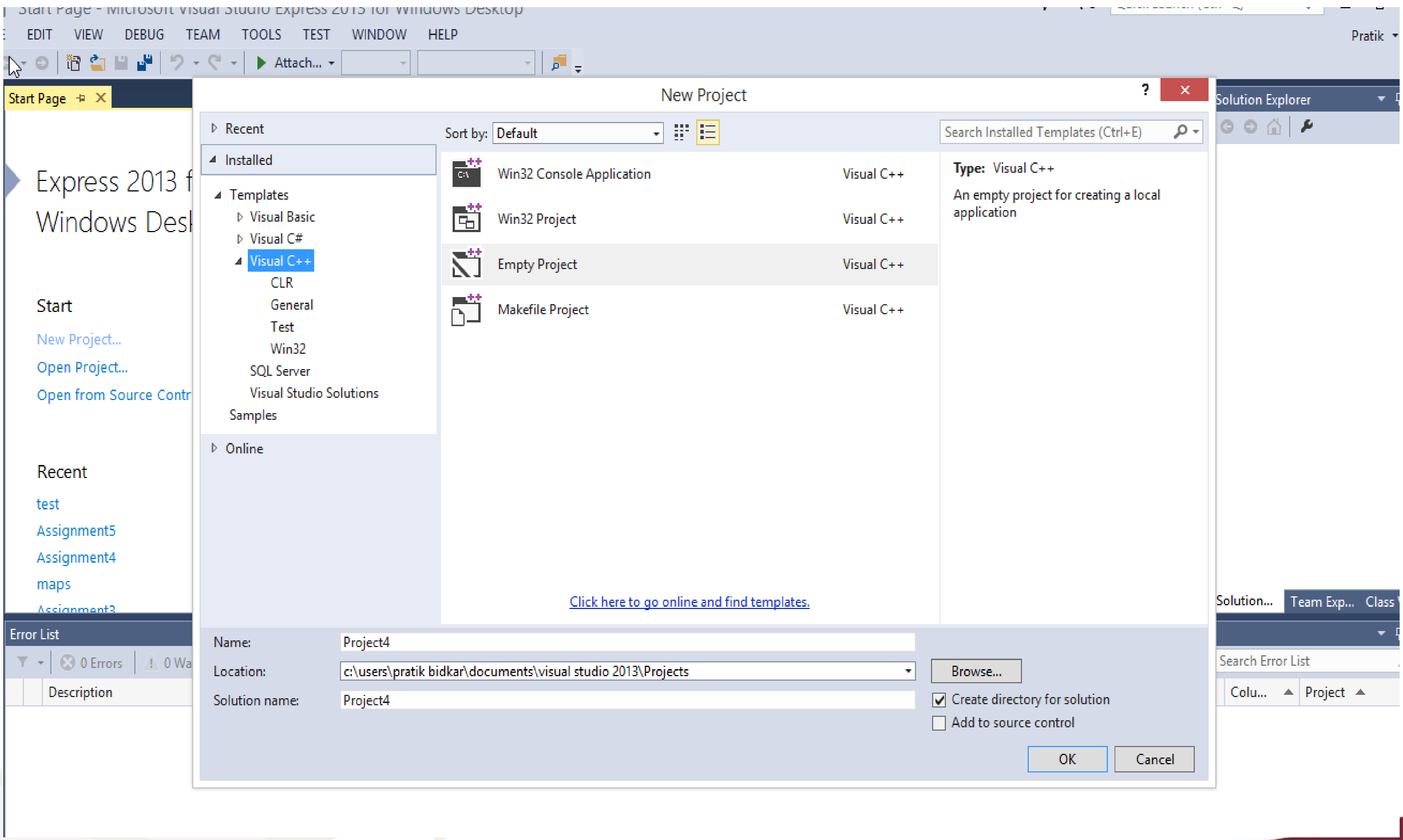
Learn about new features in Express 2013 for Windows Desktop
See what's new in .NET Framework 4.5.1
Explore what's new in Team Foundation Service

Relocate the What's New information

What's new in Windows Desktop
Visual C#
Visual Basic
Visual C++
Windows Presentation Foundation (WPF)
Windows Forms

Getting Started
Build Win32/Visual C++ applications
Build WPF applications
Build Windows Form applications
Discover extensions, add-ons and samples

Announcements
Visual Studio 2013 Update 3, Azure SDK 2.4, Windows Phone 8.1 Update and Apache Cordova Tools CTP 2



New Project

- Recent
- Installed
  - Templates
    - Visual Basic
    - Visual C#
    - Visual C++
      - CLR
      - General
      - Test
      - Win32
    - SQL Server
    - Visual Studio Solutions
  - Samples
- Online

Sort by: Default

Search Installed Templates (Ctrl+E)

Win32 Console Application	Visual C++
Win32 Project	Visual C++
Empty Project	Visual C++
Makefile Project	Visual C++

Type: Visual C++

An empty project for creating a local application

Click here to go online and find templates.

Name: Project4

Location: c:\users\pratik bidkar\documents\visual studio 2013\Projects

Solution name: Project4

Browse...

[✓] Create directory for solution

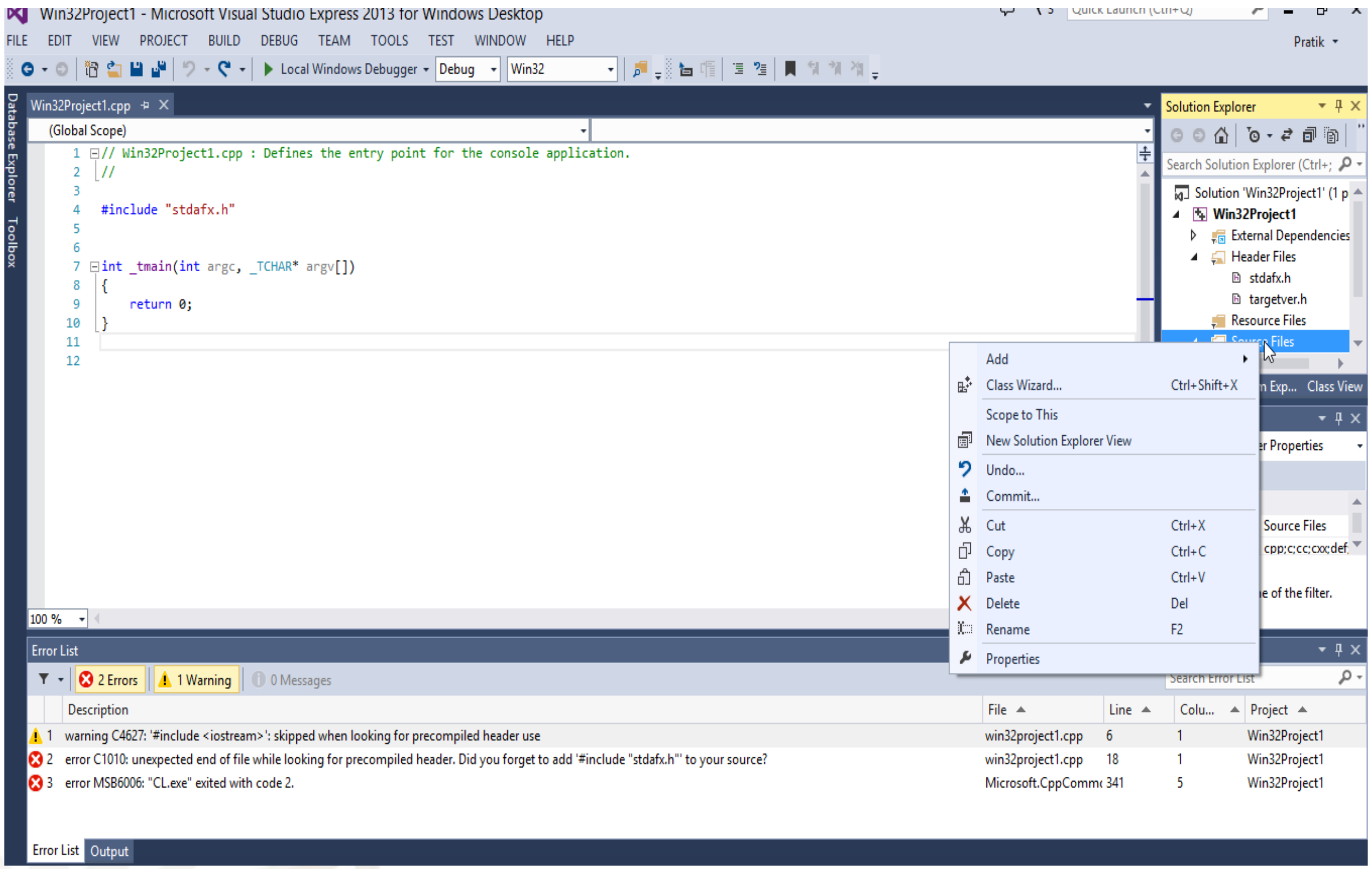
[ ] Add to source control

OK Cancel

Université Concordia University

Typing Program Code

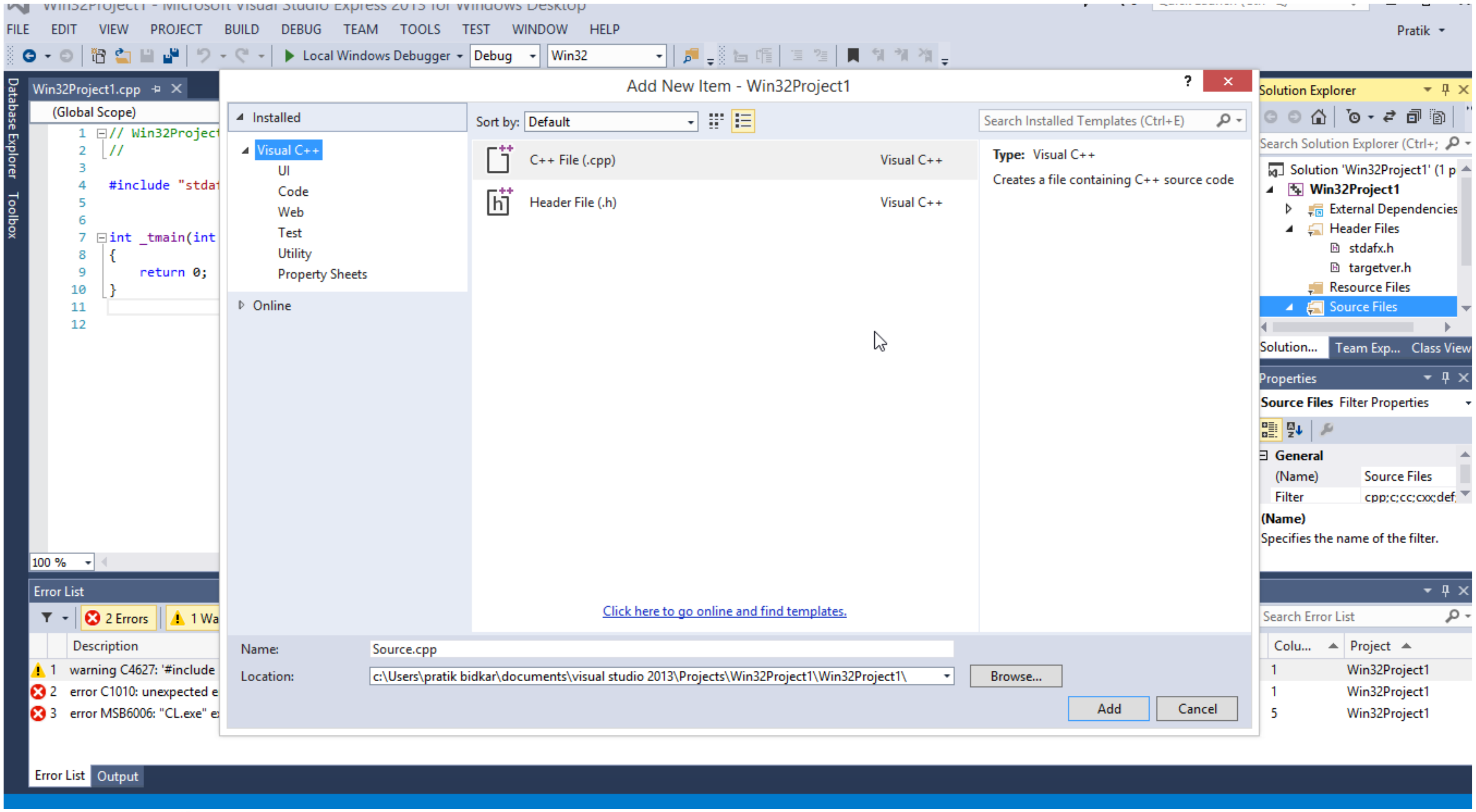
Right Click the Source Files option from Right pane (Solution Explorer) >> *Select option Add >> Source Files >> New Item*



```
// Win32Project1.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"


int _tmain(int argc, _TCHAR* argv[])
{
    return 0;
}
```

	Description	File	Line	Colu...	Project
1	warning C4627: '#include <iostream>': skipped when looking for precompiled header use	win32project1.cpp	6	1	Win32Project1
2	error C1010: unexpected end of file while looking for precompiled header. Did you forget to add '#include "stdafx.h"' to your source?	win32project1.cpp	18	1	Win32Project1
3	error MSB6006: "CL.exe" exited with code 2.	Microsoft.CppComm	341	5	Win32Project1

Once you have used either one of the options mentioned above, the *Add New Item* dialog box will appear. In the right pane, select *C++ File (.cpp).*


Win32Project1 - Microsoft Visual Studio Express 2013 for Windows Desktop
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WINDOW HELP
Pratik
Local Windows Debugger
Debug
Win32
Win32Project1.cpp
(Global Scope)
Add New Item - Win32Project1
Installed
Visual C++
UI
Code
Web
Test
Utility
Property Sheets
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
C++ File (.cpp) Visual C++
Header File (.h) Visual C++
Click here to go online and find templates.
Browse...
Add
Cancel
Solution Explorer
Search Solution Explorer (Ctrl+;
Solution 'Win32Project1' (1 p
Win32Project1
External Dependencies
Header Files
stdafx.h
targetver.h
Resource Files
Source Files
Solution... Team Exp... Class View
Properties
Source Files Filter Properties
General
Error List
2 Errors
1 Wa
Search Error List
Error List Output
Database Explorer Toolbox
100 %

```
// Win32Projec
//

#include "stdaf

int _tmain(int
{
    return 0;
}
```

Type: Visual C++

Creates a file containing C++ source code

Name: Source.cpp

Location: c:\Users\pratik bidkar\documents\visual studio 2013\Projects\Win32Project1\Win32Project1\

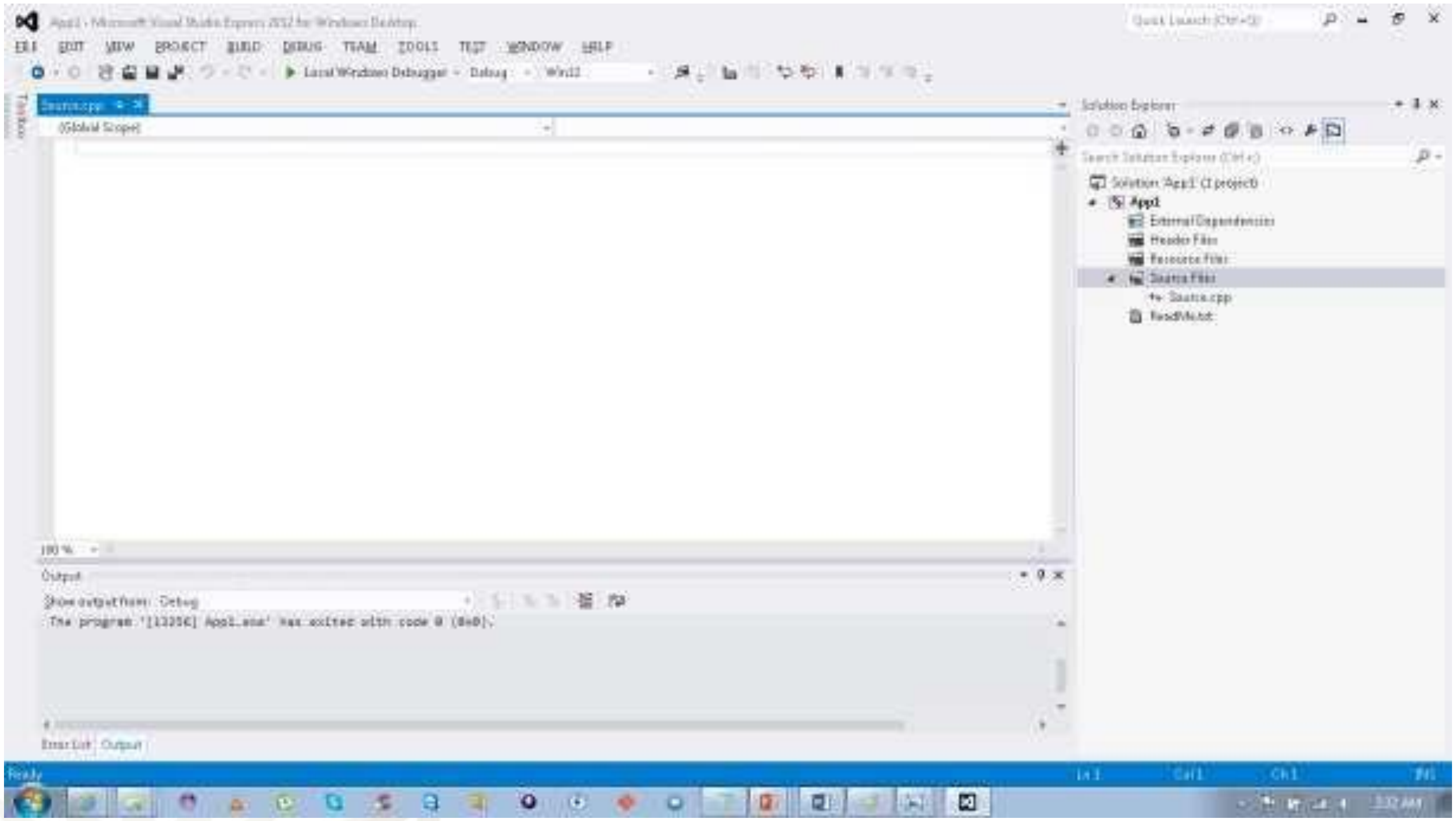
(Name)	Source Files
Filter	cpp;c;cc;cxx;def;

(Name)

Specifies the name of the filter.

	Description	Colu...	Project
1	warning C4627: '#include	1	Win32Project1
2	error C1010: unexpected e	1	Win32Project1
3	error MSB6006: "CL.exe" e	5	Win32Project1

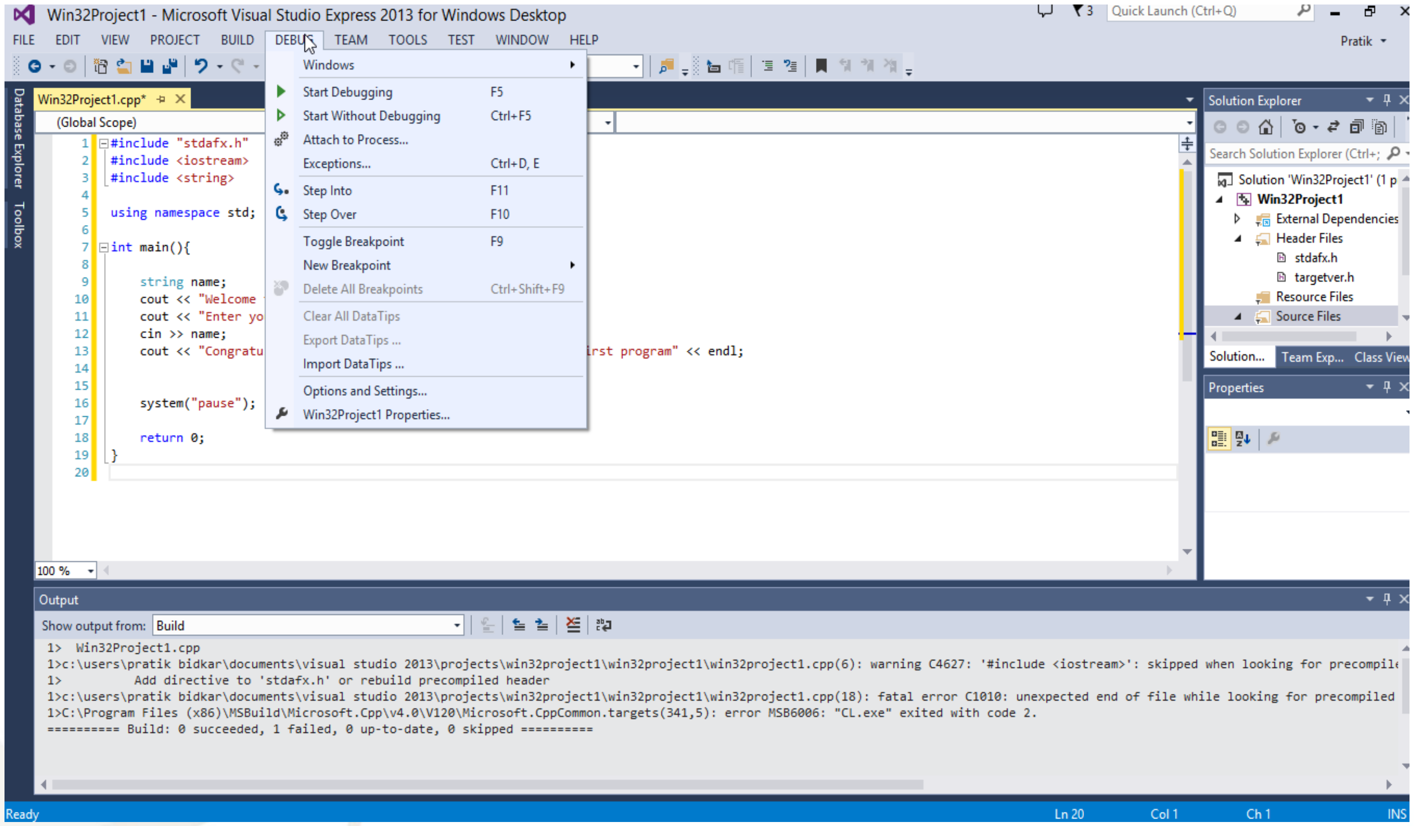
Click the *Add* button to enter the edit mode.



```
The program '[13356] App1.exe' has exited with code 0 (0x0).
```

Executing Your Program

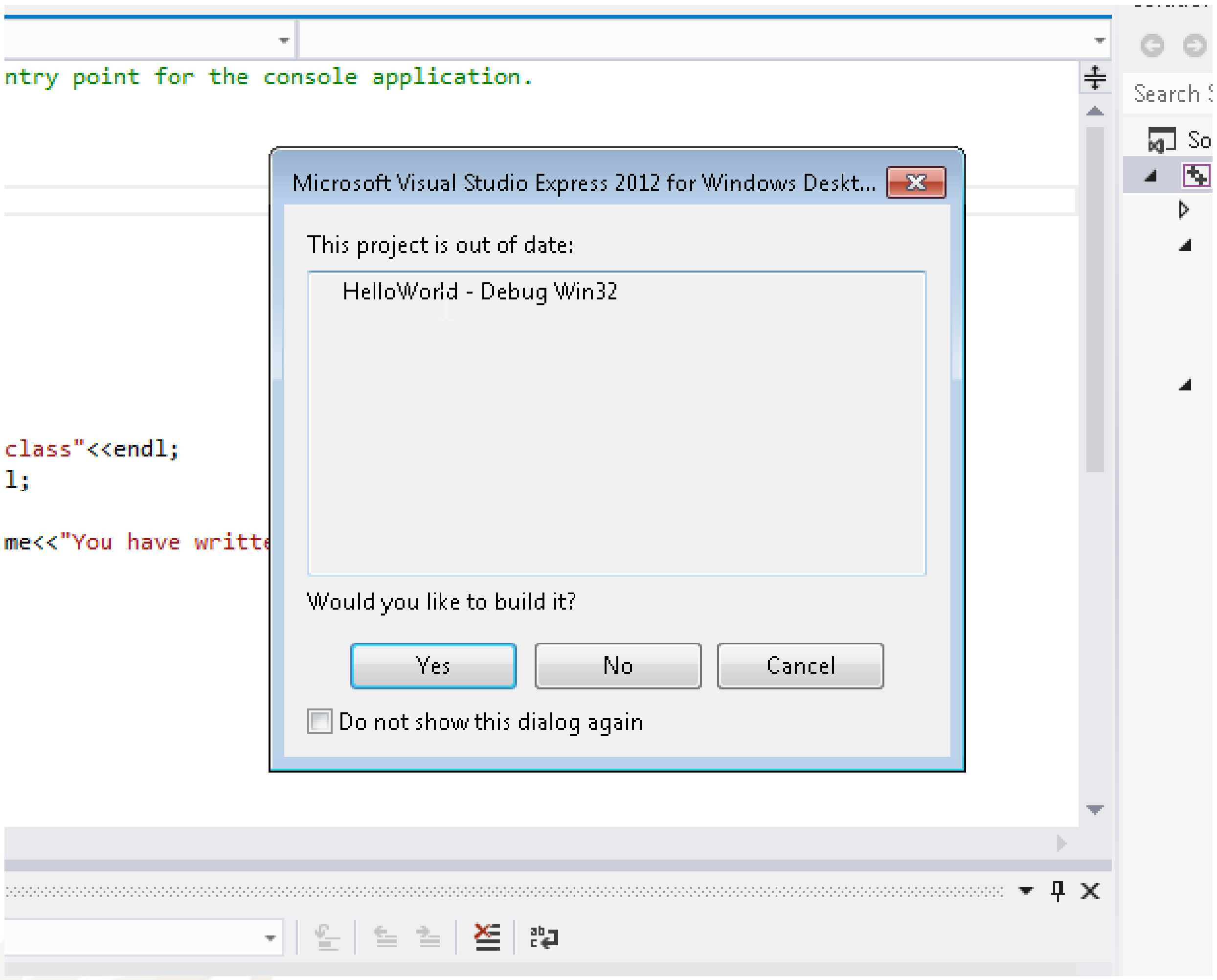
Once your program has been typed, the next step is to see if it works.
To execute your program, click the menu item
Debug >> Start Without Debugging.



```
#include "stdafx.h"
#include <iostream>
#include <string>

using namespace std;

int main(){

    string name;
    cout << "Welcome
    cout << "Enter yo
    cin >> name;
    cout << "Congratu                                irst program" << endl;


    system("pause");

    return 0;
}

```

```
1>  Win32Project1.cpp
1>c:\users\pratik bidkar\documents\visual studio 2013\projects\win32project1\win32project1\win32project1.cpp(6): warning C4627: '#include <iostream>': skipped when looking for precompile
1>          Add directive to 'stdafx.h' or rebuild precompiled header
1>c:\users\pratik bidkar\documents\visual studio 2013\projects\win32project1\win32project1\win32project1.cpp(18): fatal error C1010: unexpected end of file while looking for precompiled
1>C:\Program Files (x86)\MSBuild\Microsoft.Cpp\v4.0\V120\Microsoft.CppCommon.targets(341,5): error MSB6006: "CL.exe" exited with code 2.
========== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped ==========
```

The following message box will appear:



```
ntry point for the console application.

class"<<endl;
l;

me<<"You have writte
```

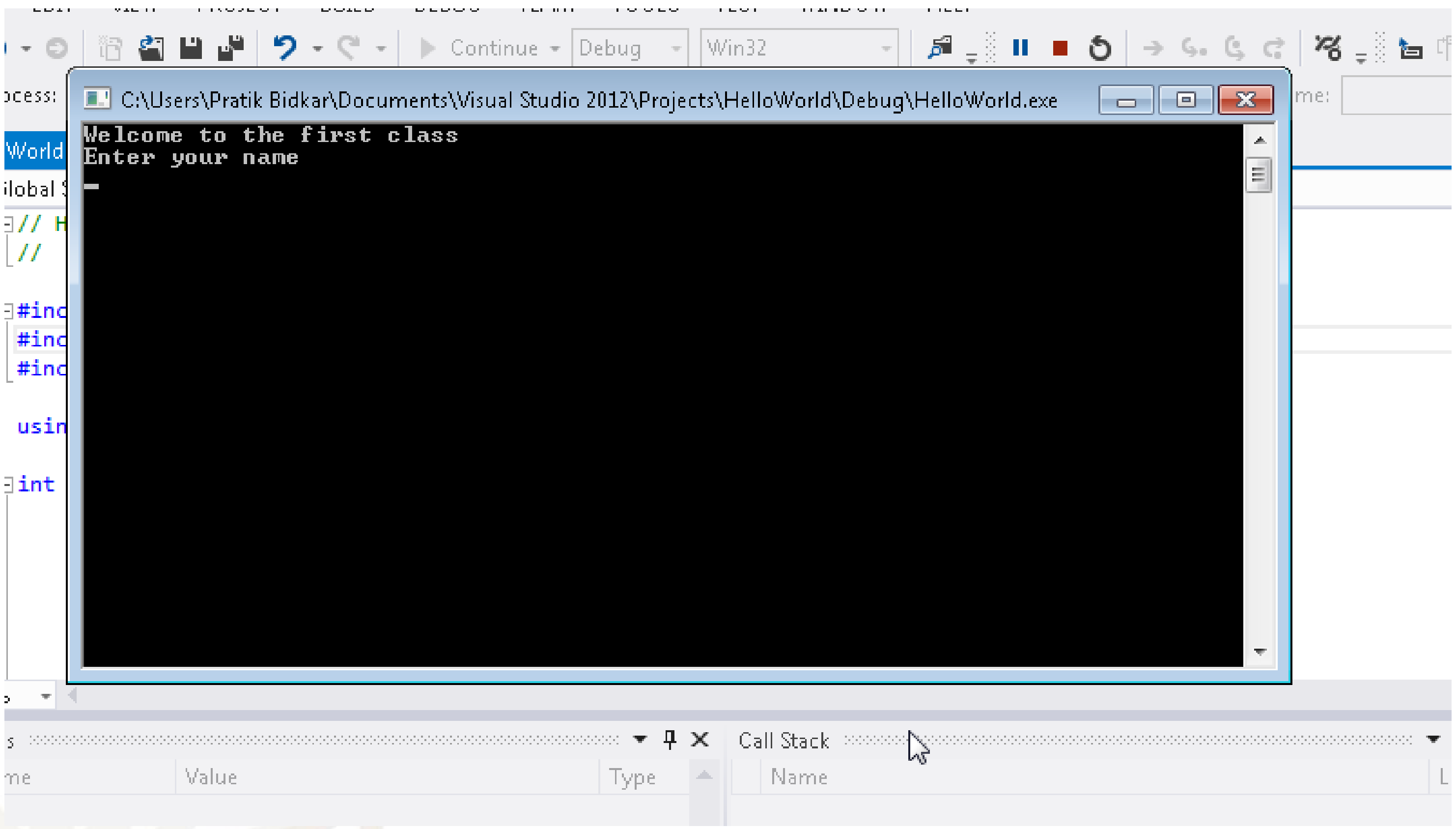
Microsoft Visual Studio Express 2012 for Windows Deskt...

This project is out of date:

HelloWorld - Debug Win32

Would you like to build it?

If your program has no code errors, you will see the following screen:


Continue
Debug
Win32
C:\Users\Pratik Bidkar\Documents\Visual Studio 2012\Projects\HelloWorld\Debug\HelloWorld.exe
Call Stack
Value
Type
Name

```
Welcome to the first class
Enter your name
_
```

Example:

```
#include <iostream> // allows program to output data to the screen
using namespace std;
int main()
{
cout << “Hello World!”;
cout << endl;
return 0;
}
```

//: is a single-line comment

#include <iostream>: is a preprocessor directive.
This line notifies the preprocessor to include in the program the contents of the input/output stream header <iostream>.
(*Lines that begin with # are processed by the preprocessor before the program is compiled*.)

using namespace std; shows that we are using a name, such as cin and cout, that belongs to “namespace” std

int main() is a part of every C++ program. C++ programs begin executing at function main.

return 0; indicate that program ended successfully

cin, cout and cerr

- Certain C++ functions take their input from cin (the standard input stream; pronounced “see-in”), which is normally the keyboard, but cin can be redirected to another device.
- Data is often output to cout (the standard output stream; pronounced “see-out”), which is normally the computer screen, but cout can be redirected to another device.
- Data may be output to other devices, such as disks and hardcopy printers.
- There is also a standard error stream referred to as cerr. The cerr stream is used for displaying error messages.

Example:

Define a class with the name ‘GradeBook’ with a member function ‘displayMessage’, and creating a GradeBook object, then call its displayMessage function.

Define two files :

Gradebook.h to place the class

main.cpp which uses class GradeBook

GradeBook.h

```
// GradeBook class definition in a separate file from main
#include <iostream>
#include <string>                  //class GradeBook uses C++ standard string class
using namespace std;
// GradeBook class definition
class GradeBook
{
public:
// constructor initializes courseName with string supplied as argument
GradeBook( string name )
{
setCourseName( name );        // call set function to initializes courseName
}
//function to set the courseName
void setCourseName( string name)
{
courseName=name;               // store the course name in the object
}                    // end function setCourseName
//function to get the courseName
 string getCourseName()
 {
return courseName;             //return object's courseName
 }                   // end function getCourseName
 // display a welcome message to the GradeBook user
void displayMessage()
{
// call getCourseName to get the courseName
cout << "Welcome to the grade book for\n"<<getCourseName()<< "!"<<endl;
}                    //end function displayMessage
private:
string courseName;             // courseName for this GradeBook
}; // end class GradeBook
```

main.cpp

```
#include <iostream>
#include "GradeBook.h"      //include definition of class
GradeBook
using namespace std;
//function main begins program execution
 int main()
 {
 //create two GradeBook objects
 GradeBook Gradebook1("COMP 352");
 GradeBook Gradebook2("COMP 345");
 //display initial value of courseName for each GradBook
 cout << "Gradebook1 created for course: " <<
Gradebook1.getCourseName()
  << "\nGradebook1 created for course: " <<
Gradebook2.getCourseName()
      <<endl;
 }// end main
```

Good Luck

and

Enjoy your programming