**Tutorial 2**

[Eclipse](http://www.eclipse.org/) is a special software system, called an Integrated Development Environment (IDE), which is used to write programs in a variety of programming languages. It allows you to perform several distinct program development steps in one place.

This tutorial is intended to help you become familiar with the Eclipse IDE. With the help of the Teaching Assistant (TA) as well as the appendix provided at the end of this tutorial, you are asked to do the following:

1. Launch Eclipse and configure the Eclipse IDE;
2. Create a new C++ project with a name of your choice;
3. Create a source file named “MyFirstProg.cpp” and enter the following source code into the MyFirstProg.cpp window:

// Text-printing program.

#include <iostream>

// function main begins program execution

int main()

{

std::cout << "Welcome to C++!\n"; // display message

return 0; // indicate that program ended successfully

} // end function main

As you edit the file, you may notice **red underlines** appearing. This is Eclipse's attempt to correct you, the same way a word processor attempts to correct spelling. If you type the code correctly, the red lines will eventually disappear, though it is normal for some to appear as you are writing programs. While you are editing the source code, it is a good practice to save your work frequently. The more often you save your work, the less work you will have to recreate in the event of a system crash or other interruption in your work.

1. Run your program and click on the tab labeled "Console" to see the output.
2. Modify the previous code using the following source code and repeat step 4:

// Printing multiple lines of text with a single statement.

#include <iostream>

// function main begins program execution

int main()

{

std::cout << "Welcome\nto\n\nC++!\n";

return 0; // indicate that program ended successfully

} // end function main

It is important to note the difference in terms of output between this code and the previous code in 3).

1. Modify the previous code using the following source code and repeat steps 4:

// Addition program.

#include <iostream>

// function main begins program execution

int main()

{

int number1; // first integer to be input by user

int number2; // second integer to be input by user

int sum; // variable in which sum will be stored

std::cout << "Enter first integer: "; // prompt

std::cin >> number1; // read an integer

std::cout << "Enter second integer: "; // prompt

std::cin >> number2; // read an integer

sum = number1 + number2; // assign result to sum

std::cout << "Sum is " << sum << “\n”; // print sum

return 0; // indicate that program ended successfully

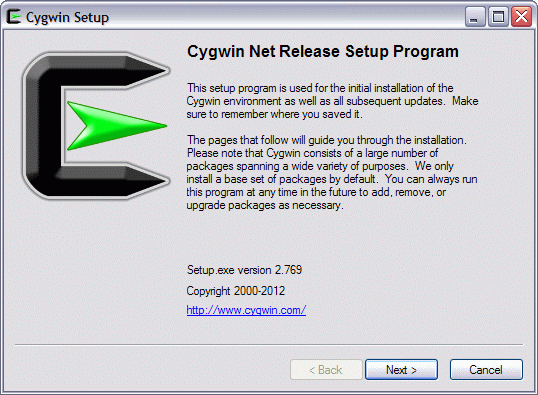
} // end function main

**APPENDIX**

1. **Eclipse CDT**

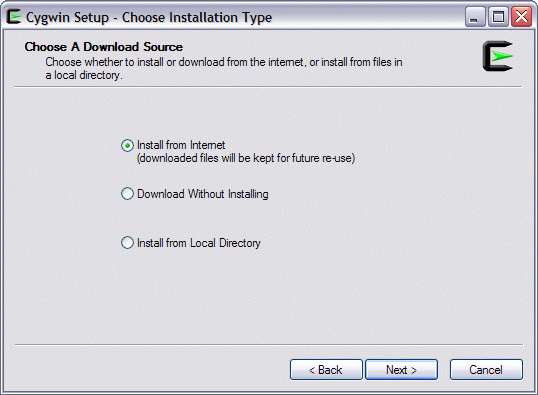
**Before installing Eclipse, a C++ compiler must be installed. There are two choices: MinGW GCC or Cygwin with GCC. Note that MinGW is lighter and easier to install, but having less features than Cygwin.**

1. **Cygwin with GCC:**(Credit: http://web.cecs.pdx.edu)
2. Navigate to Website:  [http://www.cygwin.com](http://www.cygwin.com/)
3. Download [setup-x86.exe (32-bit installation)](http://cygwin.com/setup-x86.exe) or [setup-x86\_64.exe (64-bit installation)](http://cygwin.com/setup-x86_64.exe) to a convenient directory on your local drive.
4. Run the setup-x86.exe or setu\_x86\_64.exe file to begin installation.  
   (Note: You may need local administrator rights.)
5. The initial Cygwin Setup window appears.



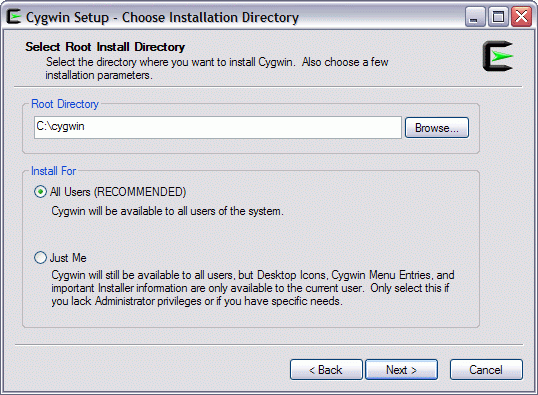
* + Click [Next >].

1. The Choose Installation Type window appears.



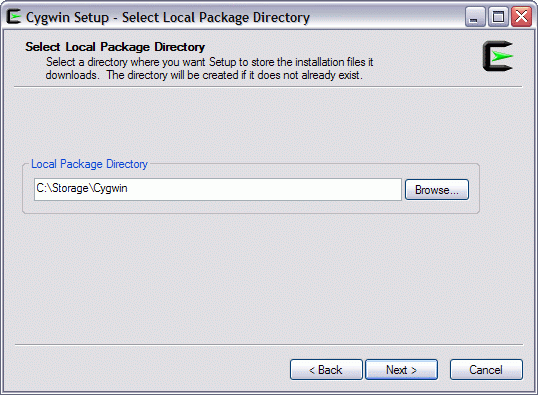
* + Select the "Install from Internet" option.
  + Click [Next >].

1. The Choose Installation Type window appears.



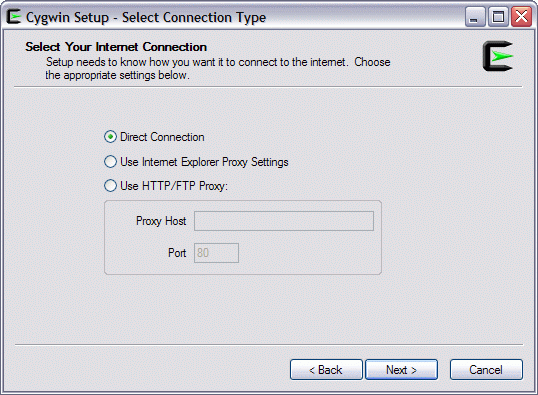
* + In the Root Directory input box, type in the directory path where you want Cygwin installed.
  + In the Install For selection box, choose the desired type (All Users or Just me).
  + Click [Next >].

1. The Select Local Package Directory window appears.



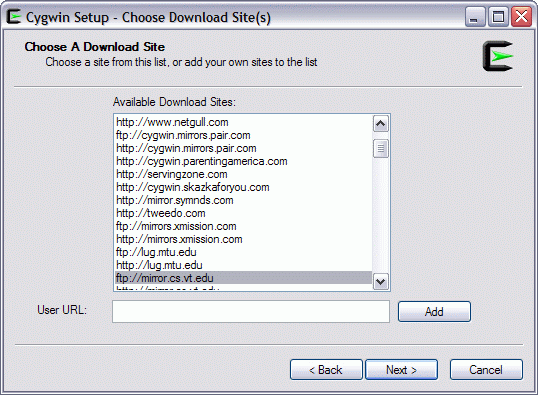
* + In the Local Package Directory input box, type in the directory path where you want to store the downloaded installation packages.  
    Note:  This is not the same as the Cygwin root directory that you have selected in the previous step.
  + Click [Next >].

1. The Select Connection Type window appears.



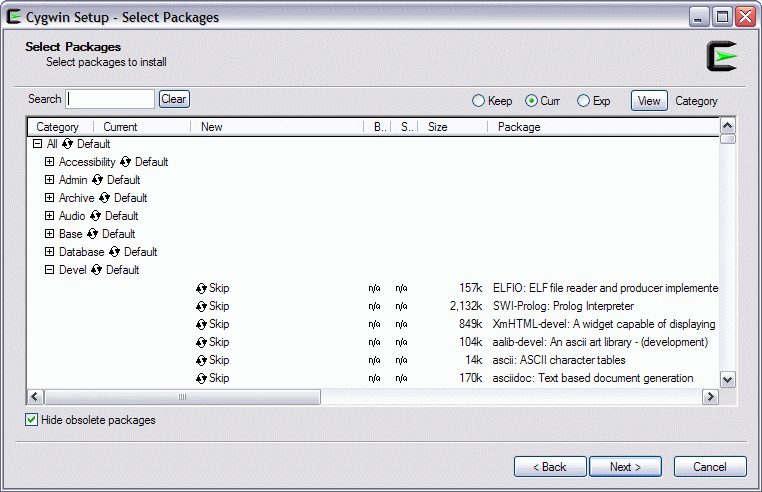
* + In most cases, the default Direct Connection option is most appropriate.
  + Click [Next >].

1. The Choose Download Site(s) window appears.



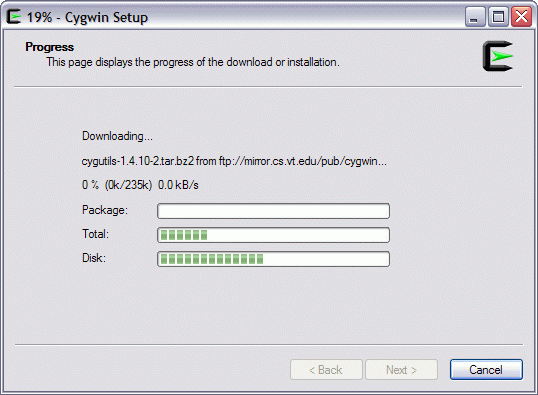
* + From the list of Available Download Sites: select a site from which to download the Cygwin packages (try a U.S. or nearby location).
  + Click [Next >].

1. After a short time, the Select Packages window appears.

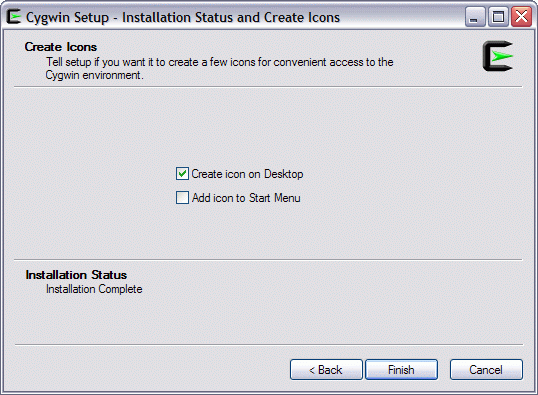


* + From this window, choose the Cygwin applications to install. For our purposes, you will select GNU C/C++ packages.
    - Click the + sign next to the Devel category to expand it.
      * You will see a long list of possible packages that can be downloaded. Scroll the list to see more packages.
      * Pick each of the following packages by clicking its corresponding "Skip" marker.
        + **gcc-core**: C compiler subpackage
        + **gcc-g++**: C++ subpackage
        + **libgcc1**: C runtime library
        + **gdb:** The GNU Debugger
        + **make**: The GNU version of the 'make' utility
      * Note: If a package has not been previously installed, then clicking the "Skip" marker selects the package and shows its version number.
      * Click the – sign next to the Devel category to close it.
    - Click the + sign next to the Libs category to expand it.
      * Pick the following package by clicking its corresponding "Skip" marker.
        + **libmpfr4:** A library for multiple-precision floating-point arithmetic with exact rounding  
          (Note: On some systems, this may already be selected. If so, you may ignore this step.)
      * Click the – sign next to the Libs category to close it.
    - In addition to the specific packages you picked, the setup program automatically selects other packages that are needed to build a working Cygwin environment.
  + Click [Next >].

1. If a "Resolving Dependencies" window appears, just click [Next >].
2. The download progress window appears. It will take a while for the downloads and installation to finish.



1. After package installation is done, the Installation Status and Create Icons window appears.



* + Make your icon selections.
  + Click [Finish].

1. **Getting MinGW:**

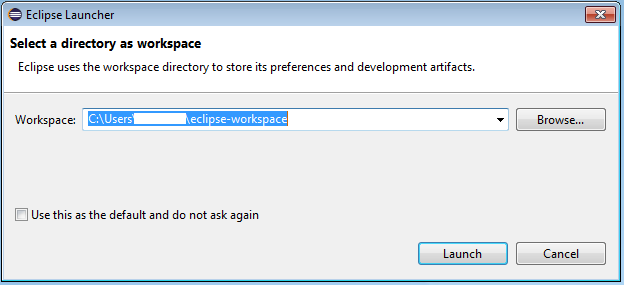
For Mac users, you may skip installing any C++ compiler as many Mac machines come with a C++ compiler (MAC OSX GCC). So, before following the steps below, make sure that the compiler has not been installed.

To install MinGW in Windows:

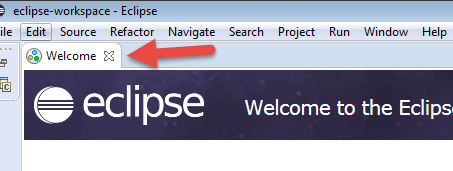
1. Go to MinGW site at http://www.mingw.org. Choose downloads, then Installer, then mingw-get-setup.exe.
2. Run the download installer. Set the installation directory. You can use the default directory of C:\MINGW. Don’t use either desktop or Program Files (or any folder with a space in the name).
3. In MinGW Installation Manager, make sure Basic Setup is chosen. Then, Select Installation->Apply Changes. This step might take a while.
4. Set your environment variable PATH to include"<MINGW\_HOME>/bin" where <MINGW\_HOME> is the MinGW installed directory that you have chosen in the previous step. (probably C:\MinGW). You may have to google how to set environment variables on your computer. On windows 8, go to control panel->system->advanced system settings->advanced->environment variables. Under System Variable, Choose the Path variable, and then choose “edit”. Add to the end, C:\MinGW\bin (no slash or “;” at the end unless you are adding something else to the path as well). Then hit a lot of ok’s.
5. **Install Eclipse (Windows and Mac)**
6. Download and run [JDK](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html) installer and follow the instructions.
7. Download Eclipse corresponding to your machine and operating system from <http://www.eclipse.org/downloads/>
8. Follow the instructions.
9. **Using Eclipse**
10. **Launch Eclipse**
    1. Start Eclipse by running "eclipse.exe" in the Eclipse installed directory or click on the icon on the Desktop.



* 1. Choose an appropriate directory for your ***workspace*** (i.e., where you would like to save your work).



* 1. If the "welcome" screen shows up, close it by clicking the "close" button.

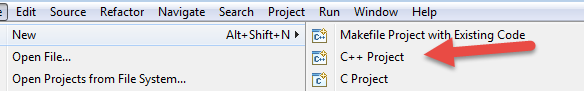


1. **Create a new C++ Project**

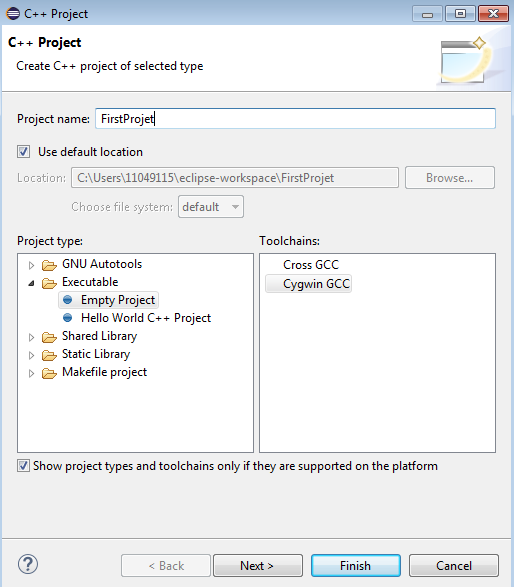
For each C++ application, you need to create a project to keep all the source codes, object files, executable files, and relevant resources.

To create a new C++ project:

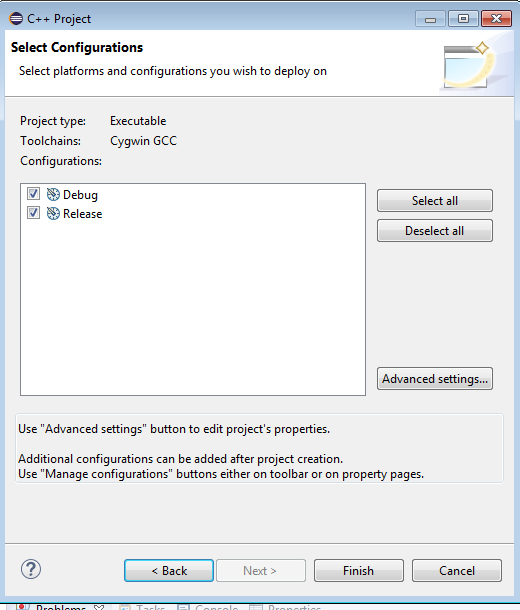
* 1. Choose "File" menu ⇒ "New" ⇒ Project... ⇒ C++ project.



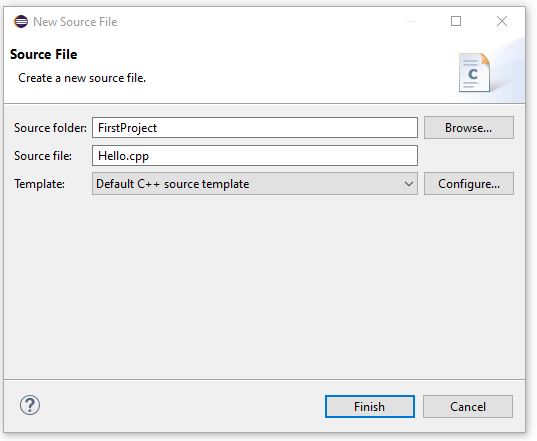
* 1. The "C++ Project" dialog pops up.
     1. In "Project name" field, enter "FirstProject";
     2. In "Project Types" box, select "Executable" ⇒ "Empty Project";
     3. In "Toolchains" box, choose your compiler, e.g., "Cygwin GCC" ⇒ Next.



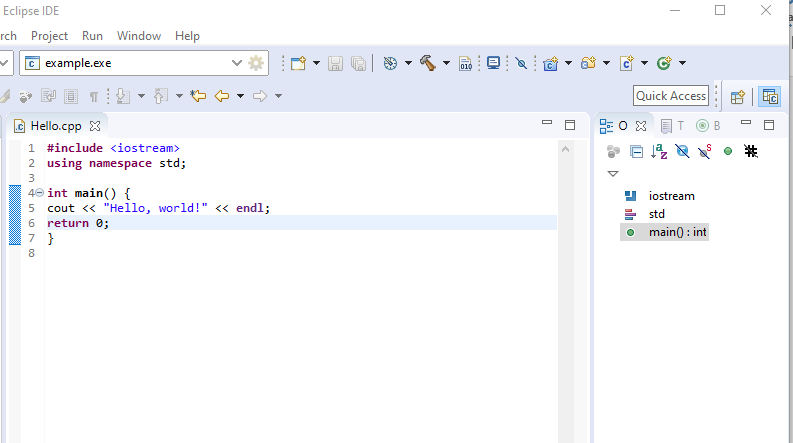
* 1. The "Select Configurations" dialog appears. Select both "Debug" and "Release" ⇒ Finish.



1. **Write a Hello-world C++ Program**
2. In the "Project Explorer" (leftmost panel) ⇒ Right-click on "FirstProject" (or use the "File" menu) ⇒ New ⇒ Source File.
3. The "New Source File" dialog pops up.
   1. In "Source file" field, enter "Hello.cpp".
   2. Click "Finish".



1. The source file "Hello.cpp" opens on the editor panel (double-click on "test.cpp" to open if necessary). Enter the following codes:

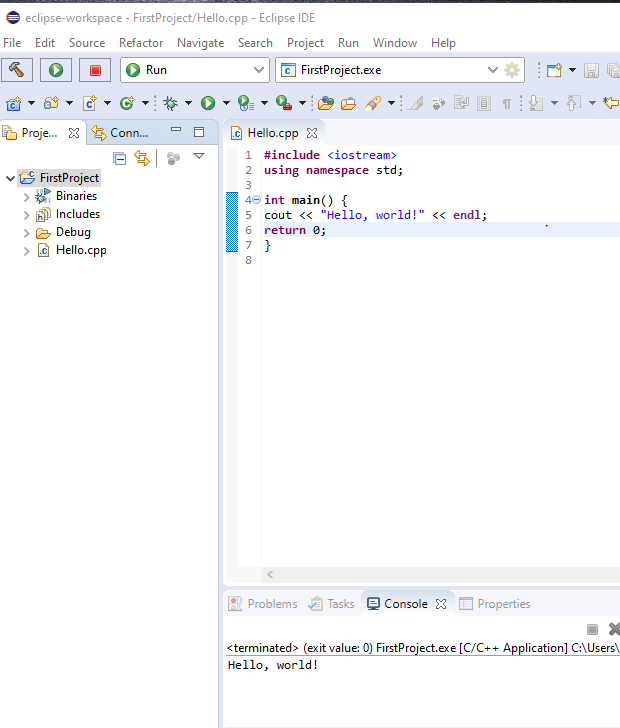


1. **Compile/Build**

Right-click on the "FirstProject" (or use the "Project" menu) ⇒ choose "Build Project" to compile and link the program.

1. **Run**

To run the program, right-click on the "FirstProject" (or anywhere on the source "Hello.cpp", or select the "Run" menu) ⇒ Run As ⇒ Local C/C++ Application ⇒ (If ask, choose Cygwin's gdb debugger) ⇒ The output "Hello, world!" appears on the "Console" panel.



**NOTE**: You need to create a *new* C++ project for EACH of your programming problems. This is messy for writing toy programs!