

# Laboratorio di Tecnologie dell'Informazione

Ing. Marco Bertini marco.bertini@unifi.it http://www.micc.unifi.it/bertini/

martedì 4 marzo 14



# Building a "Hello world" with Eclipse

"When debugging, novices insert corrective code;

experts remove defective code." - Richard Pattis



### Use the project wizard

- <u>File</u> > <u>New</u> > <u>C++ Project</u>
- Select the "Executable" type: Eclipse will manage automatically the Makefile
  - The Makefile project instead requires that the user manages the Makefile (though Eclipse can create a sample one)

	Eclipse	File	Edit	Refactor	Navigate	Se	arch	Project	Run	Window	Help
0	0	Nev			₹ ₩N	×	C.Ŷ	C Project			
] 📬 🛛 🔚 📄		Open File					🖻 C++ Project				
] 🏇 -	Or Q ]	Clo			æ		Ú	Project			
쀁 Project Explo 🛙		Close All		企業W		Convert to a C/C++ Make Project					
			Save		H	S		Source Fo	-		,,

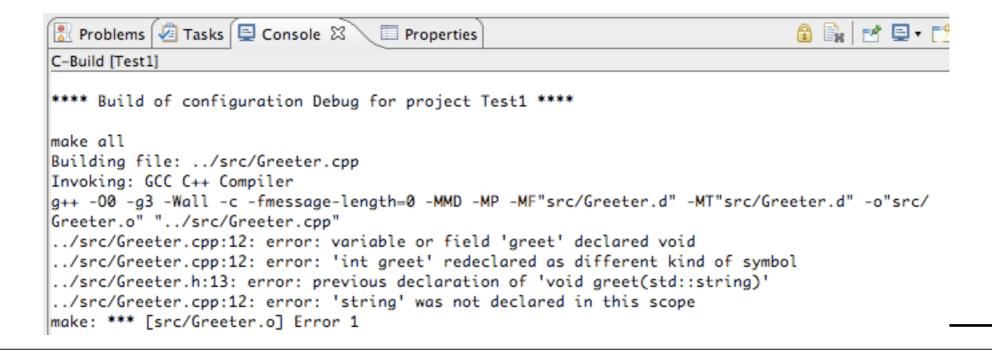
# Add a .cpp and .h files

- Add, for example a .h file that contains a function to greet a user, given his name, and add the prototype in the include
  - if the include is generated by Eclipse, it will provide automatically the #define guards



#### Compile

- Let's say the code has been written in the .cpp (including all the includes required, e.g. iostream and the greeter.h): compile using <u>Project</u> > <u>Build project</u>
- Check the compile errors (shown in the console panel and in the problems panel)





- Don't panic
- Start reading (carefully) the messages from the first to the last. Solve the first errors, perhaps they have an influence on the others.
  - In the example the first error is in the .cpp

#### Compile errors - cont.

	🕞 makefile 🔀 Test1.cpp 🕼 Greeter.h 🔂 Greeter.cpp 🛛 "70 🗖 🗍	
Eclipse shows where there's a problem	<pre>1/* 2 * Greeter.cpp 3 * 4 * Created on: 26-feb-2009 5 * Author: bertini 6 */ 7 8#include "Greeter.h" 9 10#include <iostream> 11 012void greet(string name) { 13 std::cout &lt;&lt; name &lt;&lt; std::endl; 14} 15</iostream></pre>	Read the message: the declaration does not
	Problems I Tasks I Console II Properties C-Build [Test1] **** Build of configuration Debug for project Test1 ****	match the prototype; the <u>string</u> was not declared
	<pre>make all Building file:/src/Greeter.cpp Invoking: GCC C++ Compiler g++ -00 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"src/Greeter.d" -MT"sr Greeter.o" "/src/Greeter.cpp" /src/Greeter.cpp:12: error: variable or field 'greet' declared void /src/Greeter.cpp:12: error: 'int greet' redeclared as different kind of /src/Greeter.h:13: error: previous declaration of 'void greet(std::strin /src/Greeter.cpp:12: error: 'string' was not declared in this scope make: *** [src/Greeter.o] Error 1</pre>	



- Correct the error: in this case it was necessary to add std:: to string (we are not using "using namespace std;" in this file !
  - Build again to check the correction



```
"70
                                                                    C Test1.cpp
                            h Greeter.h
                                           🖸 Greeter.cpp 🖾
a makefile
  1/*
  2 * Greeter.cpp
  3 *
  4 * Created on: 26-feb-2009
           Author: bertini
  5 *
  6 */
  7
  8#include "Greeter.h"
  9
 10#include <iostream>
 11
 12void greet(std::string name) {
       std::cout << name << std::endl;</pre>
 13
 14}
 15
📳 Problems 🙆 Tasks 📃 Console 🕱
                                   Properties
C-Build [Test1]
Finished building: ../src/Greeter.cpp
Building file: ../src/Test1.cpp
Invoking: GCC C++ Compiler
g++ -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"src/Test1.d" -MT"src
"../src/Test1.cpp"
Finished building: ../src/Test1.cpp
Building target: Test1
Invoking: MacOS X C++ Linker
g++ -o "Test1" ./src/Greeter.o ./src/Test1.o
Finished building target: Test1
```

### Run the program

 Use the menu <u>Run</u> > <u>Run as</u> > <u>Local C/C+</u> + <u>application</u>. Later on the program will appear in the <u>Run History</u>

avigate Search Project	Run Window Help		💁 📰 5 🔀 21 🏵 M		
Test1/src/Test1.cpp – Eclipse	🗞 Run 🏠 🖏 Cebug	第F11 /workspace 第F11 <sub>n Synchr</sub> <b>正</b> の	C/C++ 🏠 Resource 🕸 Debug		
> + ⇔ + ] ि +	Run History	•			
makefile C Test1.cpp X	Run As Run Configurations		on Server ①飞X R T I C/C++ Application s		
2// Name : Test1. 3// Author : Marco 4// Version : 5// Copyright : Your c 6// Description : Hello	Debug History Debug As Debug Configurations	*	<ul> <li>iostream</li> <li>Greeter.h</li> <li>std</li> <li>main() sint</li> </ul>		
<pre>7// 8 9#include <iostream> 10 11#include "Greeter.h" 12 13using namespace std; 14 15int main() { 16 cout &lt;&lt; "Hello, "; 17 string name = "Worl) 18 greet( name );</iostream></pre>	<ul> <li>Toggle Line Breakpoin</li> <li>Toggle Method Breakp</li> <li>Toggle Watchpoint</li> <li>Skip All Breakpoints</li> <li>Remove All Breakpoint</li> </ul>	oint	main() : int		
19 return 0; 20} 21	)	)4 ►			
🚼 Problems 🖉 Tasks 🗐 Console 🛛 🔲 Properties 🛛 📄 🗶 🎇 🕞 🚝 🛃 🖶 🛃 🛃 🕬 🛃 🕬					
<terminated> Test1 [C/C++ Local Application] /Users/bertini/Documents/workspace/Test1/Debug/Test1 (26/02/09 11.58) Hello, World</terminated>					



- In order to debug the program must be compiled so that additional information, useful for the debugger, is added to the files
- Add a breakpoint in Eclipse (right menu on the right side of the line), then execute the program in the debugger (Run > Debug as > Local C/C++ application)

	<pre>#include <iostream> using namespace std; int main() {</iostream></pre>						
				dl; //	prints	!!!Hello	World!!!
	Go to Annotation cppcheck	%1 ►					
•	Add Bookmark Add Task						
	✓ Show Quick Diff Show Line Numbers Folding	^ûQ ▶					
	Preferences						



- In order to debug the program must be compiled so that additional information, useful for the debugger, is added to the files
- Add a breakpoint in Eclipse (right menu on the right side of the line), then execute the program in the debugger (Run > Debug as > Local C/C++ application)

```
#include <iostream>
using namespace std;
int main() {
    cout << "!!!Hello World!!!" << endl; // prints !!!Hello World!!!
    return 0;
}</pre>
```



- In order to debug the program must be compiled so that additional information, useful for the debugger, is added to the files
- Add a breakpoint in Eclipse (right menu on the right side of the line), then execute the program in the debugger (Run > Debug as > Local C/C++ application)

Run Window Help		
🗞 Run 🍬 Debug	12 X X 11	pse_Workspace_Corsi/esercizi
<b>Q</b> <sub>≡</sub> Profile		
Profile History	•	
Profile As Profile Configurations	•	
Run History	•	
Run As Run Configurations	•	
Debug History	•	
Debug As	•	I Local C/C++ Application
Debug Configurations		

# Debug the program

- In order to debug the program must be compiled so that additional information, useful for the debugger, is added to the files
- Add a breakpoint in Eclipse (right menu on the right side of the line), then execute the program in the debugger (Run > Debug as > Local C/C++ application)

😑 🔿 🔿 🕞 Debug	- TestDebug/src/TestDebug.cpp - Eclipse - /Users/b	ertini/Documents/workspace/Eclipse_Worksp	ace_Corsi/esercizi				
] 📬 • 🖆 • 📄 🕼 🇁 🛑 ] 🏇 • 💽 • 🍫 • ] 🅭 🅭	। 😂 🖉 🖋 • ] 🍠 🐦 🖉 ] 🐓 न 🖓 • 🐤 🔶 •			😭 棼 Debug 뎞 C/C++			
🏇 Debug 🕱 🕴 Servers 📎	🗖 🖓 🐨 😸 🤧 🐟 🍕 🐜 🔳 🖉 🐂	🗱 🗣 Breakpoints 👷 Expression	s 🚈 Interactive Console	👬 Registers 🛋 Modules 🛛 🗖 🗖			
TestDebug [C/C++ Application]				() 🐗 🖂 🖇 💥 📑 😴 💙			
TestDebug		Name Type		Value			
Thread [1] (Suspended : Breakpoint) main() at TestDebug.cpp:13 0x100000d5							
gdb	4						
<b>9</b>							
C TestDebug.cpp 🕱	······································		- 🗆 🗄 Outline 🕅	;≥ ↓ <u>2</u> ≥ ×			
//			iostream				
// Name : TestDebug.cpp			std				
// Author :			main() : int				
// Version :							
<pre>// Copyright : Your copyright notice // Description : Hello World in C++, Ansi</pre>	-s+v]e						
//							
<pre>#include <iostream> using namespace std;</iostream></pre>							
using numespace sta;							
<pre>int main() {</pre>							
<pre>cout &lt;&lt; "!!!Hello World!!!" &lt;&lt; endl;</pre>	// prints !!!Hello World!!!						
return 0;							
3							
📮 Console 🕴 🖉 Tasks 🗜 Problems 🜔 Executables 🐺 Debug Output 🗳 Browser Output 🚺 Memory							
TestDebug [C/C++ Application] TestDebug							

# Some style guidelines

- There are a plethora of C++ coding style recommendations, sometimes even contradictory.
- Two very good recommendations:
- I. Any violation to the guidelines is allowed if it enhances readability.
- 2. The rules can be violated if there are strong personal objections against them.

# Naming conventions

- Names representing types must be in mixed case starting with upper case: follow this rule when writing classes.
- Variable names must be in mixed case starting with lower case (like Java).
- Names representing methods or functions must be verbs and written in mixed case starting with lower case (like Java).



- Names representing namespaces should be all lowercase.
- All names should be written in English.

#### Files

- C++ header files should have the extension .h (preferred) or .hpp. Source files can have the extension .c++, .C, .cc or .cpp.
- A class should be declared in a header file and defined in a source file where the name of the files match the name of the class.
- Header files must contain an include guard.
- Include statements must be located at the top of a file only.