CEE432/CEE532
Developing Software for Engineering Applications
MAE 541
CAD Tools for Engineers

Lecture 1: Overview
(Chapter 1)

Course Objectives

- Understand the link between engineering software development and structural engineering
- Learn the concepts associated with
  - object-oriented problem solving
  - algorithm development, implementation and debugging
  - resource allocation and performance tuning in engineering software
- Understand basics of finite element analysis

Blackboard Web Site

- Use http://myasucourses.asu.edu
- There will be no handouts in class
- Read the course syllabus

C++ Textbooks

- **Horstmann and Budd, *Big C++*, Wiley.**
- Deitel and Deitel, *C++ How to Program*, 4th Ed, Prentice Hall.
- Shtern, *Core C++ - A Software Engineering Approach*, Prentice Hall.
- Lipmann et. al., *C++ Primer*, Addison Wesley
- Stroustrup, *The C++ Programming Language*, Addison Wesley.

Books on STL

- **Josuttis, *The C++ Standard Library: A Tutorial and Reference*, Addison-Wesley.**
- Musser, Derge and Saini, *STL Tutorial and Reference Guide*, 2nd Edition, Addison Wesley.

Numerical Analysis Books

- Chapra and Canale, *Numerical Methods for Engineers*, 4th Ed, McGraw-Hill.
- **Press, Teukolsky, Vetterling and Flannery, *Numerical Recipes in C++: The Art of Scientific Computing*, Cambridge University Press.**

FEA Books

- **Rajan, *Introduction to Structural Analysis & Design*, Wiley.**
- Several books available in Noble library

Ethics

- Know and practice ethical behavior
- Assignment 0: Read, understand and acknowledge the class ethical policy. There are serious repercussions if you do not practice ethical behavior.
- I am here to guide you and help you help yourself.

Computing Resources

- Operating Systems
  - Microsoft Windows 7 or 8
  - Linux
- C++ Compilers and IDE
  - Windows: Microsoft Visual Studio 2012
  - Linux: Eclipse with Intel C++, gcc

Terminology

- Program - A set of instructions
- OS - Operating System
- Programming Language
- Compiler
- Executable image
- We will learn ISO C++11 (though not all the latest language features)

Engineering Software

- Usually is
  - Described by a PDE, ODE or integral equations
  - Requires a numerical, approximate solution
  - Needs to be as general as possible
- Usually requires large resources
  - Memory
  - Compute intensive
  - Input and output are graphics-based

Numerical Simulation

- Requires an array of solution techniques and has traits such as
  - One and multi-dimensional arrays
  - Real and complex numbers
  - Precision is an issue
  - Linear and nonlinear equation solvers
  - Numerical integration and differentiation
  - Statistical operations

Algorithm

- Describes the solution steps involved in a numerical analysis procedure
- Should be detailed enough so that it is
  - Accurate
  - Robust
  - Can be translated readily into a computer program or module

An Example

- Find the maximum of a set of numbers.
  - Input: A set of numbers
  - Output: A single number that is the maximum value of the input set.
- Assumptions
  - Input set contains real numbers
  - Numbers are not necessarily unique
  - Numbers are randomly placed (not sorted)

Algorithm

- Step 1: Ask user for size of input set, *n*.
- Step 2: Read in the input set storing the values in a vector *a* of size *n*.
- Step 3: Set the maximum value, $a_{max} = -10^{20}$.
- Step 4: Loop thro' all the numbers, *i=1 through n*.
- Step 5: Compare a_i to a_{max}. If a_i is greater than a_{max} set $a_{max} = a_i$.
- Step 6: End loop.
- Step 7: Display a_{max}.

What are Objects?

- Objects are routinely identified by human beings. They are given a name, defined as having properties (what is it?) and capabilities to do something (what does it do?). Computer scientists term these characteristics as *attributes* and *behavior*.

Object-Oriented (OO) Numerical Analysis

- The proficiency of higher-level OO model should provide the software designer with real-world, programmable components, thereby reducing software development costs.
- Its capability to share and reuse code with OO techniques will reduce time to develop an application.
- Its capability to localize and minimize the effects of modifications through programming abstraction mechanisms will allow for faster enhancement development and will provide more reliable and more robust software.
- Its capability to manage complexity allows developers to address more difficult applications.

Terminology

- Building a Program
  - Source Code - High-level language
  - Compiler – Binary code
  - Linker – Executable image

Terminology

- Integrated Development Environment
  - Editor
  - Compiler
  - Linker
  - Debugger

Building a Program

- Microsoft Visual C++ IDE
  - Installation
  - Executing
  - Exiting
  - Saving and handling files

The Simplest C++ Program

```
#include <iostream>
using std::cout;
int main ()
{
  cout << “Hello World”;
  return (0);
}
```

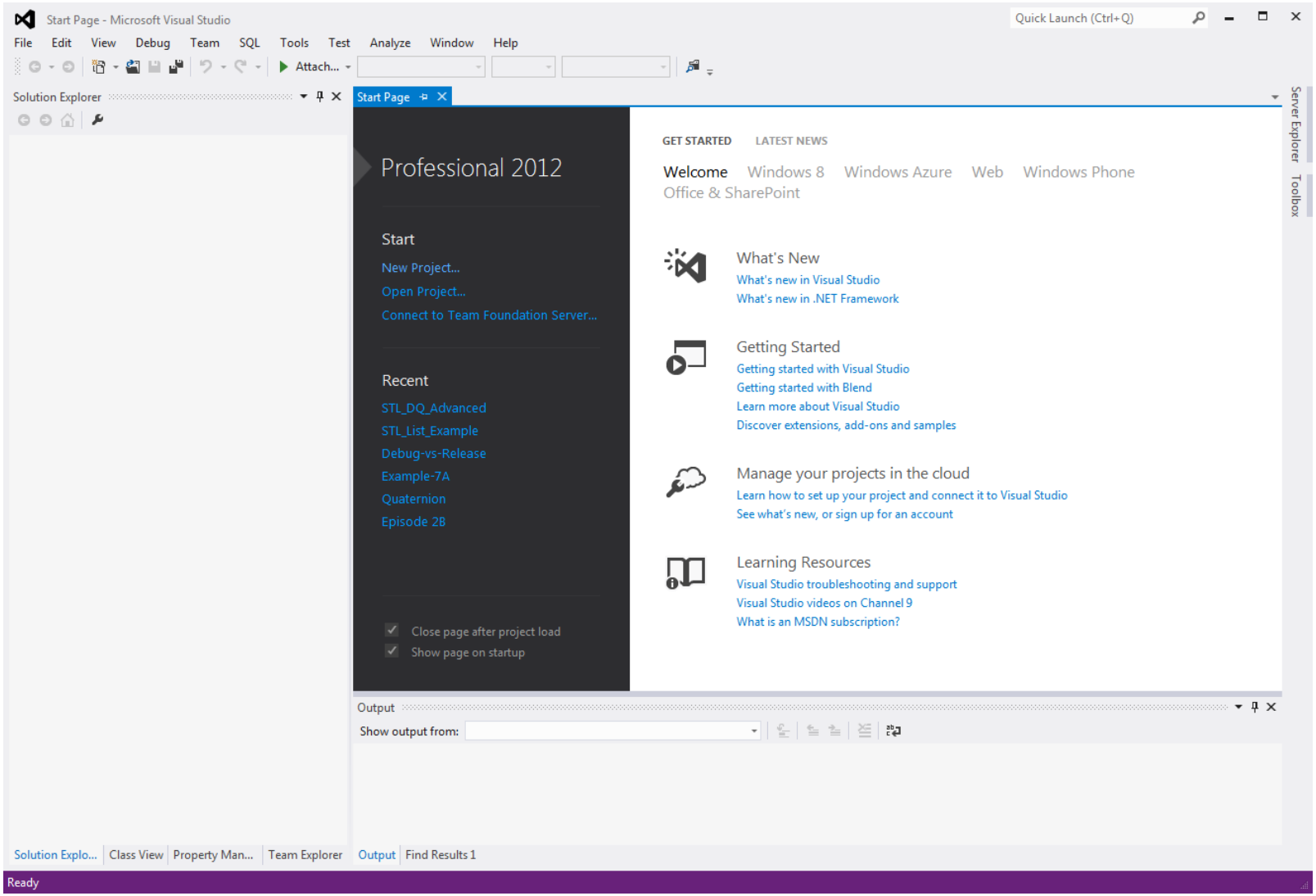
Creating a Project

- The Visual C++ Environment
- What is a project?
- Creating a simple program
- Compiling
- Building (Compiling and Linking)
- Executing the program

Exercise: Using Visual C++

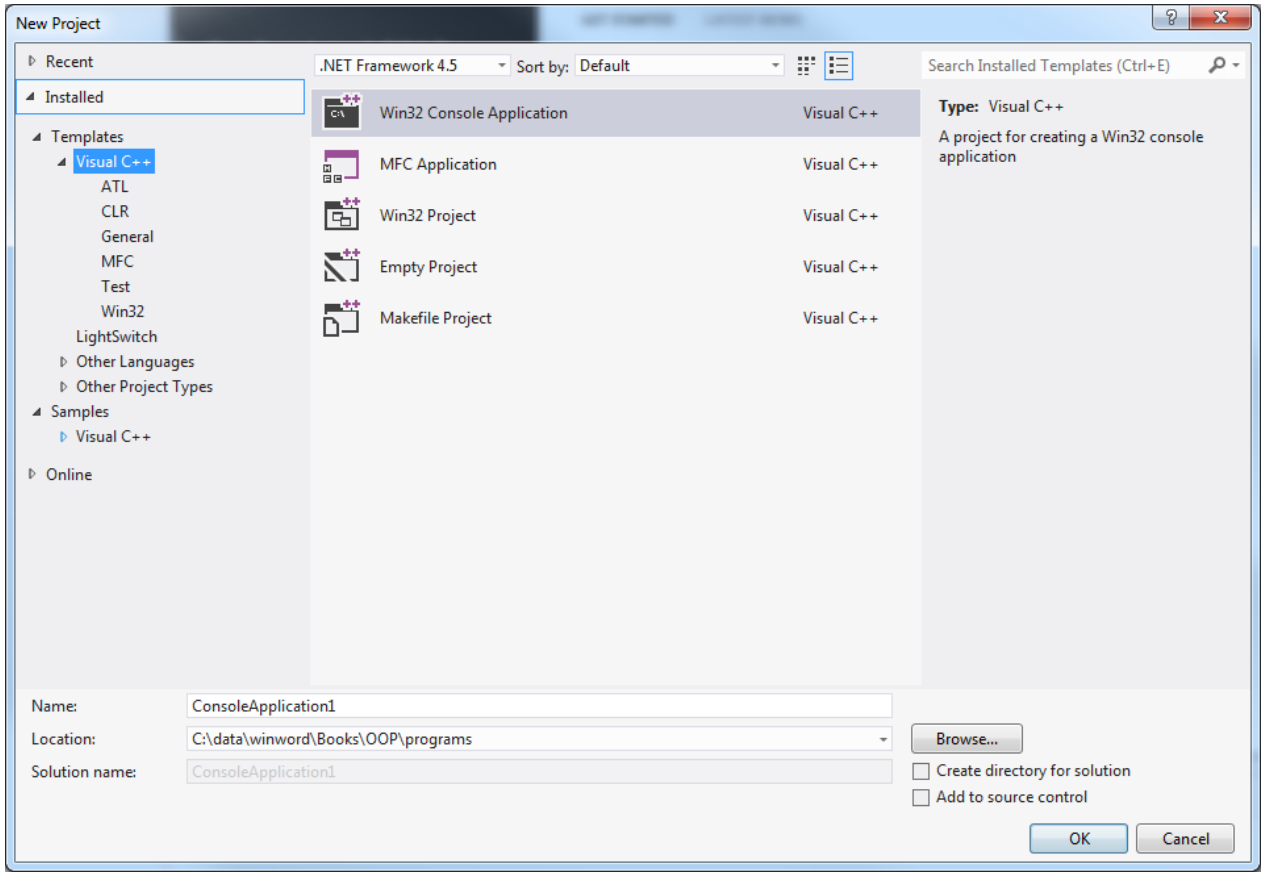
- Launch Visual C++
- Create the simple program
- Execute the simple program
- Modify the simple program

Startup Screen



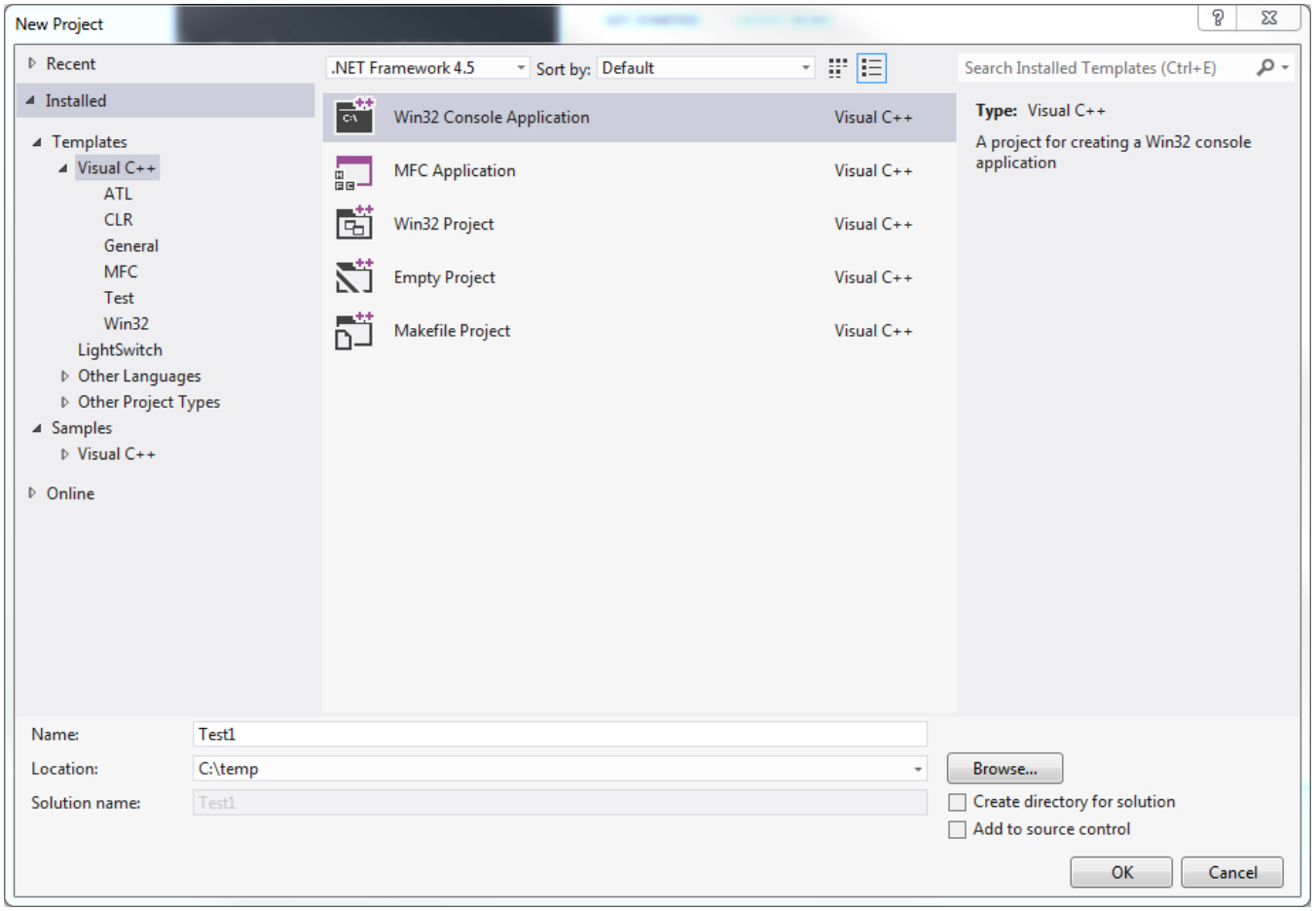
Creating a Project

- Click on File, New, Project



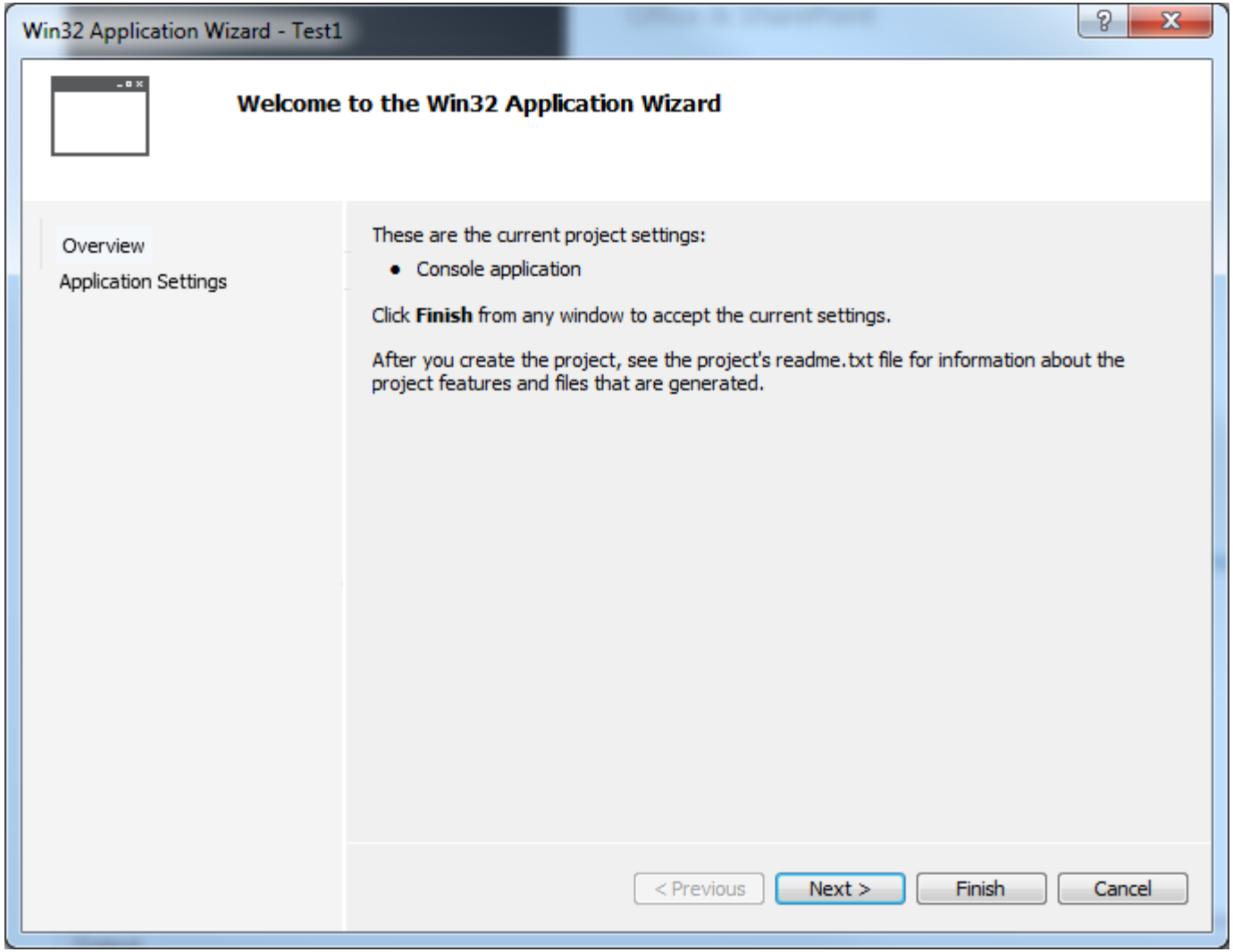
- Project Type: Visual C++ Projects
- Template: Win32 Console Project

Creating a Project



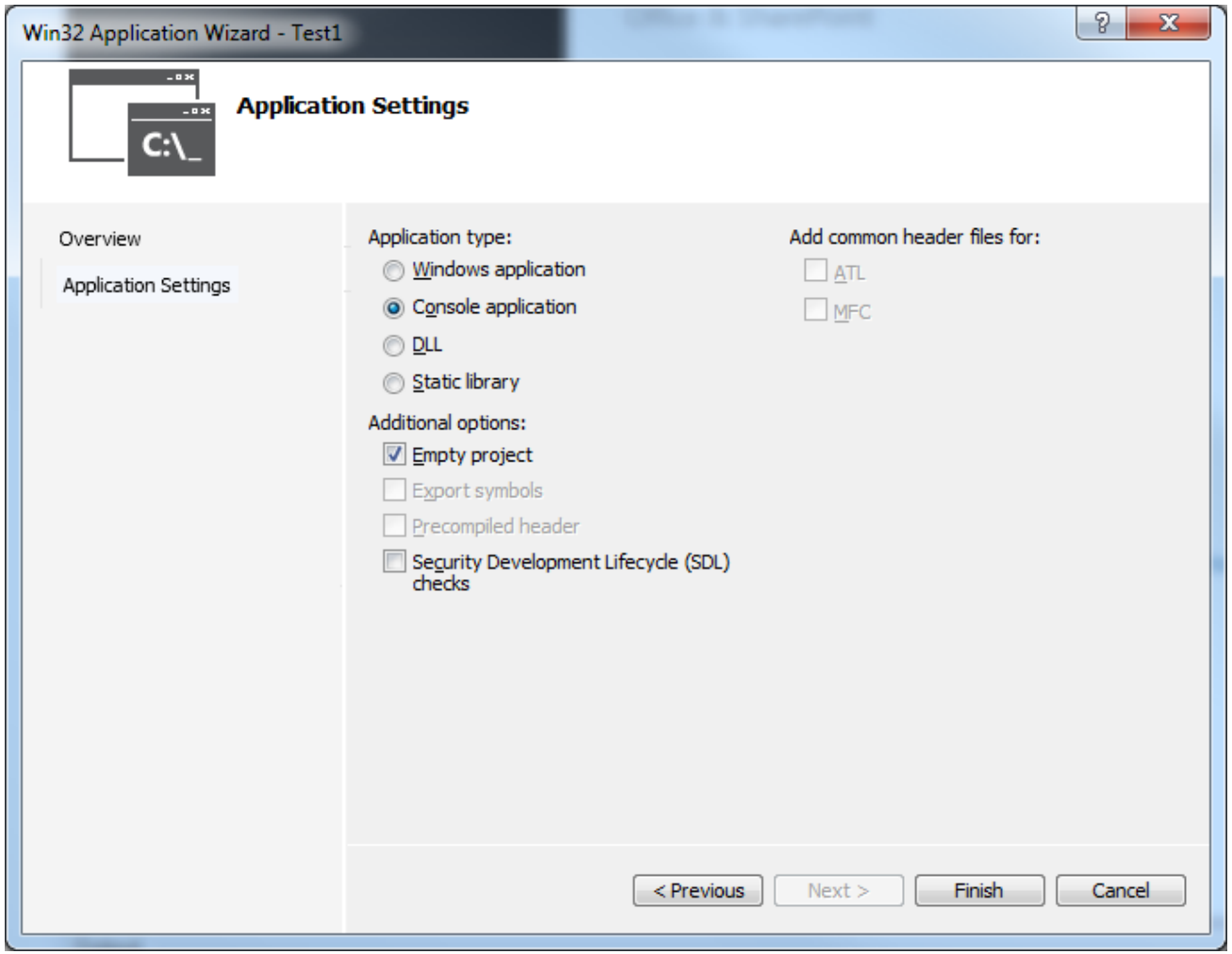
- Select Location and Type Name
- Click OK

Creating a Project



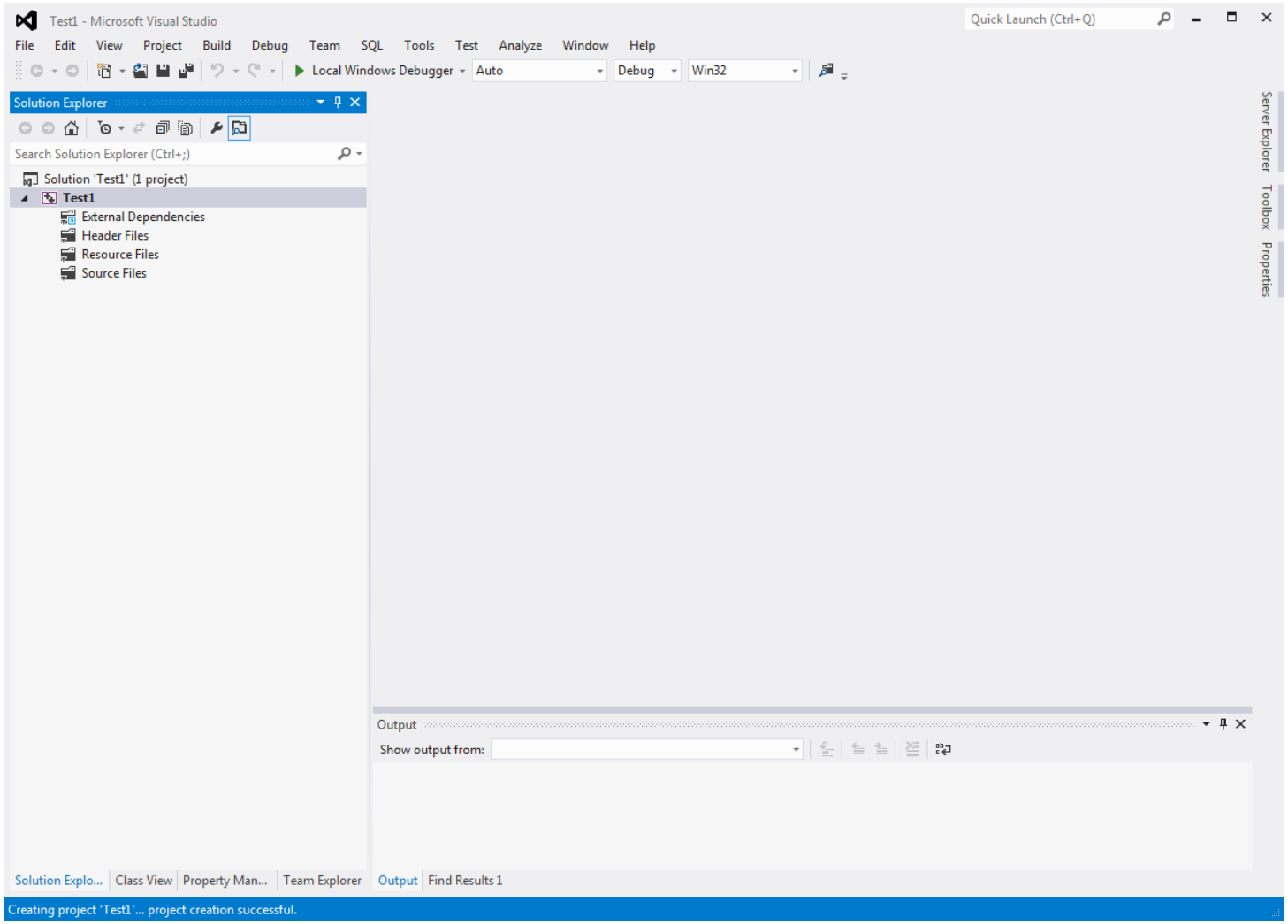
- Click Next

Creating a Project

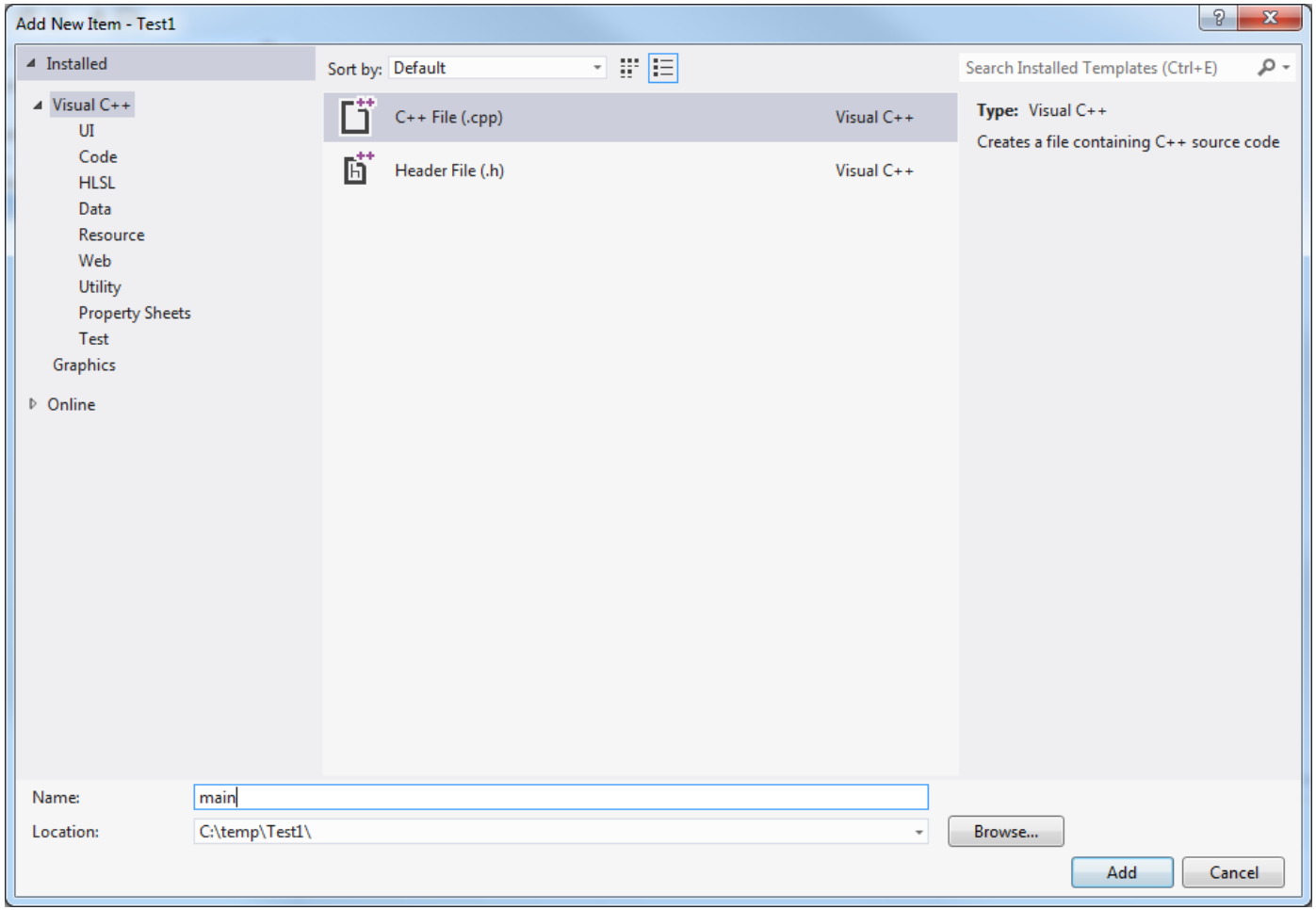


- Check Empty project.
- Uncheck SDL.
- Click Finish.

Creating a Project

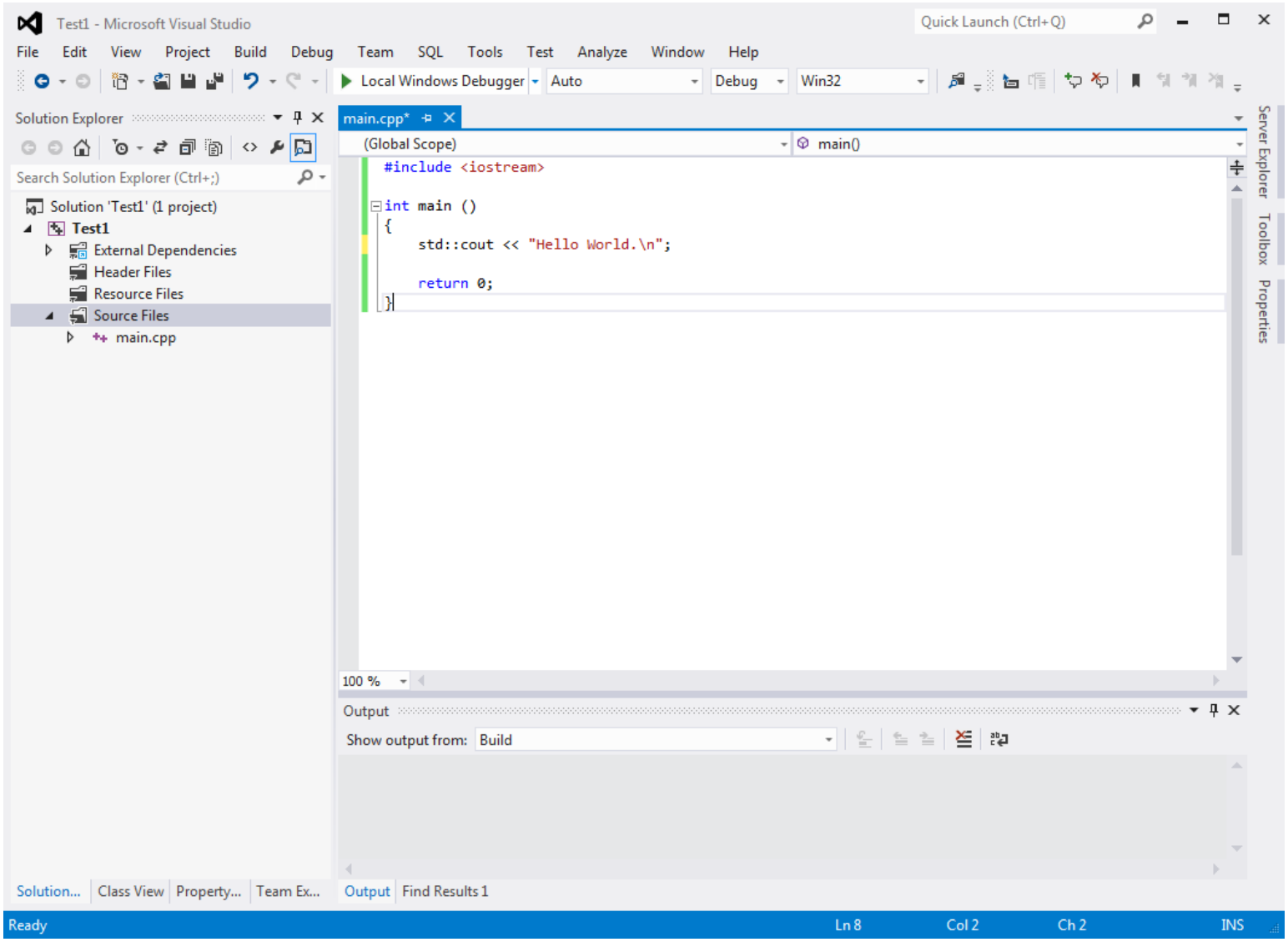


Creating a Project



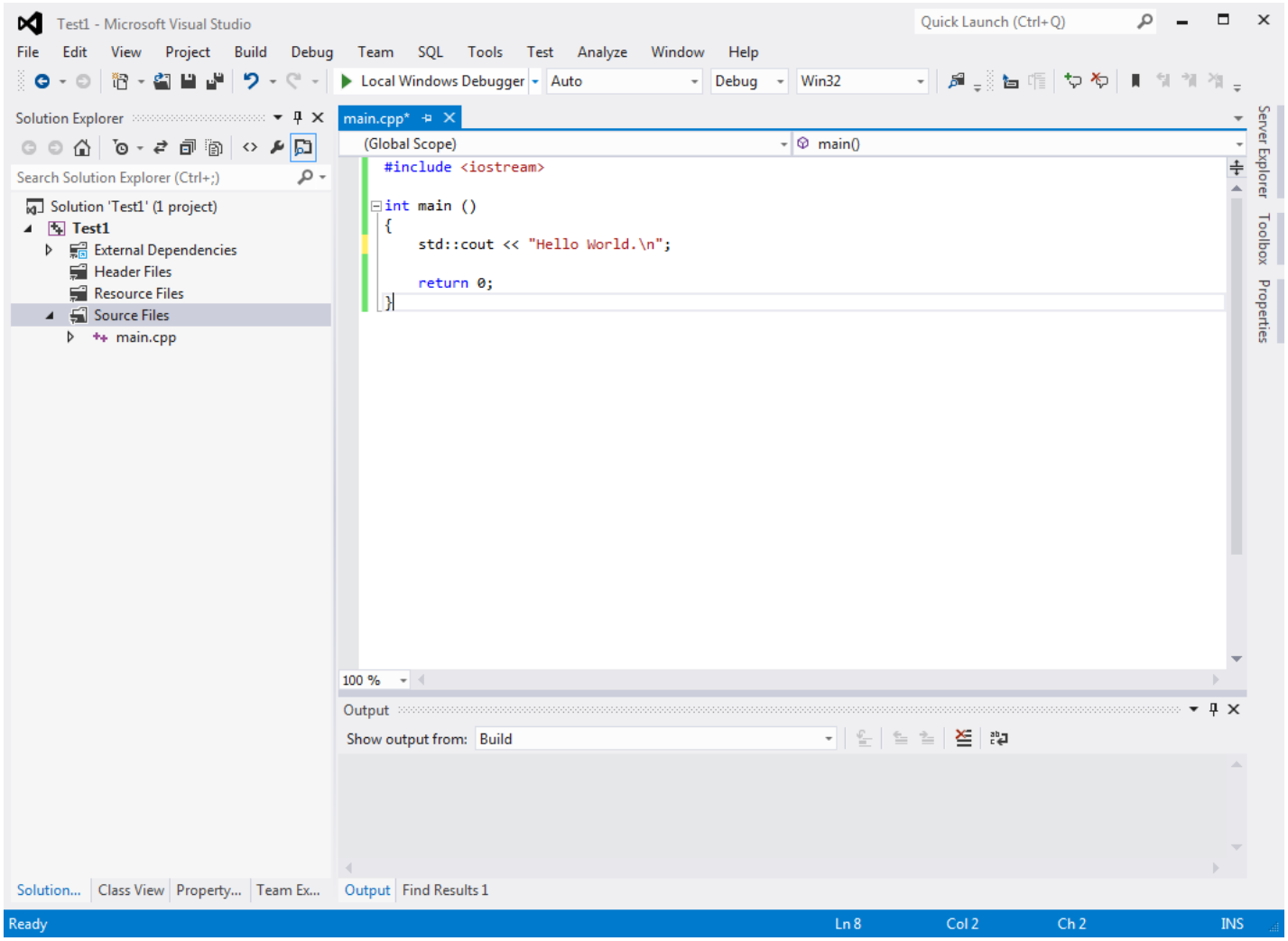
- Right click Source Files
- Select Add, New Item.
- Enter Name.
- Click Add.

Creating a Project



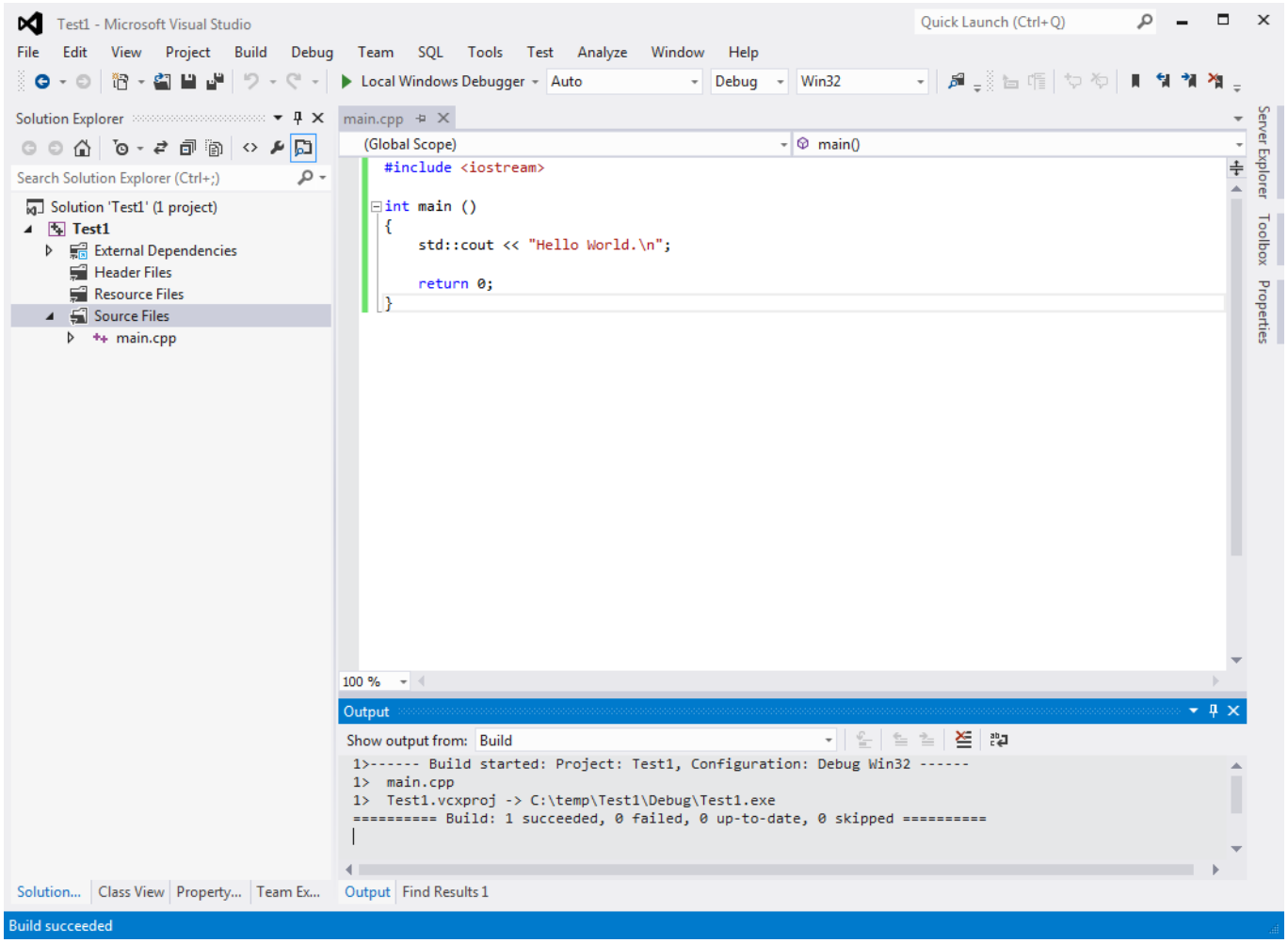
- Type in the program

Creating a Project



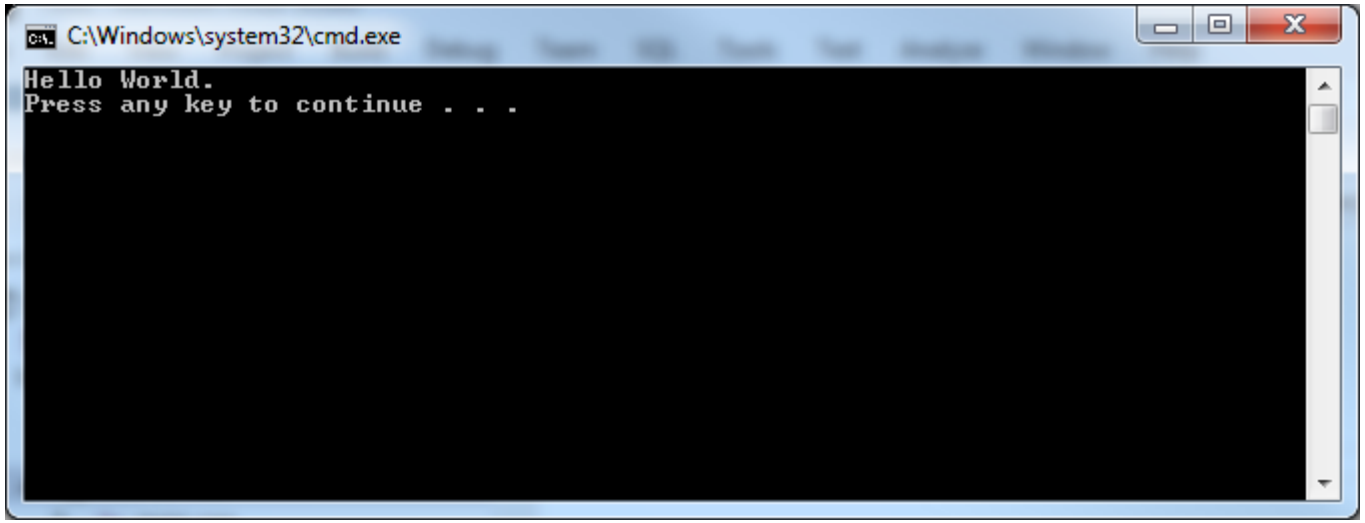
- Click on Build, Build Test1.

Executing the Project



- Click on Debug, Start Without Debugging

Executing the Project



Summary

- Computerspeak
  - Computer
  - Hardware
  - Software
  - Firmware
  - Operating System

Summary

- Can you tell the differences between
  - Programming language
  - Compiler
  - Linker
  - Executable image
  - Source Code
  - Editor
  - Debugger