

**Baton Rouge Community College STEM (Computer Science)**

**High Performance Computing Curriculum Module**

**Debra Miller Borskey, CS Associate Professor**

**LSU LASIGMA RET Summer 2014**

**Introduction to C++ using Visual Studio**

**Course: CSCI 192 Introduction to Computers: Programming Logic and Design**

**Purpose: To help students become familiar with programming in C++ using Visual Studio**

**Objectives: At the end of this module students will be able to:**

- Understand and use the basic programming constructs of C++
- Manipulate C++ data types, such as integers and strings
- Isolate and fix common errors in C++ programs
- Apply object-oriented approaches to software problems in C++
- Write small-scale C++ programs using the above skills
- Use an Integrated Development Environment to program in C++

**“Hello World”**

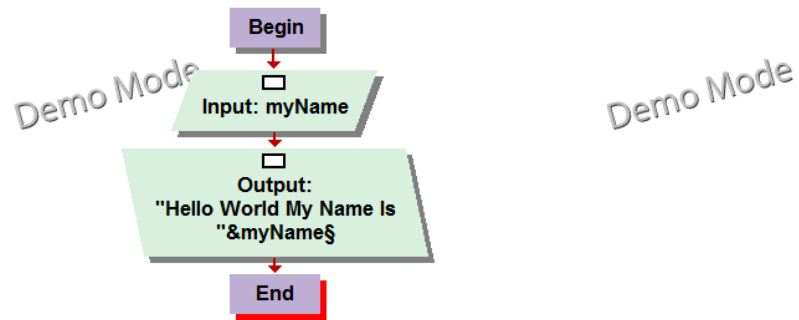
Using your notes and the example below, provide the flow chart logic for a C++ program that produces the following output:

**“Hello World My Name is Your Complete Name”**

Demo Mode

Demo Mode

Der



```
Please enter your first and last name in quotes "Debra Borskey"  
Hello World My Name Is Debra Borskey
```

**Provide the pseudo code of the program typed below.**

Start

Declarations

String myName

input myName

output "Hello World My Name is " myName

stop

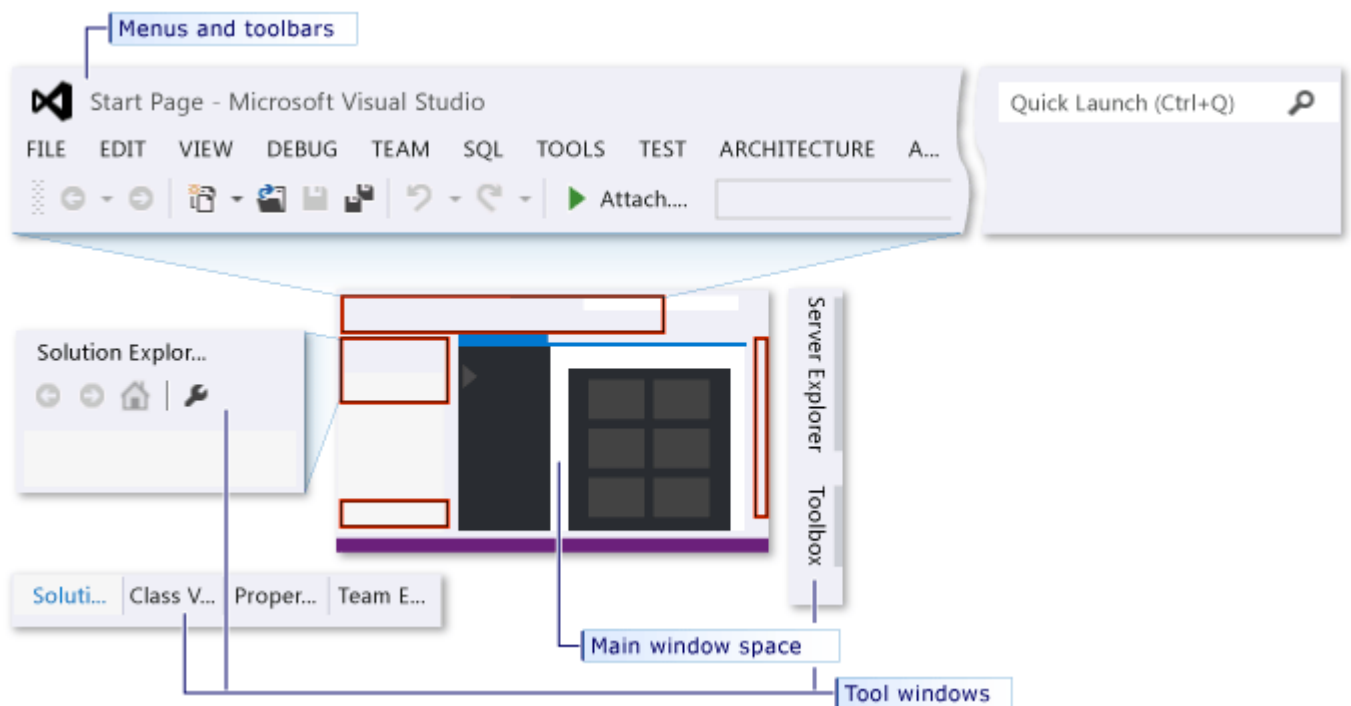
# Getting Started with C++ in Visual Studio

## Visual Studio 2013

By completing this walkthrough, students will become familiar with many of the tools and dialog boxes that may be used when developing apps with Visual Studio. Students will create a simple “Hello, World” app while learning more about working in the integrated development environment (IDE).

### Open Visual Studio 2013

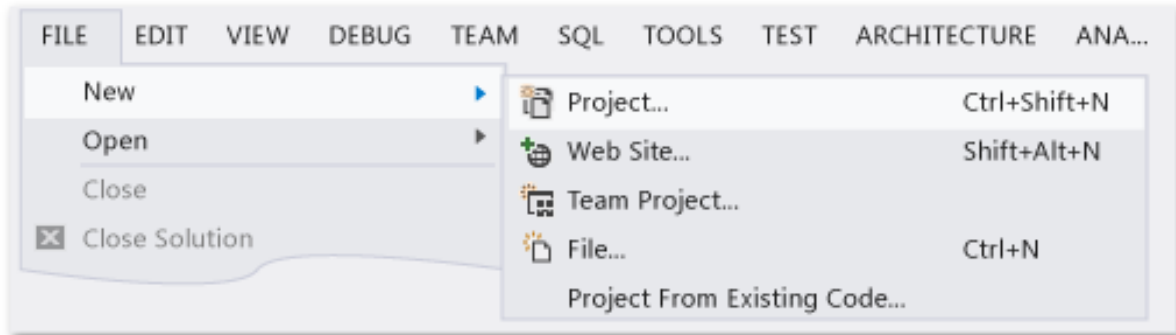
After opening Visual Studio, students see the three basic parts of the IDE: tool windows, menus and toolbars, and the main window space. Tool windows are docked on the left and right sides of the app window, with Quick Launch, the menu bar, and the standard toolbar at the top. The center of the application window contains the Start Page. When students open a solution or project, editors and designers appear in this space.



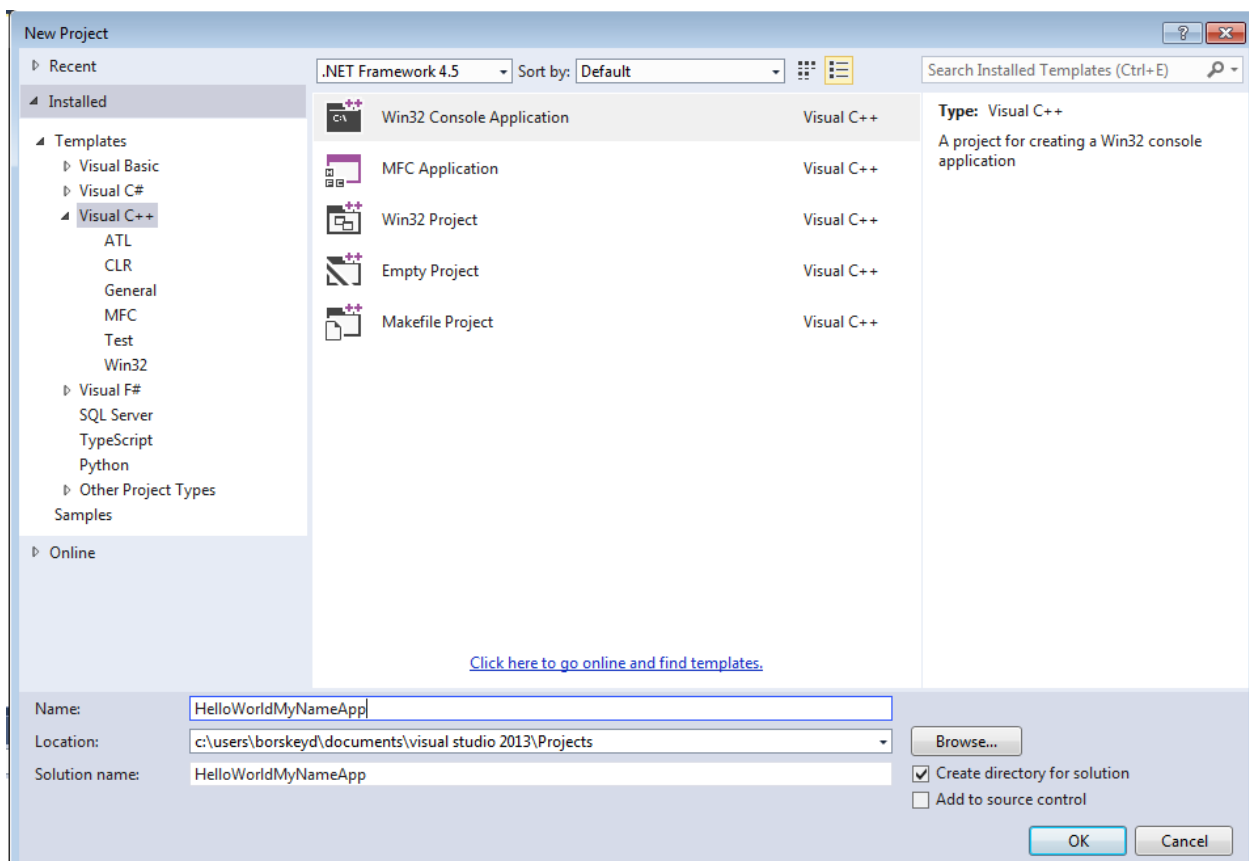
When creating an app in Visual Studio, first create a project and a solution. For this example, students will create a Windows console app.

### To create a console app

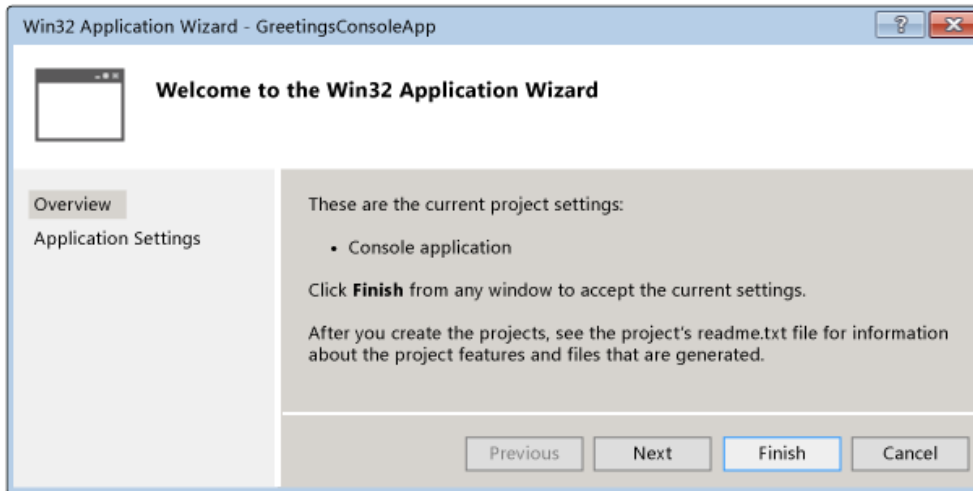
1. On the menu bar, choose File, New, Project.



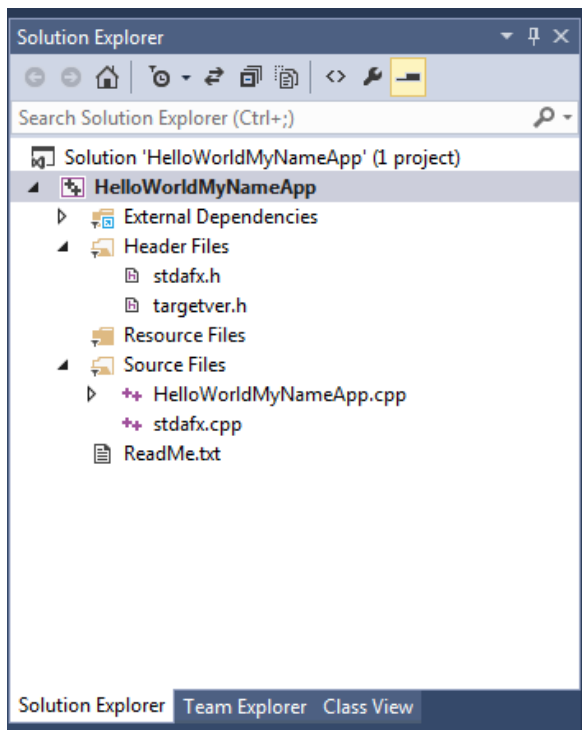
2. Select the Visual C++ category, choose the Win32 Console Application template, and then name the project HelloWorldMyNameApp. Next, click OK.



- When the Win32 Application Wizard appears, choose the Finish button.



- The HelloWorldMyNameApp project and solution, with the basic files for a Win32 console app, are created and automatically loaded into Solution Explorer. The HelloWorldMyNameApp.cpp file is opened in the code editor. The following items appear in Solution Explorer:



- Next, students will add code to display the word "Hello World My Name is Your Name" in the console window.

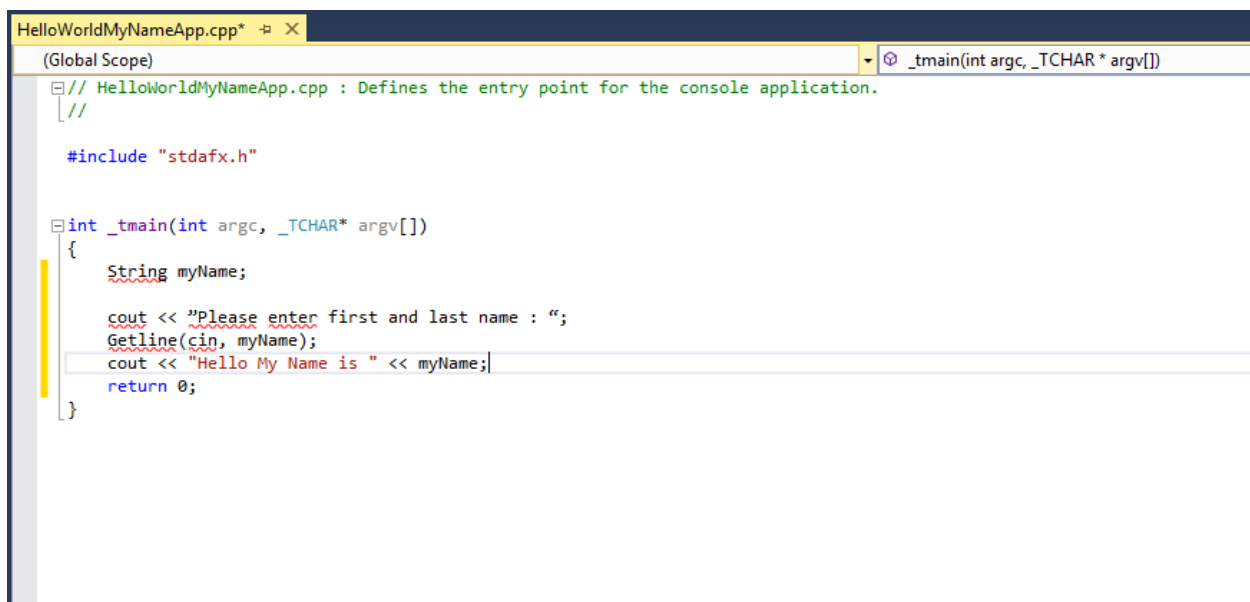
## To enter your name and display “Hello My Name is Your Name” in the console window

1. In the GreetingsConsoleApp.cpp file, enter a blank line before the line return 0; and then enter the following code:

```
String myName;

cout << "Please enter first and last name: ";
Getline(cin, myName);
cout << "Hello My Name is " << myName;
```

Notice the red squiggly lines and an error message appears when pointing to each one.



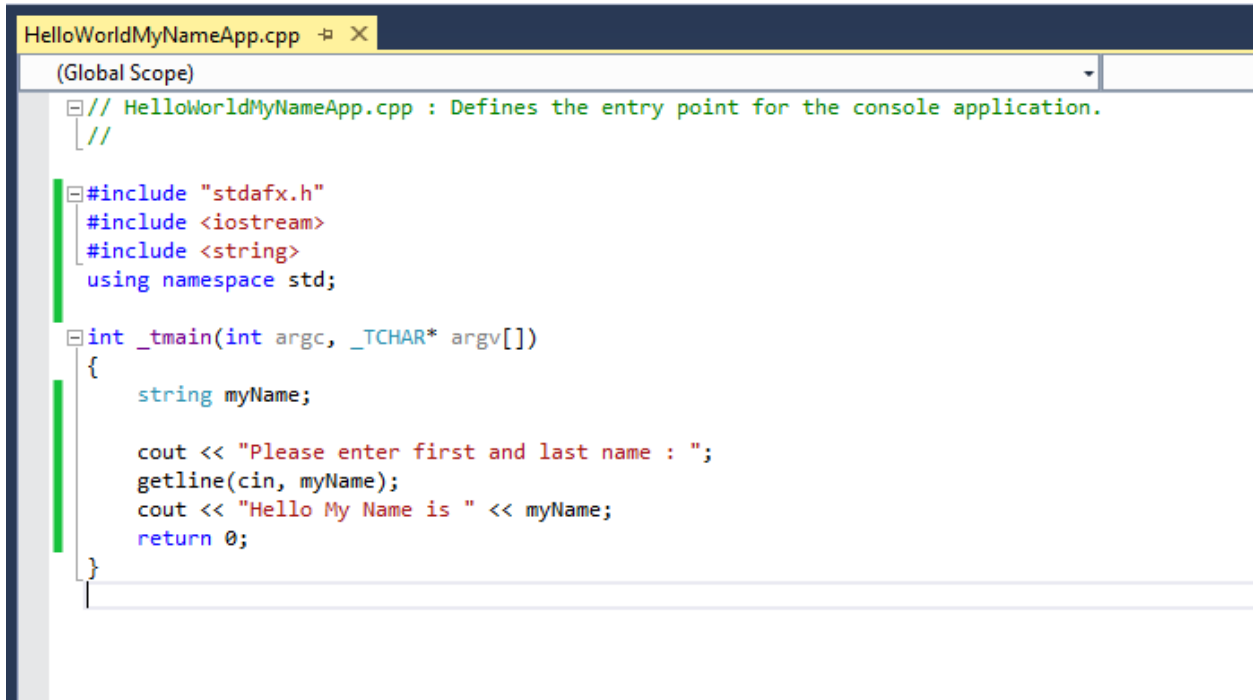
2. The error messages also appear in the Error List window. Display the window by, on the menu bar, choosing View, Error List.

To include the iostream header and the string header, enter the following code after #include "stdafx.h":

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

Notice that a box appeared when entering code, suggestions are provided for the characters entered. This box is part of C++ IntelliSense, which provides coding prompts, including listing class or interface members and parameter information. Code snippets, which are pre-defined blocks of code may also be used. The red squiggly lines disappear when each error is fixed.

3. Save the changes to the file.



```
>HelloWorldMyNameApp.cpp [X]
(Global Scope)
// HelloWorldMyNameApp.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include <iostream>
#include <string>
using namespace std;

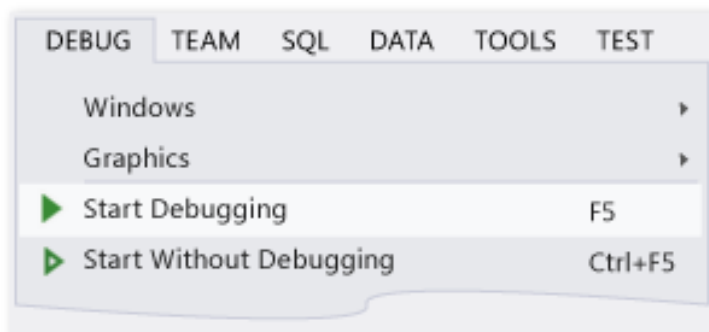
int _tmain(int argc, _TCHAR* argv[])
{
    string myName;

    cout << "Please enter first and last name : ";
    getline(cin, myName);
    cout << "Hello My Name is " << myName;
    return 0;
}
```

Debug HelloWorldMyNameApp to see if the phrase "Hello World My Name is Your Name" appears in the console window.

## To debug the app

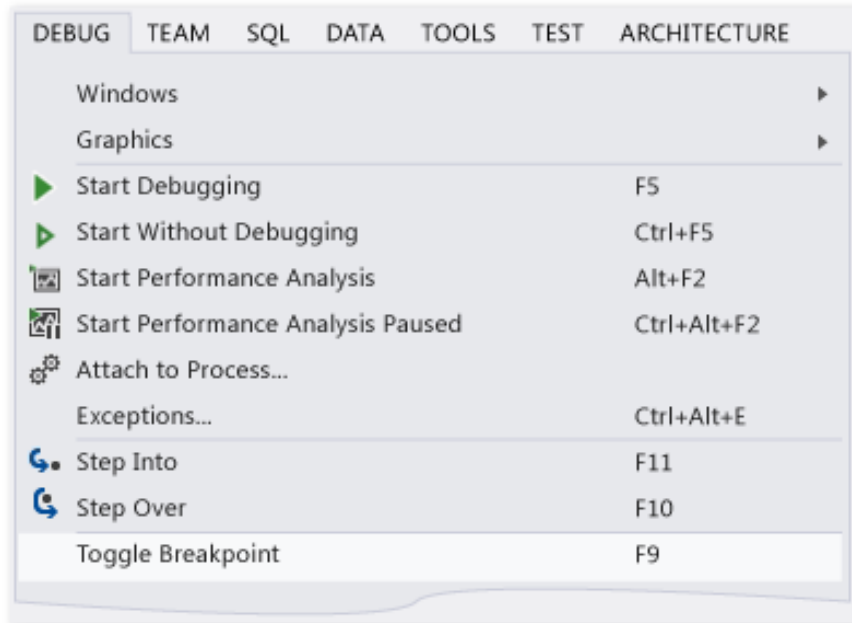
1. Start the debugger.



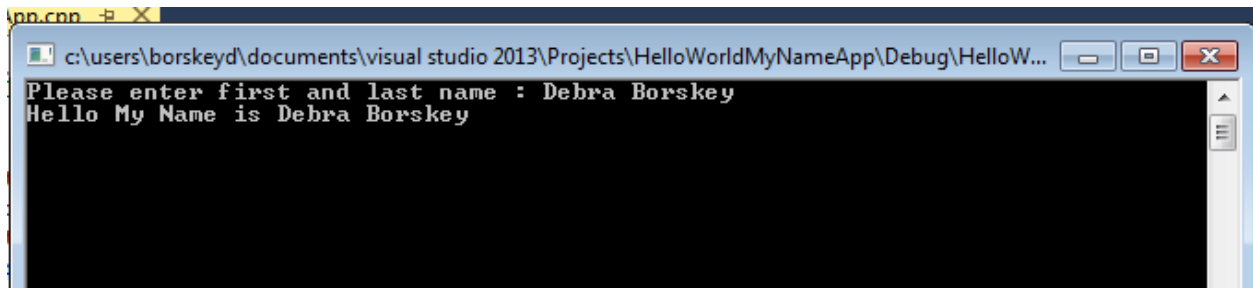
2. The debugger starts and runs the code. The console window (a separate window that looks like a command prompt) appears and prompts the user to enter a name. Once the name is entered, for a few seconds the console window appears. However the window closes quickly when the debugger stops running. To see the text, set a breakpoint to stop program execution.

## To add a breakpoint

1. Add a breakpoint from the menu bar at the line return 0;. Also users can just click in the left margin to set a breakpoint.



2. A red circle appears next to the line of code in the far left margin of the editor window.
3. Choose the F5 key to start debugging.
4. The debugger starts, and a console window appears prompting the user for a first and last name, then showing the phrase “Hello World My Name is Your Name”.



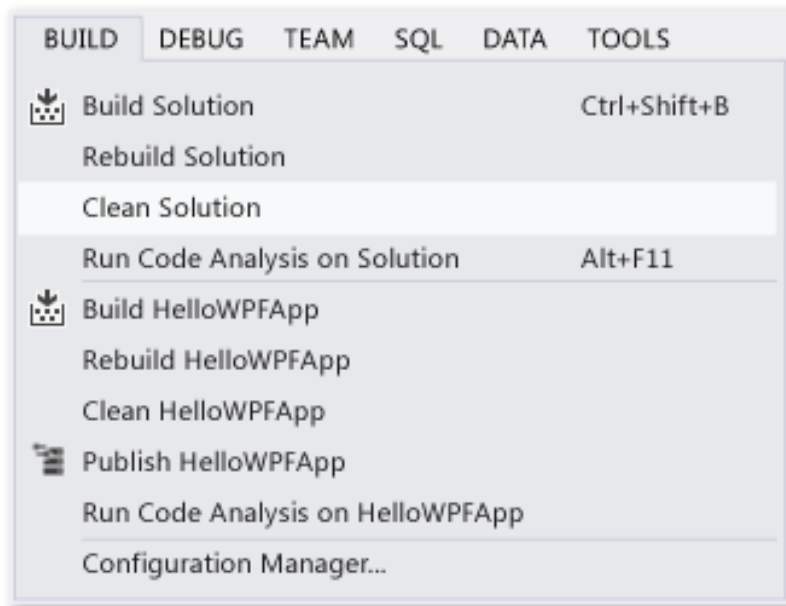
5. Press SHIFT + F5 to stop debugging.
-



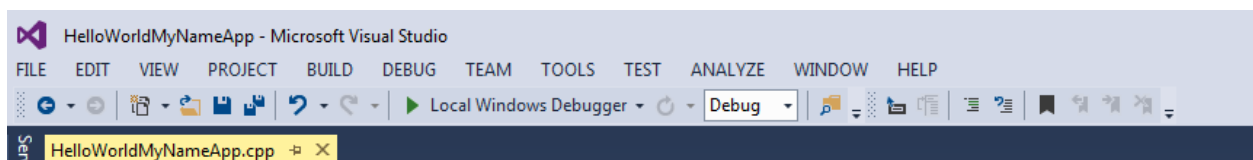
Now that users have verified everything works, next students may prepare a release build of the app.

### To clean the solution files and build a release version

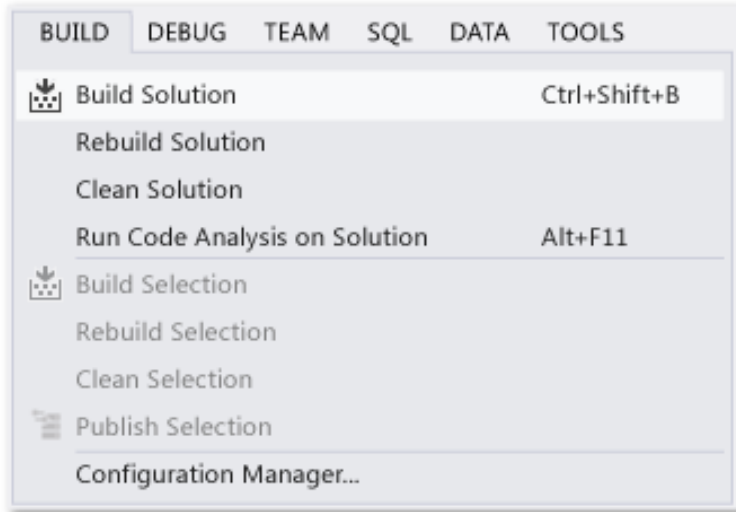
1. From the menu bar, delete intermediate files and output files that were created during previous builds.



2. Change the build configuration for HelloWorldMyNameApp from Debug to Release.



Build the solution.

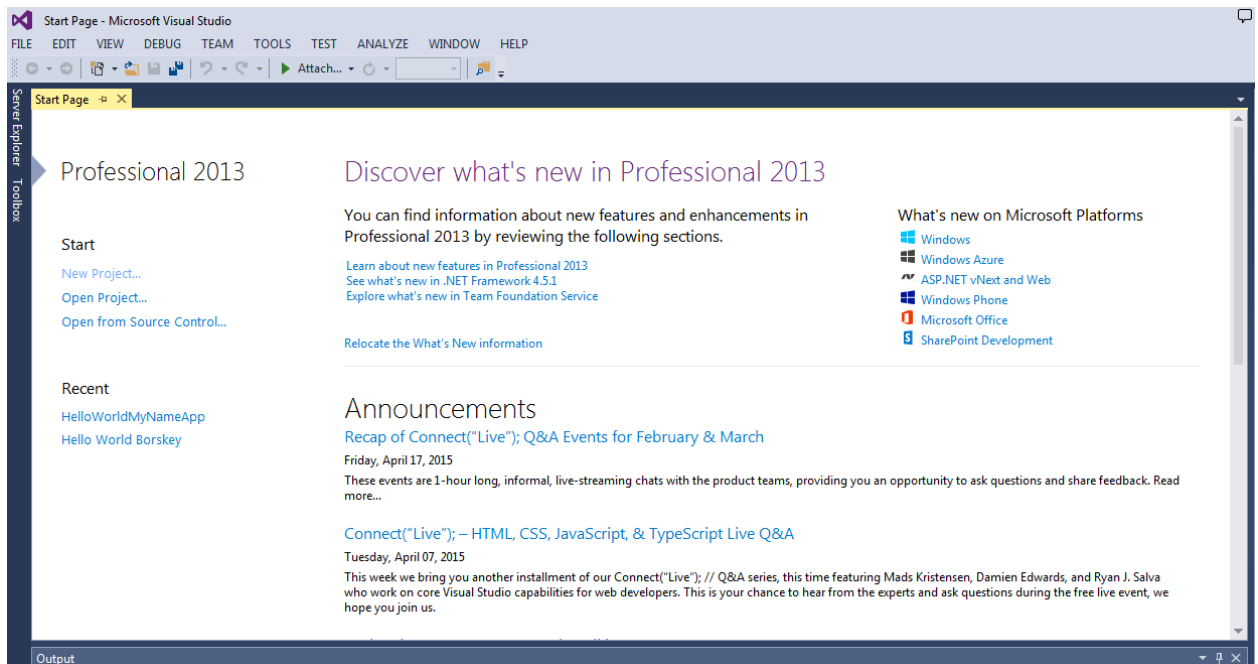


Congratulations on completing the first walkthrough!

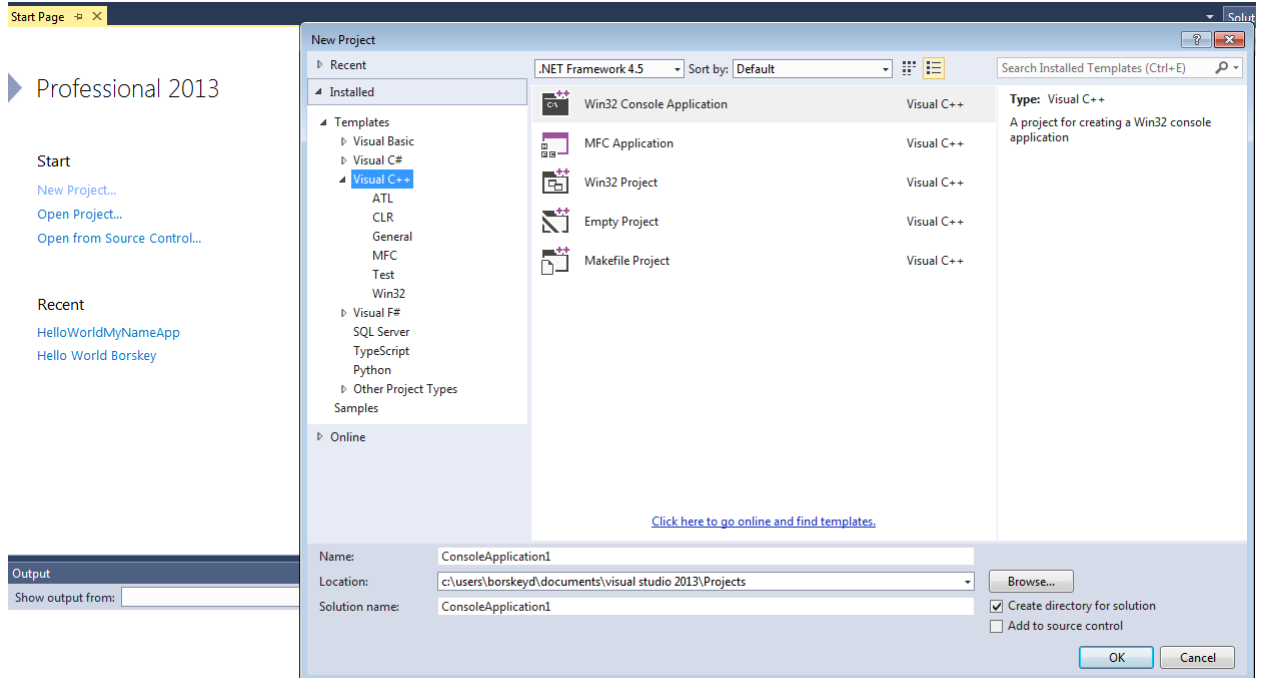
**Students may also try using Visual Studio and the steps provided to run, compile, debug and save the program. Remember, format work nicely.**

Steps may vary to Create, Check for Errors and Execute a C++ Program

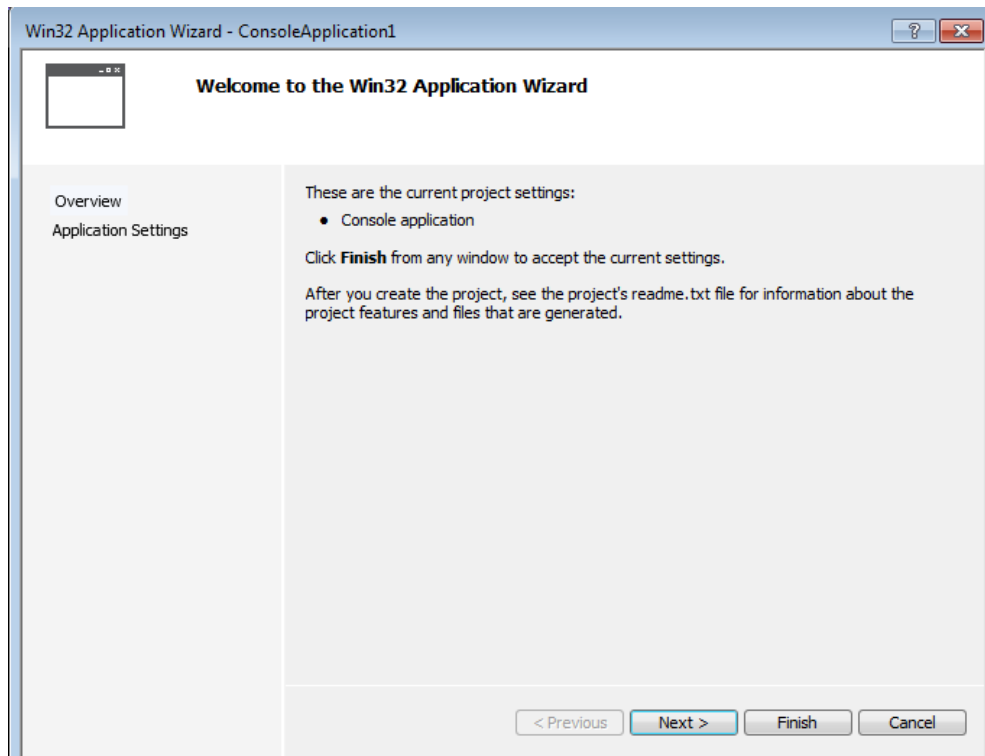
1. Open Visual Studio 2013



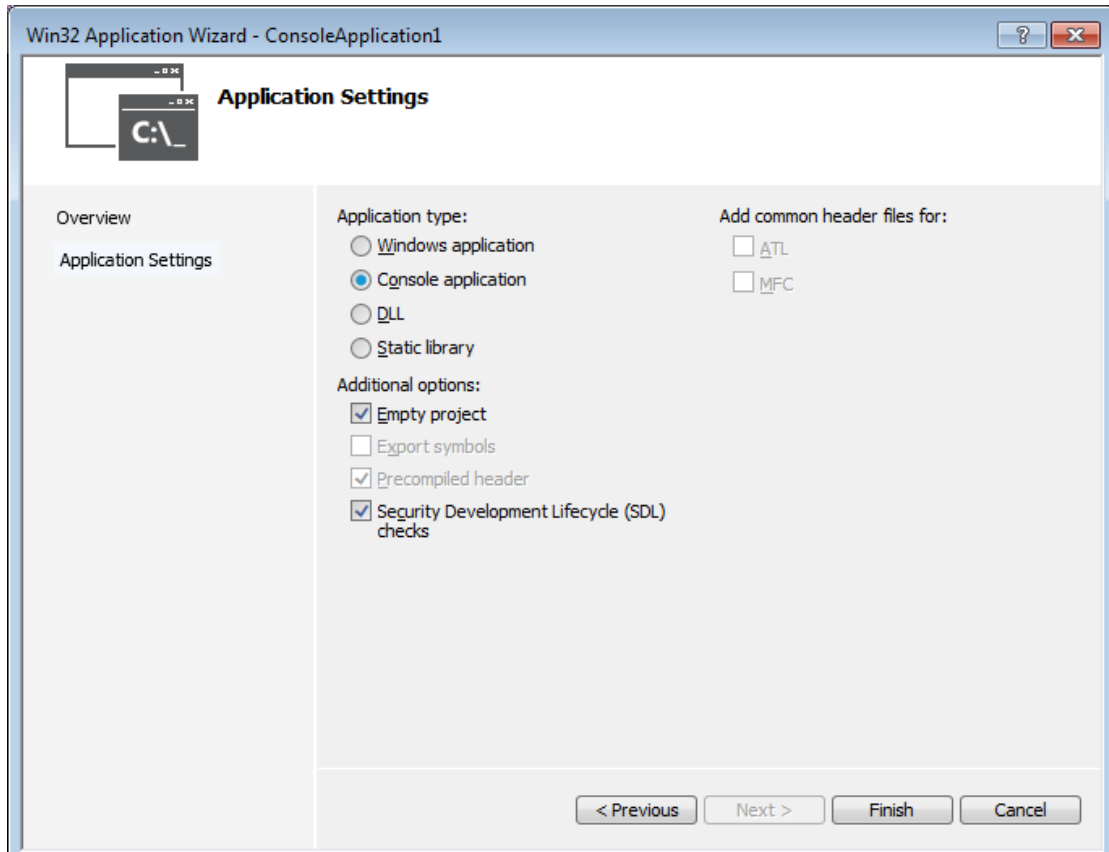
2. Select Start New project and select the Visual C++ - Template on Left the Hand Side, Win 32 Console Application on Right Hand Side, Name the Project – on the Bottom Center of the Dialog Box, next Click Ok – on the Bottom Right of the Dialog Box



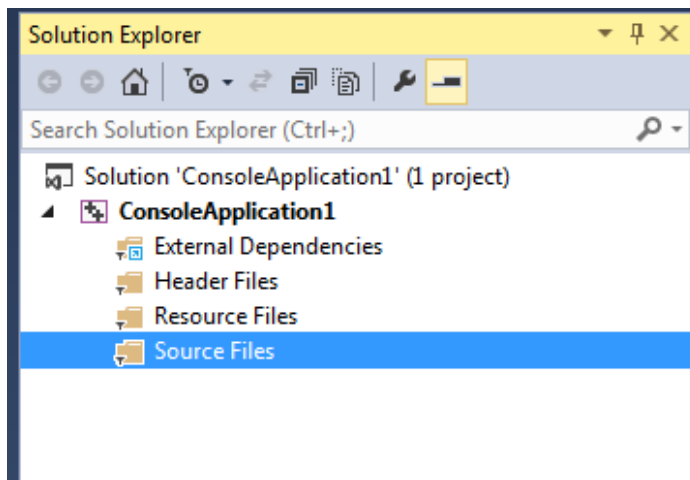
3. Use the Wizard Dialog Box and Click Application Settings on Left Side of the Dialog Box



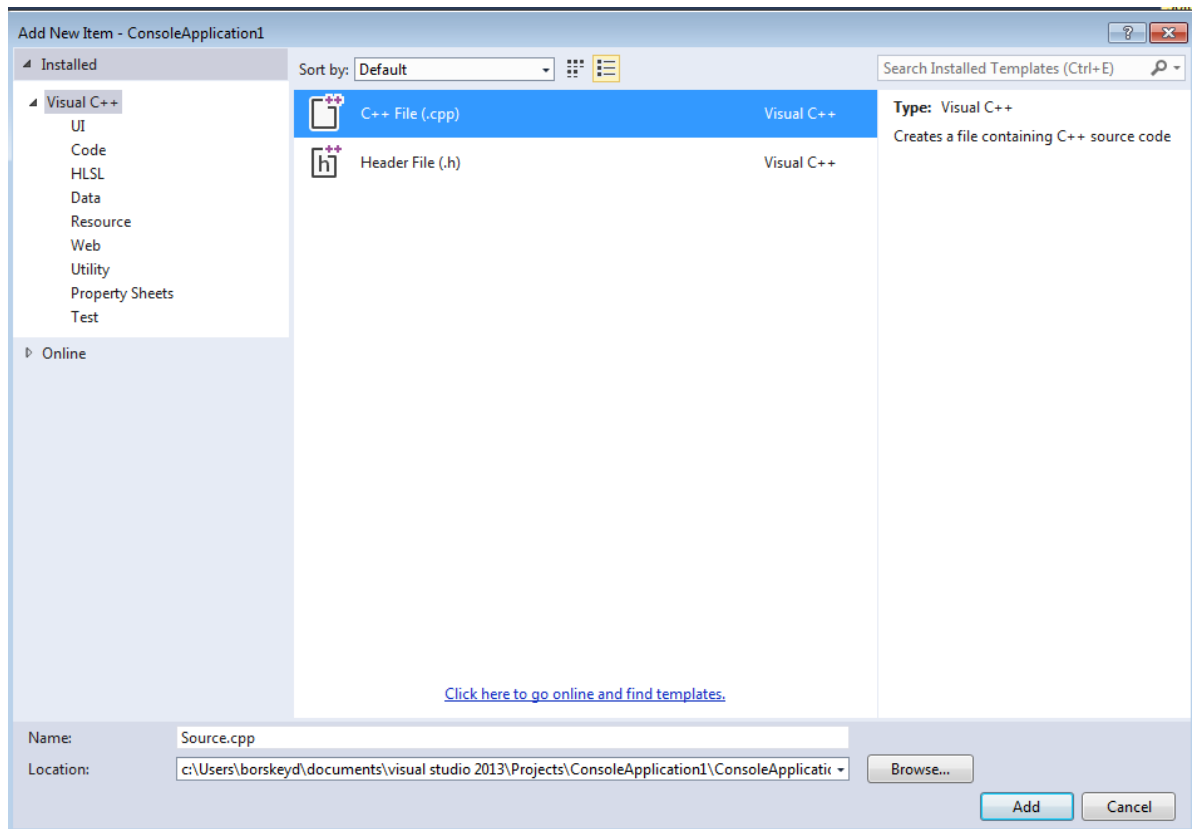
- Under Additional Options on the Center Section of the Dialog Box, select Empty Project and Click Finish.



- On the Right Hand Side Right Click on Source Files and Select Add New Item



6. On the Left Side Select Visual C++ and On the Right Side Select C++ File and you may use the same project name for the File on Bottom Center then Click Add on Bottom Right



7. Type in your C++ program
8. Select Build or F7 Check for Errors
9. Select Debug or CTRL F5 to Execute and See Results
10. Students may also use the debug menu and select start without debugging to have the screen pause automatically.

## **“All About Me”**

Using the examples provided below and your notes, provide the logic for a C++ program that declares the necessary variables needed and then prompts you the user to input your first and last name, birth city, state, one hobby and your age. The program next calculates the number of years before you may withdraw Social Security given 62 begins eligibility. The program also calculates how much money you will have invested in Social Security provided 600 dollars a year is saved from your current age to age 62. The program should use good input prompts and output all input data plus calculations. (Be sure to clearly indicate all data in the output). Remember to use the necessary C++ statements as practiced in class.

1. As discussed in class, students need to include the following lines before main in the C++ Program:

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

int main()
```

2. Using the code below, next type the main module of the program.

```
int main()
{
    string myName;
    string birthCityState;
    string myHobby;
    int myAge;
    int mySocialAge;
    int myInvest;

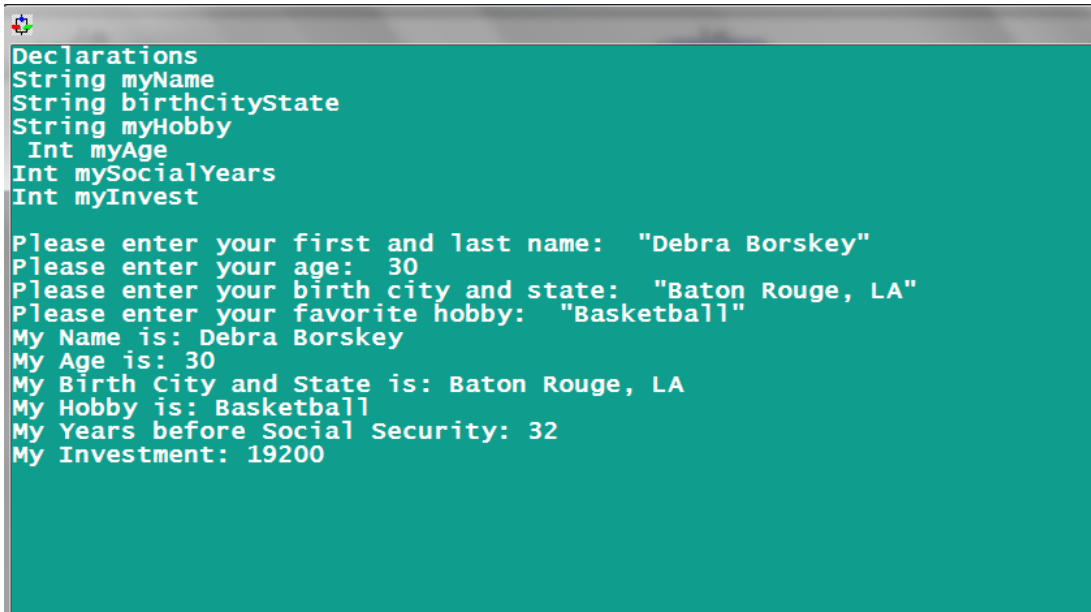
    cout<< "Please enter first and last name: ";
    getline(cin,myName);
    cout<< "\nPlease enter your birth city and state: ";
    getline(cin,birthCityState);
    cout<< "\nPlease enter your favorite hobby: ";
    getline(cin,myHobby);
    cout<< "\nPlease enter your age: ";
    cin>> myAge;

    mySocialAge = 62 - myAge;
    myInvest = myAge * 600;

    cout<< "\nMy full name: " << myName;
    cout<< "\nMy age: " << myAge;
    cout<< "\nMy birth city and state: " << birthCityState;
    cout<< "\nMy favorite hobby: " << myHobby;
    cout<< "\nMy age for Social Security: " << mySocialAge;
    cout<< "\nMy Investment: " << myInvest << endl;

    return 0;
}
```

3. Provide the visual logic flowchart output clipping (similar to the example below, use appropriate spacing) and pseudo code of the program typed below.

A screenshot of a terminal window with a dark green background and white text. The text shows the output of a C++ program. It starts with variable declarations, followed by prompts for user input and the corresponding output values.

```
Declarations
String myName
String birthCityState
String myHobby
  Int myAge
Int mySocialYears
Int myInvest

Please enter your first and last name: "Debra Borskey"
Please enter your age: 30
Please enter your birth city and state: "Baton Rouge, LA"
Please enter your favorite hobby: "Basketball"
My Name is: Debra Borskey
My Age is: 30
My Birth City and State is: Baton Rouge, LA
My Hobby is: Basketball
My Years before Social Security: 32
My Investment: 19200
```

4. Use Visual Studio 2013 to complete your C++ program. Feel free to use the steps provided to run, compile, debug and save your program. Format your work nicely. Remember, steps may vary.