


# Introduction to Computing


## Lecture 3

Dr. Naveed Anwar Bhatti

**Webpage:** [naveedanwarbhatti.github.io](http://naveedanwarbhatti.github.io)



# Introduction to C++ Programming





Machine  
Language

Assembly  
Language

High-Level  
Language



## Machine Language:

- Only language computer directly understands
- Defined by hardware design
  - Machine-dependent
- Generally consist of 0s and 1s (**Binary Language**)
- Instruct computers to perform elementary operations
  - One at a time
- Cumbersome for humans
- Example:  
10100111010  
01011101001



## Assembly Language:

- English-like abbreviations representing elementary computer operations
- Clearer to humans
- Incomprehensible to computers
  - Translator programs (assemblers)
    - Convert to machine language
- Example:

|              |                 |
|--------------|-----------------|
| <b>LOAD</b>  | <b>BASEPAY</b>  |
| <b>ADD</b>   | <b>OVERPAY</b>  |
| <b>STORE</b> | <b>GROSSPAY</b> |



## High-Level Language:

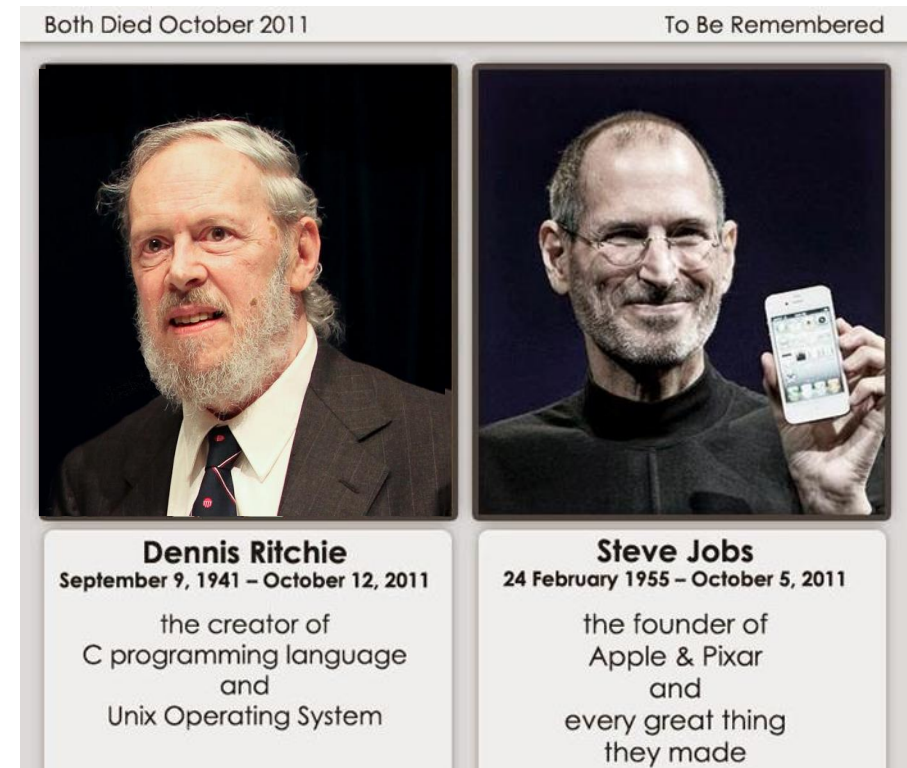
- Similar to everyday English, use common mathematical notations
- Single statements accomplish substantial tasks
  - Assembly language requires many instructions to accomplish simple tasks
- Translator programs (compilers)
  - Convert to machine language
- Example:

**grossPay = basePay + overTimePay**



# History of C and C++

- History of C
  - Evolved from other programming language 'B'
  - Dennis Ritchie (Bell Laboratories)
    - Added data typing, other features
  - Hardware independent
- History of C++
  - Extension of C
  - Early 1980s: Bjarne Stroustrup (Bell Laboratories)
  - Provides capabilities for object-oriented programming
    - Objects: reusable software components
      - Model items in real world
    - Object-oriented programs
      - Easy to understand, correct and modify





# C++ Standard Library

- C++ programs
  - Built from pieces called classes and functions
- C++ standard library
  - Rich collections of existing classes and functions





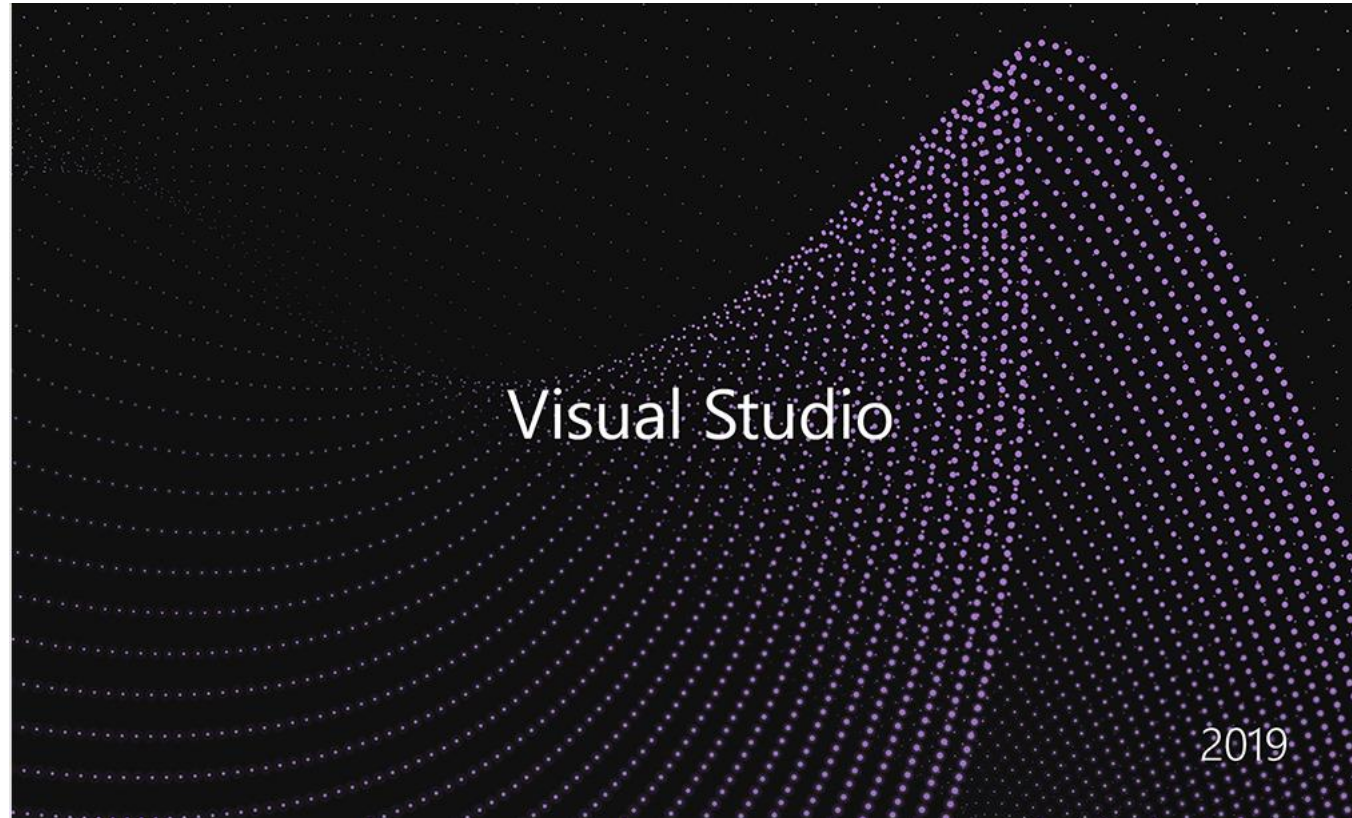
# Basics of a Typical C++ Environment

- C++ systems
  - IDE (Integrated Development Environment)
  - Language
  - C++ Standard Library
- C++ program extension
  - .cpp
  - .h (header file)

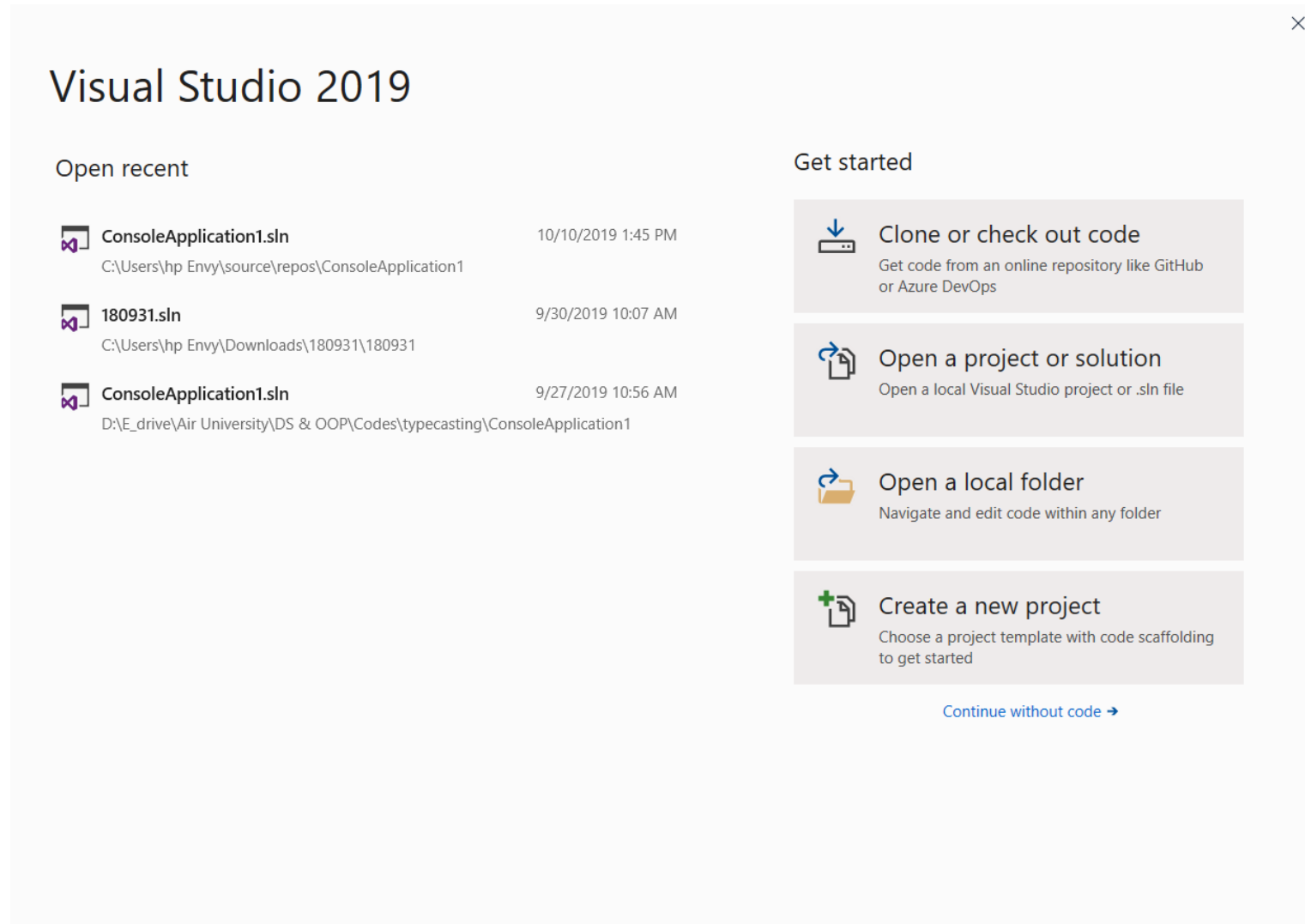


Web-based IDE's  
can work as well,  
but functionality  
is limited

# Basics of a Typical C++ Environment

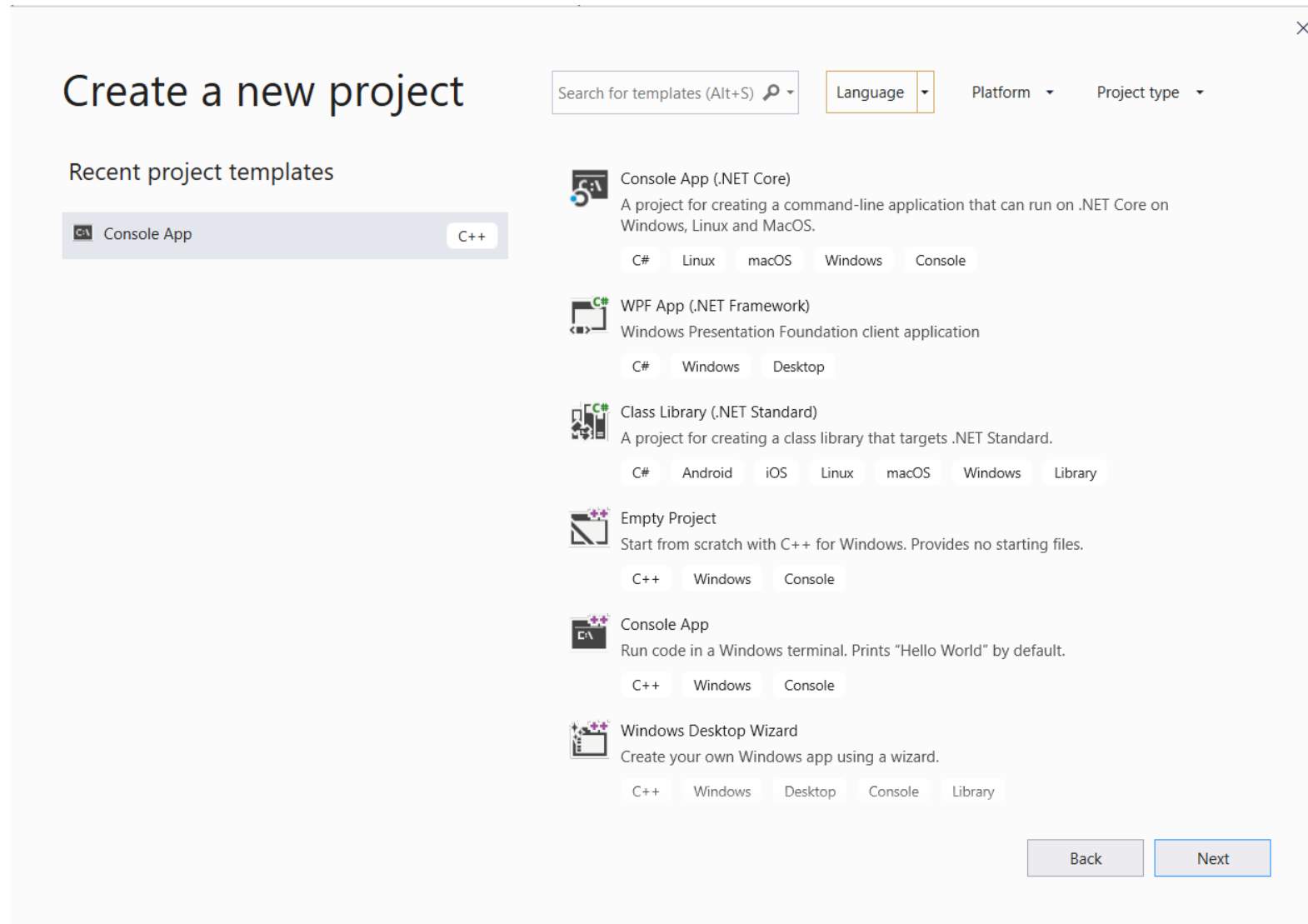


# Basics of a Typical C++ Environment





# Basics of a Typical C++ Environment






# Basics of a Typical C++ Environment

## Create a new project

Recent project templates


 Console App C++


Search for templates (Alt+S) 🔍


Language ▼  
All Languages  
C#  
**C++**  
F#  
Java  
JavaScript  
Python  
Query Language  
TypeScript  
Visual Basic


Platform ▼


Project type ▼


 Console App (.NET Core)  
A project for creating a console application that can run on .NET Core on Windows, Linux and macOS.  
C# Linux macOS

 WPF App (.NET Framework)  
Windows Presentation Foundation application.  
C# Windows Desktop

 Class Library (.NET Standard)  
A project for creating a class library that targets .NET Standard.  
C# Android iOS Linux macOS Windows Library

 Empty Project  
Start from scratch with C++ for Windows. Provides no starting files.  
C++ Windows Console

 Console App  
Run code in a Windows terminal. Prints "Hello World" by default.  
C++ Windows Console

 Windows Desktop Wizard  
Create your own Windows app using a wizard.  
C++ Windows Desktop Console Library

Back


Next



# Basics of a Typical C++ Environment

## Create a new project

Recent project templates

 Console App C++


Search for templates (Alt+S) 🔍


Language


Platform


Project type


Filtering by: C++ [Clear filter](#)


**Empty Project**  
Start from scratch with C++ for Windows. Provides no starting files.  
C++ Windows Console

**Console App**  
Run code in a Windows terminal. Prints "Hello World" by default.  
C++ Windows Console

**Windows Desktop Wizard**  
Create your own Windows app using a wizard.  
C++ Windows Desktop Console Library

**Windows Desktop Application**  
A project for an application with a graphical user interface that runs on Windows.  
C++ Windows Desktop

**Shared Items Project**  
A Shared Items project is used for sharing files between multiple projects.  
C++ Windows Android iOS Linux Desktop Console  
Library UWP Games Mobile

**Dynamic-Link Library (DLL)**

Back

Next



# Basics of a Typical C++ Environment

×

## Configure your new project

Empty Project C++ Windows Console

Project name

Location

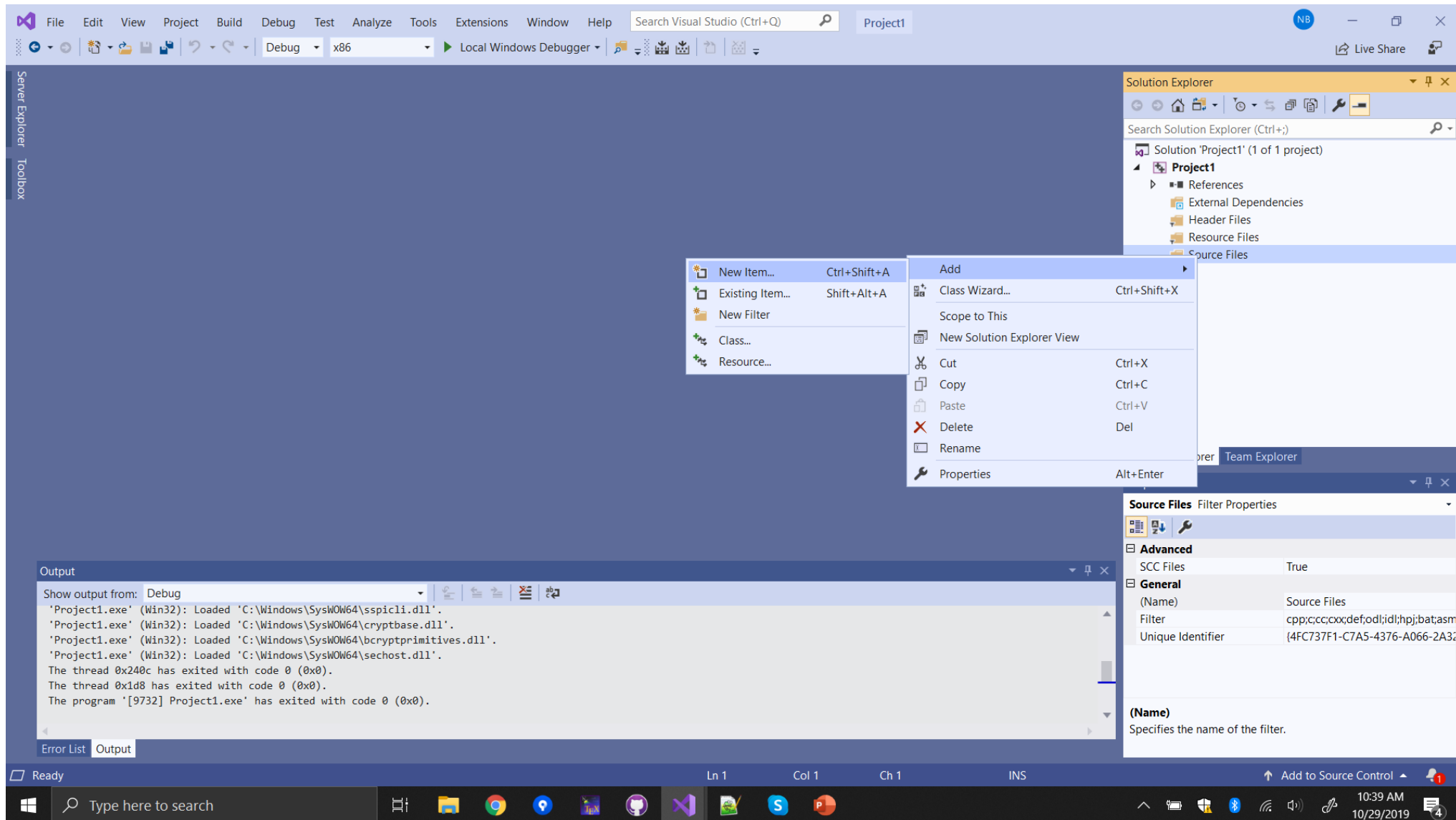
...

Solution name i

☐ Place solution and project in the same directory

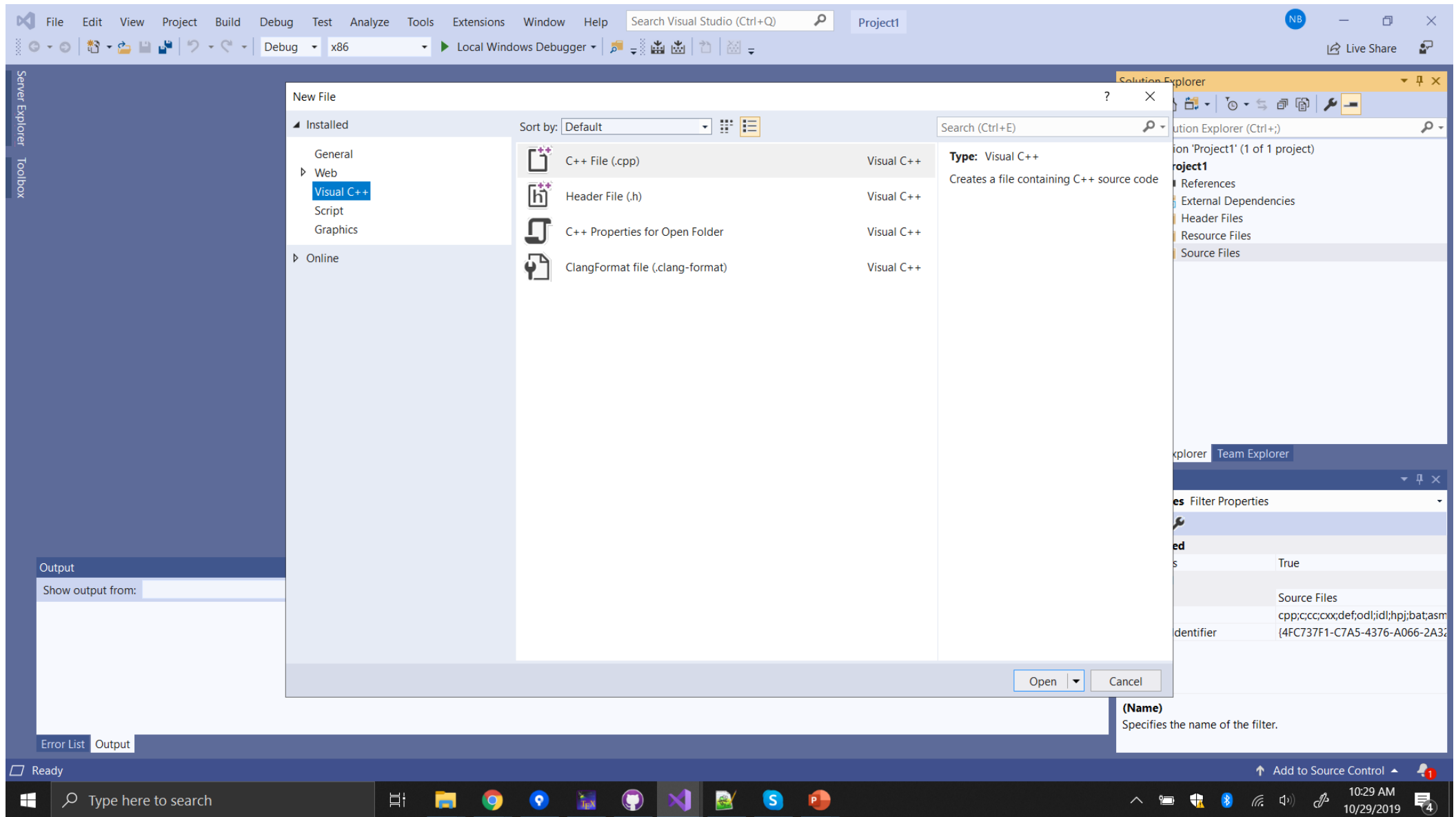
Back Create

# Basics of a Typical C++ Environment

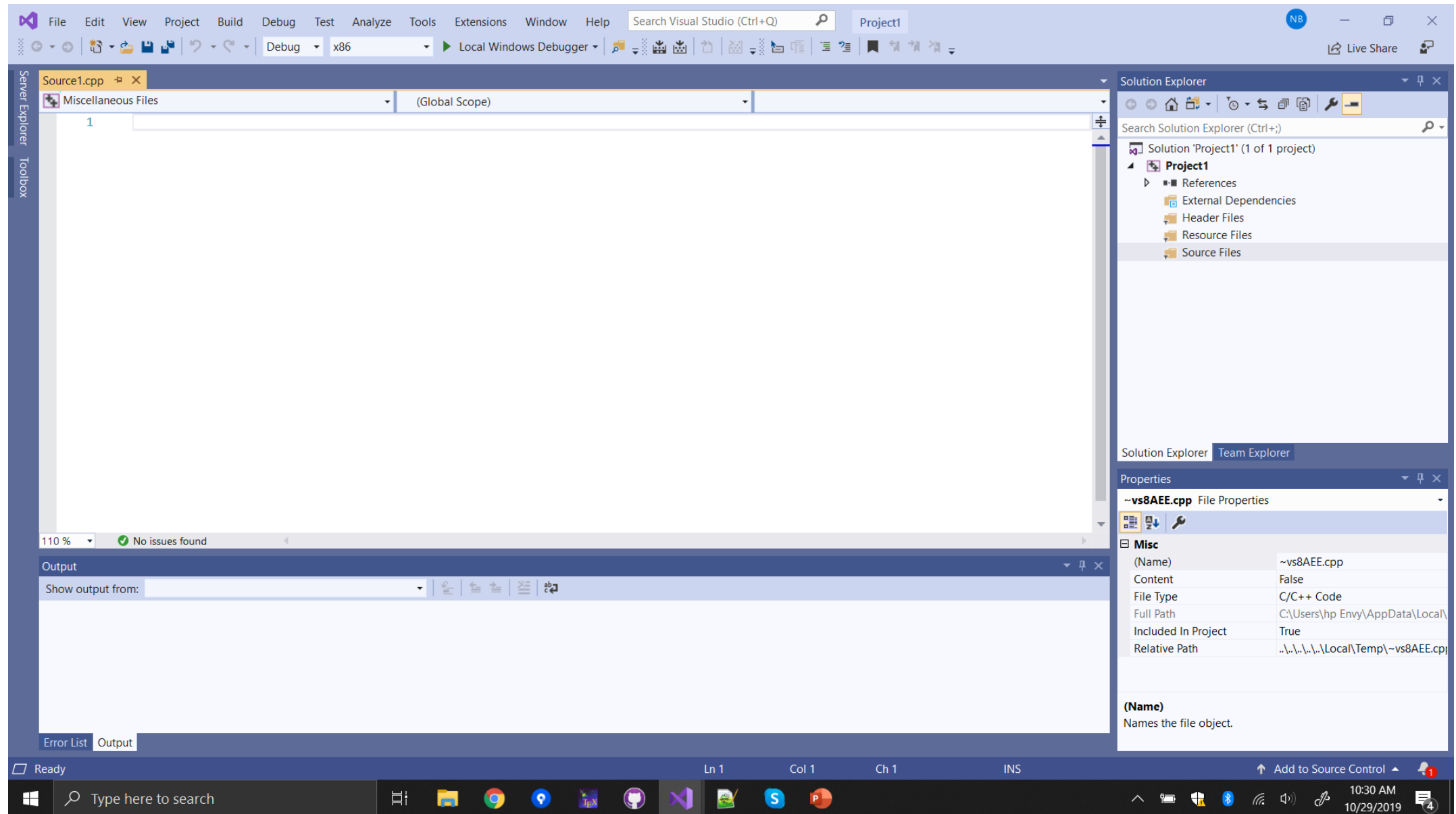




# Basics of a Typical C++ Environment



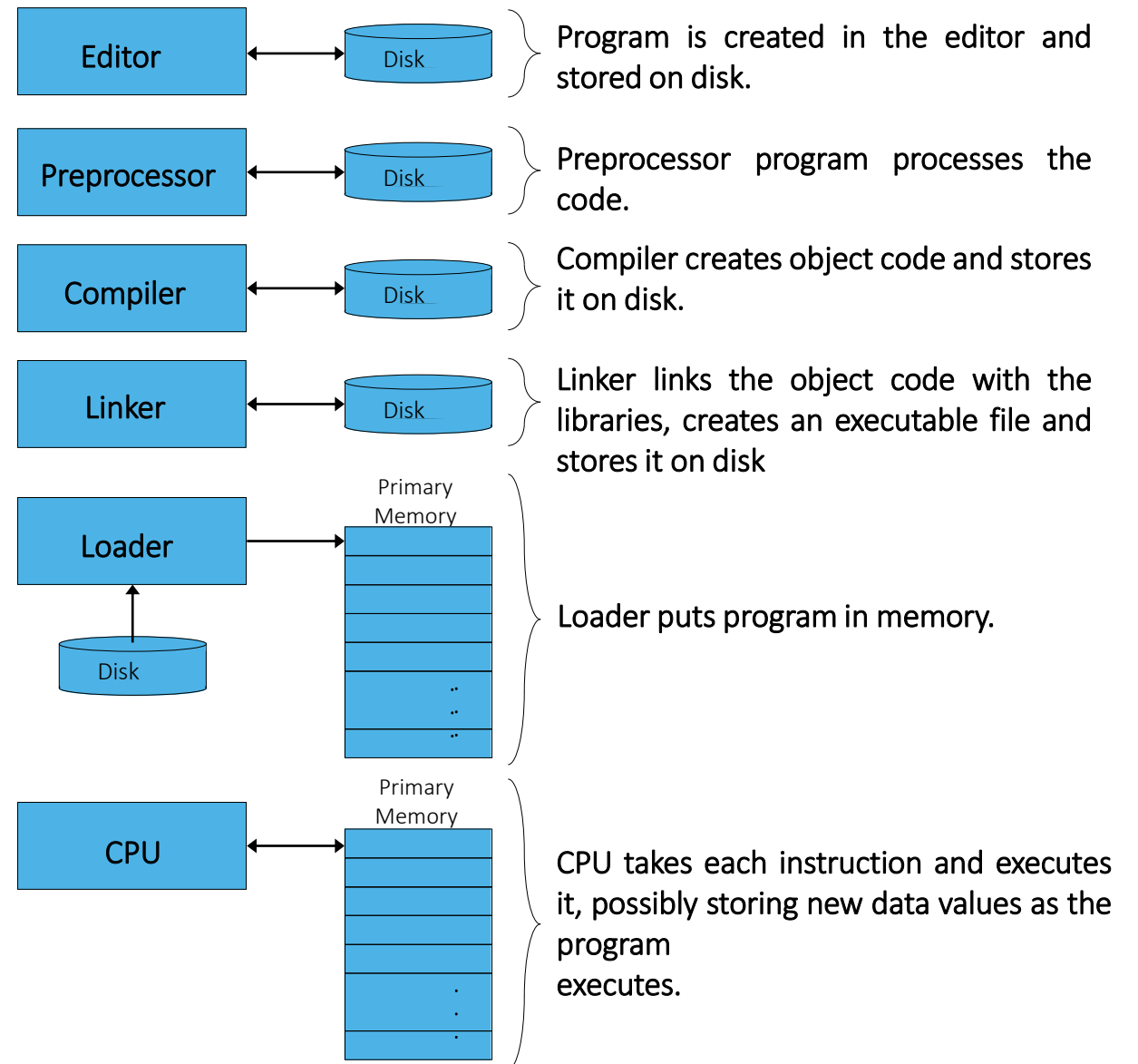
# Basics of a Typical C++ Environment



# Basics of a Typical C++ Environment

## Phases of C++ Programs:

1. Edit
2. Preprocess
3. Compile
4. Link
5. Load
6. Execute





# Basics of a Typical C++ Environment

- Common Input/output functions
  - `cin`
    - Standard input stream
    - Normally keyboard
  - `cout`
    - Standard output stream
    - Normally computer screen



# A Simple Program: Printing a Line of Text

- Before writing the programs
  - Comments
    - Document programs
    - Improve program readability
    - Ignored by compiler
    - Single-line comment
      - Use C's comment `/* .. */` OR Begin with `//` or
  - Preprocessor directives
    - Processed by preprocessor before compiling
    - Begin with `#`



# A Simple Program: Printing a Line of Text

```
1. // A first program in C++.
2. #include <iostream>
3. using namespace std;
4. // function main begins program execution
5. int main()
6. {
7.     cout << "Welcome to C++!\n";
8.
9.     return 0; // indicate that program ended successfully
10.
11. } // end function main
```

Single-line comments.

```
Welcome to C++!
```



# A Simple Program: Printing a Line of Text

```
1. // A first program in C++.
2. #include <iostream>
3. using namespace std;
4. // function main begins program execution
5. int main()
6. {
7.     cout << "Welcome to C++!\n";
8.
9.     return 0; // indicate that program ended successfully
10.
11. } // end function main
```

Preprocessor directive to include input/output stream header file **<iostream>**.

```
Welcome to C++!
```



# A Simple Program: Printing a Line of Text

```
1. // A first program in C++.
2. #include <iostream>
3. using namespace std;
4. // function main begins program execution
5. int main()
6. {
7.     cout << "Welcome to C++!\n";
8.
9.     return 0; // indicate that program ended successfully
10.
11. } // end function main
```

**using namespace std**  
means that we can use names  
for objects and variables from  
the *standard library*

```
Welcome to C++!
```





# A Simple Program: Printing a Line of Text

```
1. // A first program in C++.
2. #include <iostream>
3. using namespace std;
4. // function main begins program execution
5. int main()
6. {
7.     cout << "Welcome to C++!\n";
8.
9.     return 0; // indicate that program ended successfully
10.
11. } // end function main
```

Function **main** returns an integer value

```
Welcome to C++!
```



# A Simple Program: Printing a Line of Text

```
1. // A first program in C++.
2. #include <iostream>
3. using namespace std;
4. // function main begins program
5. int main()
6. {
7.     cout << "Welcome to C++!\n";
8.
9.     return 0; // indicate that program ended successfully
10.
11. } // end function main
```

Function **main** appears exactly once in every C++ program

```
Welcome to C++!
```



# A Simple Program: Printing a Line of Text

```
1. // A first program in C++.
2. #include <iostream>
3. using namespace std;
4. // function main begins
5. int main()
6. {
7.     cout << "Welcome to C++!\n";
8.
9.     return 0; // indicate that program ended successfully
10.
11. } // end function main
```

Left brace { begins function body

```
Welcome to C++!
```



# A Simple Program: Printing a Line of Text

```
1. // A first program in C++.
2. #include <iostream>
3. using namespace std;
4. // function main begins program execution
5. int main()
6. {
7.     cout << "Welcome to C++!\n";
8.
9.     return 0; // indicate th
10.
11. } // end function main
```

Standard output stream

```
Welcome to C++!
```



# A Simple Program: Printing a Line of Text

```
1. // A first program in C++.
2. #include <iostream>
3. using namespace std;
4. // function main begins program execution
5. int main()
6. {
7.     cout << "Welcome to C++!\n";
8.
9.     return 0; // indicate that program has ended successfully
10.
11. } // end function main
```

Stream insertion operator

```
Welcome to C++!
```



# A Simple Program: Printing a Line of Text

```
1. // A first program in C++.
2. #include <iostream>
3. using namespace std;
4. // function main begins program execution
5. int main()
6. {
7.     cout << "Welcome to C++!\n";
8.
9.     return 0; // indicate that program ended successfully
10.
11. } // end function main
```

Special character "\" is used. "n" means new line

Another way to insert a new line, is with the **endl** manipulator

```
Welcome to C++!
```



# A Simple Program: Printing a Line of Text

```
1. // A first program in C++.
2. #include <iostream>
3. using namespace std;
4. // function main begins program execution
5. int main()
6. {
7.     cout << "Welcome to C++!\n";
8.
9.     return 0; // indicate that program ended successfully
10.
11. } // end function main
```

Statements end with a  
semicolon ;

```
Welcome to C++!
```



# A Simple Program: Printing a Line of Text

```
1. // A first program in C++.
2. #include <iostream>
3. using namespace std;
4. // function main begins program execution
5. int main()
6. {
7.     cout << "Welcome to C++!\n";
8.
9.     return 0; // indicate that program ended successfully
10.
11. } // end function main
```

Keyword **return** is one of several means to exit function; value **0** indicates program terminated successfully

Welcome to C++!





# A Simple Program: Printing a Line of Text

- Standard output stream object
  - “Connected” to screen
  - `<<`
    - Stream insertion operator
    - Value to right (right operand) inserted into output stream
- Escape characters
  - `\`
  - Indicates “special” character output



## A Simple Program: Printing a Line of Text

| Escape Sequence | Description  |
|-----------------|--|
| <code>\n</code> | Newline. Position the screen cursor to the beginning of the next line. |
| <code>\t</code> | Horizontal tab. Move the screen cursor to the next tab stop.           |
| <code>\\</code> | Backslash. Used to print a backslash character.                        |
| <code>\"</code> | Double quote. Used to print a double quote character.                  |



# Basics of a Typical C++ Environment

The screenshot displays the Visual Studio IDE interface. The main editor window shows a C++ source file named `Source1.cpp` with the following code:

```
1 // A first program in C++.
2 #include <iostream>
3
4 // function main begins program execution
5 int main()
6 {
7     std::cout << "Welcome to C++!\n";
8
9     return 0; // indicate that program ended successfully
10
11 } // end function main
12
```

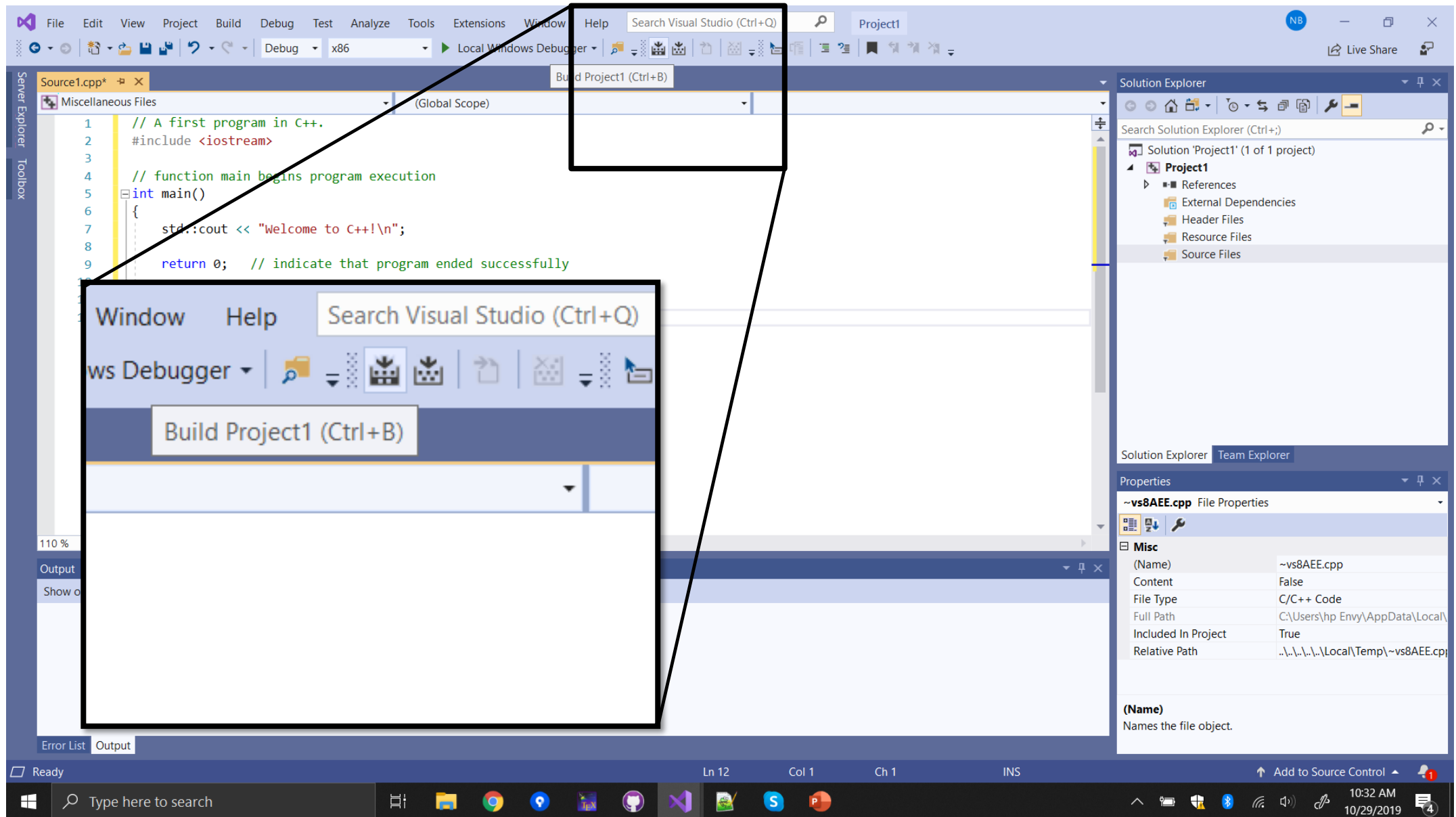
The Solution Explorer on the right shows a solution named 'Project1' containing a single project named 'Project1'. The project structure includes References, External Dependencies, Header Files, Resource Files, and Source Files.

The Properties window on the right shows the file properties for `~vs8AEE.cpp`. The properties are as follows:

| ~vs8AEE.cpp File Properties |                                   |
|-----------------------------|-----------------------------------|
| <b>Misc</b>                 |                                   |
| (Name)                      | ~vs8AEE.cpp                       |
| Content                     | False                             |
| File Type                   | C/C++ Code                        |
| Full Path                   | C:\Users\hp Envy\AppData\Local\   |
| Included In Project         | True                              |
| Relative Path               | ..\..\..\..\Local\Temp\~vs8AEE.cp |

The bottom status bar shows the current line is 12, column is 1, and the character set is INS. The system tray at the bottom indicates the time is 10:31 AM on 10/29/2019.

# Basics of a Typical C++ Environment



# Basics of a Typical C++ Environment

The screenshot displays the Visual Studio IDE interface for a C++ project named "Project1".

**Source File (Source.cpp):**

```
1 // A first program in C++.
2 #include <iostream>
3
4 // function main begins program execution
5 int main()
6 {
7     std::cout << "Welcome to C++!\n";
8
9     return 0; // indicate that program ended successfully
10
11 } // end function main
12
```

**Solution Explorer:** Shows the project structure for "Solution 'Project1' (1 of 1 project)". The project "Project1" contains the following folders and files:

- References
- External Dependencies
- Header Files
- Resource Files
- Source Files
  - Source.cpp

**Output Window:** Shows the build process results.

```
Show output from: Build
1>----- Build started: Project: Project1, Configuration: Debug Win32 -----
1>Source.cpp
1>Project1.vcxproj -> C:\Users\hp\Envy\source\repos\Project1\Debug\Project1.exe
----- Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped -----
```

**Status Bar:** Indicates "Build succeeded".



# Basics of a Typical C++ Environment

The screenshot displays the Visual Studio IDE interface for a C++ project named "Project1". The main editor window shows the source code for "Source.cpp", which includes a C++ program structure with a main function and a "Welcome to C++!" message. The code is as follows:

```
1 // A first program in C++.
2 #include <iostream>
3
4 // function main: basic program execution
5 int main()
6 {
7     std::cout << "Welcome to C++!\n";
8     return 0;
9 }
10
11 // end of program
12
```

The "Microsoft Visual Studio Debug Console" window is open, showing the output of the program execution:

```
C:\Users\hp Envy\source\repos\Project1\Debug\Project1.exe (process 22752) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

The "Solution Explorer" on the right side of the IDE shows the project structure, including "References", "External Dependencies", "Header Files", "Resource Files", "Source Files", and "Source.cpp". The "Properties" window is also visible at the bottom right.

The "Output" window at the bottom left shows the output of the program execution, including the "Welcome to C++!" message and the exit code 0 (0x0).

The Windows taskbar at the bottom shows the system clock as 10:42 AM on 10/29/2019, along with various application icons and a search bar.

Thanks a lot



If you are taking a Nap, **wake up**.....Lecture Over