



COMP2012H Honors Object-Oriented Programming and Data Structures

Topic 1: Introduction

Dr. Desmond Tsoi

Department of Computer Science & Engineering
The Hong Kong University of Science and Technology
Hong Kong SAR, China



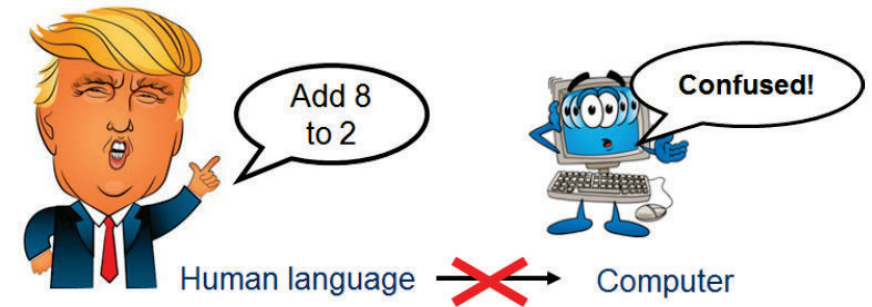
Rm 3553, desmond@ust.hk

COMP2012H (Fall 2021)

1 / 28

Computer Programming

- Learning computer programming is just like learning a natural language such as English, Japanese, Korean, etc.
- Although they are similar, it doesn't mean they are exactly the same.



- Don't worry. Actually this is good. Since **computer programming** is **more systematic and should be much easier to learn**, in my opinion. ;) (Good news!)

Rm 3553, desmond@ust.hk

COMP2012H (Fall 2021)

2 / 28

Programming Languages

- Computer programs are written in programming languages.
- Different to those human languages, a programming language defines A SET OF INSTRUCTIONS in SPECIFIC FORMAT that can be given to a computer.
- Two important issues on writing programs:
 1. Program syntax – Is the grammar of the instructions correct?
 2. Program logic – Is the program able to solve the problem?



Rm 3553, desmond@ust.hk

COMP2012H (Fall 2021)

3 / 28

Machine Code: Can You Understand This?

```

0000100100101110011001100101001011011000110010100001001001000100110110001100101100011011101000111
010101111000100100101000110000100111001100011001000100000101001100111011000110110001101100011011111
01100010101111011010110101100000101100010110100011001010100000101110001101100000101000101100111
001010100101011000101110100010100101011101101110000010001000100010110011101000101001011110001
01110100001000100000101000001001001011011011000101101000101001011011011100010000000101000000
10100000100100101110011001110110110001011101101000100110000010110101100000101100010110001
0110111000001010000010010010111001110100011100101110000010010100000101101010000010110
10010101110001011000010001011001100111010101101100100010101101000101001010111010111000001010
00001001001011100111000001110010010101110100011000010010011000000101000101010101100010110
101010101110001101100000101000001001001000100100011010100000101001001111010011000100010111
001010100010010010001001000000010000000010000000101100110100010110110010010010001000000010
010101010011011100000001010000101010010100010011000001100000100100010010110110011011000000001010
000010100000100100100001001000101010000010100100011101001100010011101000110101010101001010010
0011001000000010001000010100000100101101101011101101101100010000000110001001011000010010101111
001010000000010000001001011001011010000100000000100101011100110000000001001011011000000101010
011001110000000101010010010001010000010110000010000000010010101011011011011000000010100010101010
0110011100000001010100100100010000001011000001000000001001010101010110110110110110001000000010010
0010100001001010101110011000000001010000010010111001011101000010000000100101010111001100000010
11000101011001001001100111000000010110011001000110100010110110000101000001000101011000110000
001000000010101100100101010010011100000010100100100001000000101101001010000100010101101110011
00000000010100000100101101000010000001010110010010101001010110000000101001010010001001000
0101101001011000010010101011100110010001000000101000001010000010100100010100100010010000
11110011000000101000010010101011100110001001011000010010101011100110000000010100000100101110011
0111010000010000000100101011100110000000101000101010010010101001001100010011000000010100101000011
10000101110100001010000010010101010110110110100010000001010000010100001001010101001001010000
0000101000001001011000100010000000101110010011000100100001010000010010101011001011011101
0000000001000010111000100100010110000100010001010000010010101011001011011101
00000100100101100010010011000010001000101000001010000010010101011001011011101
00000100100101100010010011000010001000101000001010000010010101011001011011000001010
0110010100001010100101011000001010000010100010100010010001001001001001000001001010101010
1101011000010101001010101100000101000001001001011001010001010010001001010110010110000001001
0010000100100011011000010100000100101100100000000100000100100101001010010100101000000011
00100001011100011100000101110001000100100001000001000001001001001001001001001000000010
00100001011100011100000101110001000100100001000001000001001001001001001001001000000010

```

Rm 3553, desmond@ust.hk

COMP2012H (Fall 2021)

4 / 28

Assembly Language: How About This?

```
main:
    !#PROLOGUE# 0
    save %sp,-128,%sp

    !#PROLOGUE# 1
    mov 1,%o0
    st %o0,[%fp-20]
    mov 2,%o0
    st %o0,[%fp-24]
    ld [%fp-20],%o0
    ld [%fp-24],%o1
    add %o0,%o1,%o0
    st %o0,[%fp-28]
    mov 0,%i0
    nop
```



High-Level Language: Is This Better Now?

```
int main( )
{
    int x, y, z;

    x = 1;
    y = 2;
    z = x+y;

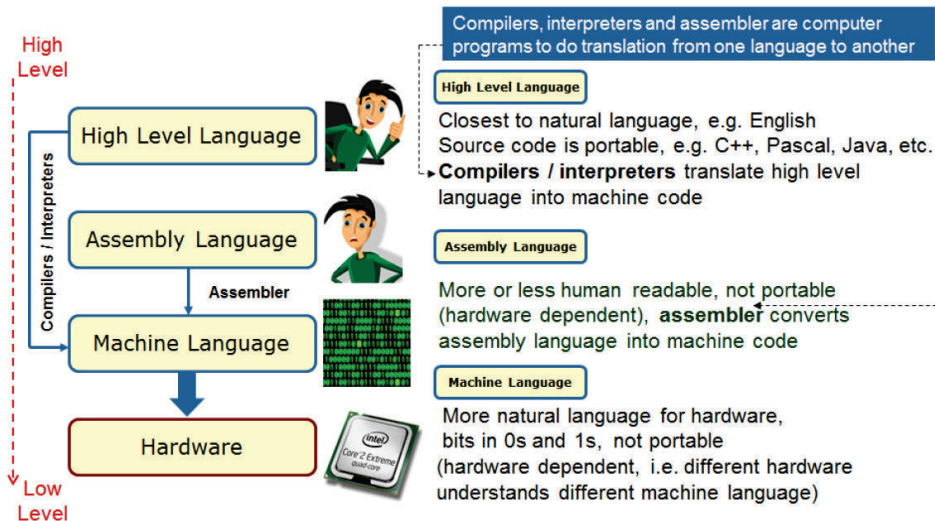
    return 0;
}
```



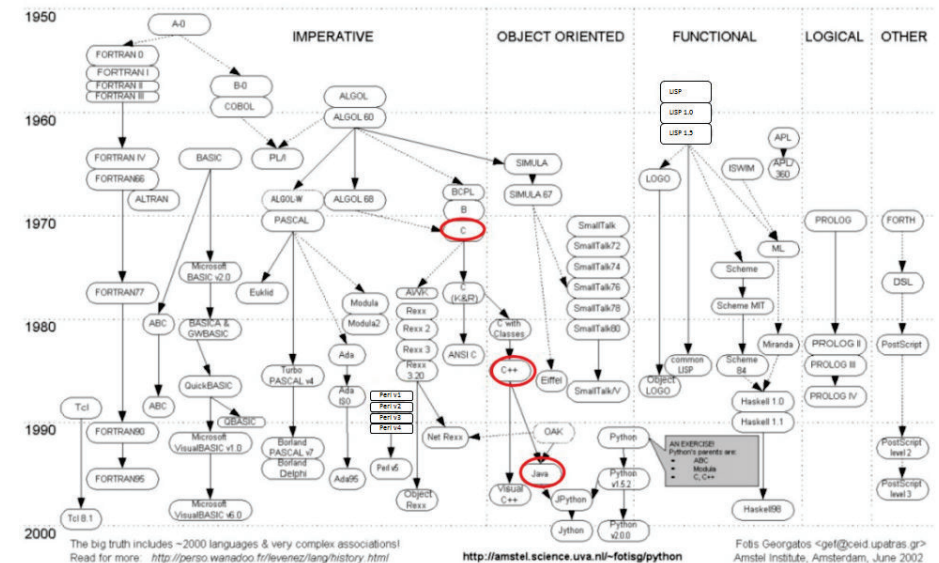
Write a Program to Sum 2 Numbers

- There are 3 integer-value-holding objects: x, y, and z.
- x and y have the value of 1 and 2 respectively.
- z's value is the sum of x's and y's.

Levels of Programming Languages



Chronology of Some High Level Programming Languages



Which Programming Language Are We Going to Use?

We are going to use C++ in this course!

• Why C++?

Read the [FAQ](#) from the designer of C++, [Bjarne Stroustrup](#).

• Which C++?

- ▶ The language has been **evolving**:
C++ 1983 \Rightarrow C++ 1998 \Rightarrow C++ 2003 \Rightarrow C++ 2011 \Rightarrow ...
- ▶ We will learn C++11 (but not all the new features).

• Which compiler?

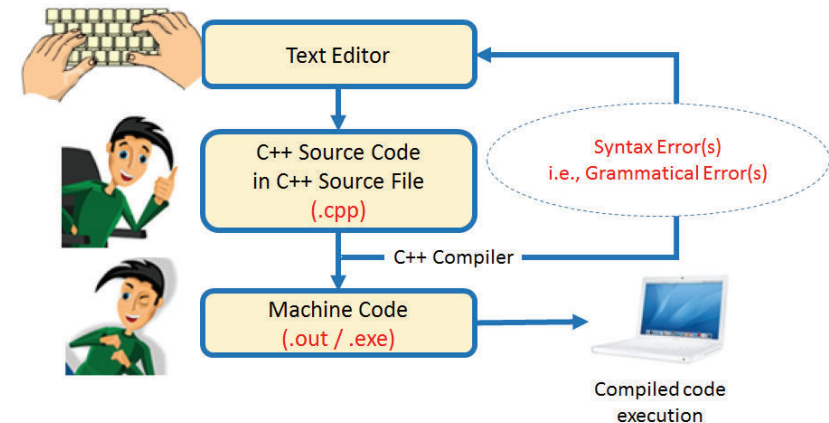
GNU gcc/g++. It is free.

(The compiler you will use is C++11-compliant.)

• Which IDE (integrated development environment) for writing programs?

VSCode. It is free and supported by many operating systems such as Windows, Mac OS, and Linux.

Development Cycle of a C++ Program



- A **compiler** translates **source programs** into **machine codes** that run directly on the target computer.
- For example, `a.cpp` \rightarrow `a.out` (or `a.exe`).
- Some C++ compilers: `gcc/g++`, `VC++`.

Example: Hello World!

```
/*
 * File: hello-world.cpp
 * A common program used to demo a new language
 */

#include <iostream>    // Load info of a Standard C++ library
using namespace std;  // Standard C++ namespace

int main()            // Program's entry point
{
    /* Major program codes */
    cout << "Hello World!" << endl;

    return 0;         // A nice ending
}
```



Write, Compile, and Run a Program in a Terminal

- STEP 1** : Write the program using an **editor**.
e.g., **VSCode**, **vi** (Unix/Linux), **MS Word** (Windows)
- STEP 2** : Save the program into a file called **hello-world.cpp**.
- STEP 3** : Compile the program using **g++** compiler.

```
g++ -o hello-world hello-world.cpp
```

If you don't specify the output filename using the **"-o"** option, the default is **a.out**.

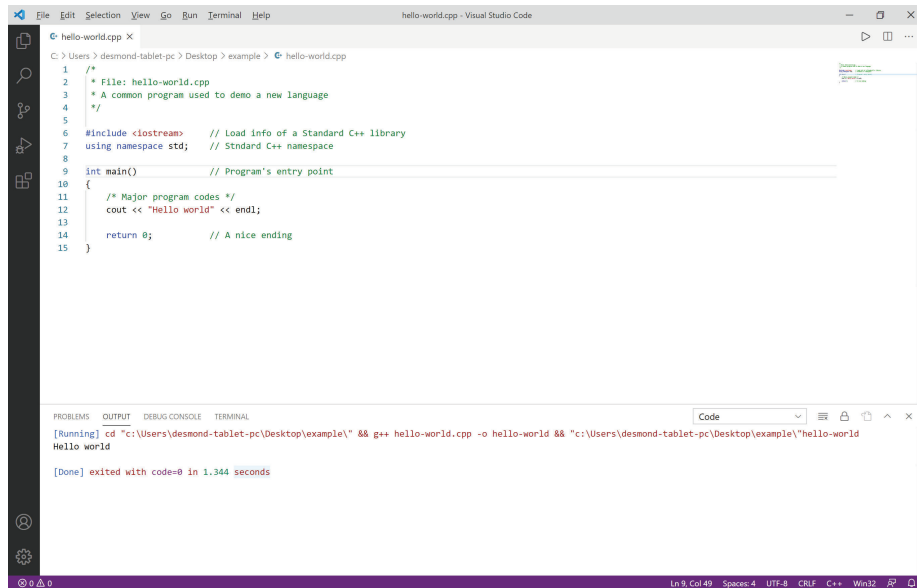
```
g++ hello-world.cpp
```

- STEP 4** : Run the program in a terminal (command window):

```
linux:: hello-world
Hello World!
```

VSCode IDE for C/C++

In the lab, you will use **VSCode** (similar to MS Visual Studio).



```
1 /*  
2  * File: hello-world.cpp  
3  * A common program used to demo a new language  
4  */  
5  
6 #include <iostream> // Load info of a Standard C++ library  
7 using namespace std; // Standard C++ namespace  
8  
9 int main() // Program's entry point  
10 {  
11     /* Major program codes */  
12     cout << "Hello world" << endl;  
13  
14     return 0; // A nice ending  
15 }
```

```
[Running] cd "C:\Users\desmond-tablet-pc\Desktop\example\" && g++ hello-world.cpp -o hello-world && "C:\Users\desmond-tablet-pc\Desktop\example\hello-world"  
Hello world  
[Done] exited with code=0 in 1.344 seconds
```

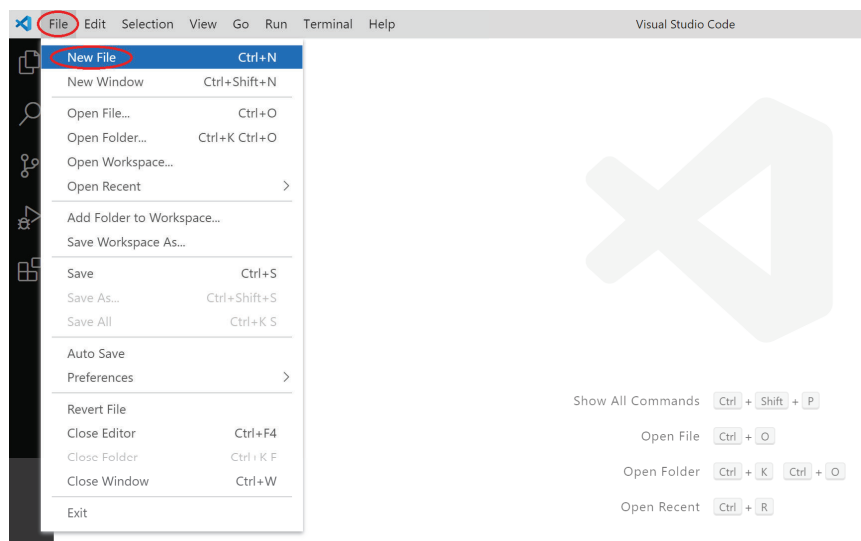
Program Development using VSCode



Step 1	Create a new file	Ctrl-N (or Click File → New File)
Step 2	Write program	VSCode built-in editor
Step 3	Save program	Ctrl-S (or Click File → Save)
Step 4	Compile and run program	F1 → “Run code”

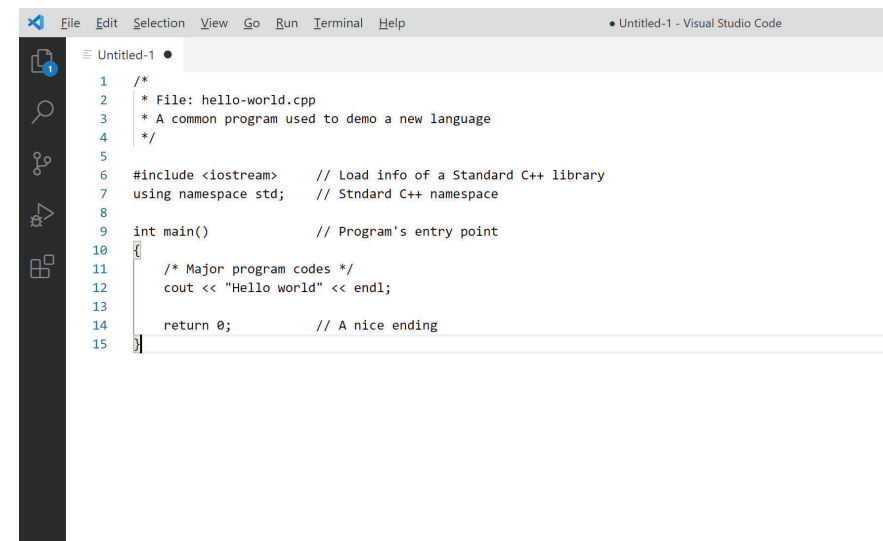
Step 1a: Create a new file

- Ctrl-N (or Click “File” → “New File”)



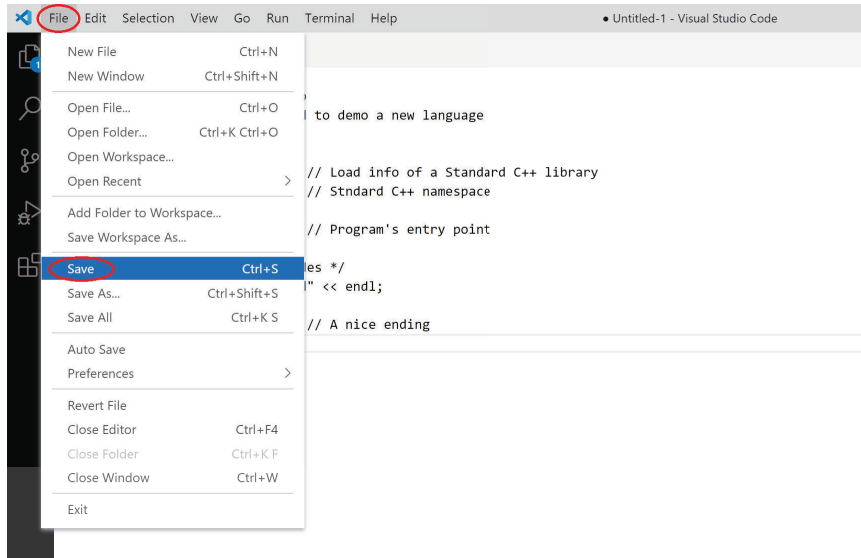
Step 2: Write program

- Write your program



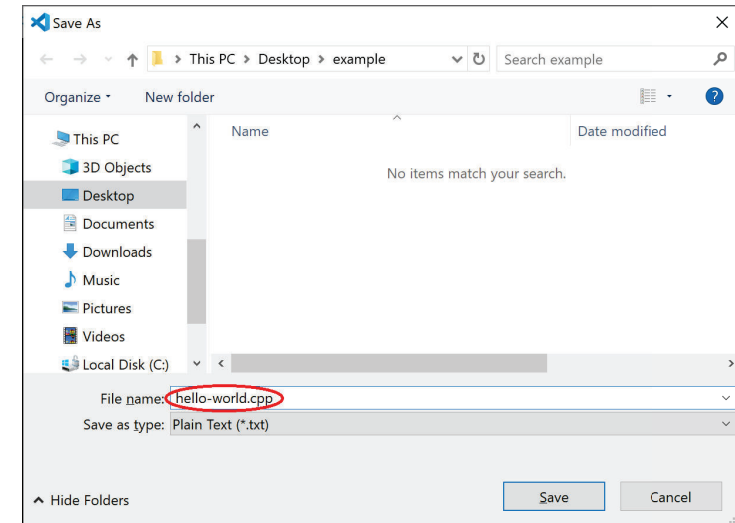
Step 3a: Save program

- Ctrl-S (or File → Save)



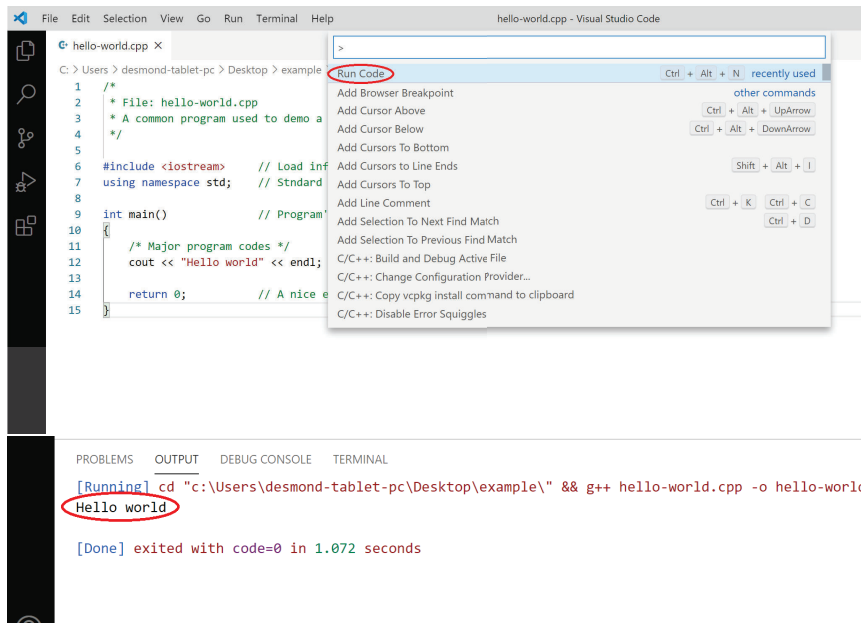
Step 3b: Save program

- Choose a location, give it a proper name, and click “Save”.
Note: make sure the name is something which ends with .cpp to indicate that is a C++ source file (e.g., hello-world.cpp).



Step 4: Compile and run program

- F1 → click “Run code”



Main: the Entry Point

- Every program must have exactly one and only one `main()` function.

Simple Form of the `main` Function

```
int main () { ... }
```

General Form of the `main` Function

```
int main (int argc, char** argv) { ... }
```

(We'll talk about `argc` and `argv` later.)

- Between the braces “{” and “}” are the program codes consisting of zero or more program **statements**.
- Each simple C++ statement ends in a semicolon “;”.

C++ Comments

- Use `/* ... */` for multiple-line comments.

```
/*  
 * A common program used to demo a new language  
 */
```

- Single-line comments start with `//`.

```
// Program's entry point
```

- Comments are just for human to read.
- They will not be translated by the compiler into machine codes.

#include and Standard C++ Libraries

- `#include` will include information of a **library** — a collection of sub-programs. e.g. `#include <iostream>` gets the information of the **standard C++ library** called **iostream** that deals with I/O:
 - ▶ **cin**: an object to read, e.g., from the keyboard or file
 - ▶ **cout**: an object to print out, e.g., to the screen or file
 - ▶ **cerr**: an object to print error message, e.g., to the screen or file

Examples

```
// endl means "end of a line"  
cout << "Einstein: God does not play dice." << endl;  
  
// You may also break down the message in several lines  
cerr << "Error: "  
      << "There is no stress and tension in HKUST!"  
      << endl;
```

- These library information files are called **header files**.

#include and User-defined Libraries

- You may also define your **own** library.
- Again you need to use `#include` to include its information into your sub-programs.
- Example: `#include "drawing.h"` gets the information of a **user-defined C++ library** about drawing.
- By convention, the **header file** of a **user-defined library** ends in `".h"` or `".hpp"`, while **Standard C++ library** header files have **no** file suffix.
- Also by convention, the **header file** of a **user-defined library** is delimited using double-quotes `"..."`, while **Standard C++ library** header files use `< ... >`.

C++ is a Free Format Language

- Extra **blanks**, **tabs**, **lines** are **ignored**.
- Thus, codes may be indented in any way to enhance **readability**.
- More than one statement can be on one line.
- Here is the same Hello World program:

```
#include <iostream> /* File: hello-world-too.cpp */  
using namespace std; int main (int argc,  
    char** argv) { cout<<"Hello World!"<<endl;return 0;}
```

- On the other hand, a single statement may be spread over several lines.

```
cout << "Hello World!"  
      << endl;
```

Good Programming Style

- Place **each** statement on a line by itself.
- For **long** statements
 - ▶ if possible, break it down into several shorter statements.
 - ▶ wrap it around with proper indentation (since extra space doesn't matter!)
- Use blank lines to **separate** sections of related codes that together perform some action.
- **Indent consistently**. Use the same indentation for the same block of codes.



Programming as Problem Solving

- **Understand** and **define** the problem clearly.
 - ▶ What are the input(s) and output(s)?
 - ▶ Any constraints?
 - ▶ Which information is essential?
- **Develop** a solution.
 - ▶ Construct an algorithm.
- **Translate** the algorithm into a C++ program.
- **Compile** the program.
- **Test** the program.
- **Debug** the program.
- **Document** the program as you write the program.
- **Maintain** the program
 - ▶ modify the codes when conditions change.
 - ▶ enhance the codes to improve the solution.



What Makes a Good Program?

- **Correctness**
 - ▶ Meets the problem requirements
 - ▶ Produces correct results
- **Easy to read and understand**
- **Easy to modify**
- **Easy to debug**
- **Efficient**
 - ▶ Fast
 - ▶ Requires less memory



That's all!

Any questions?

