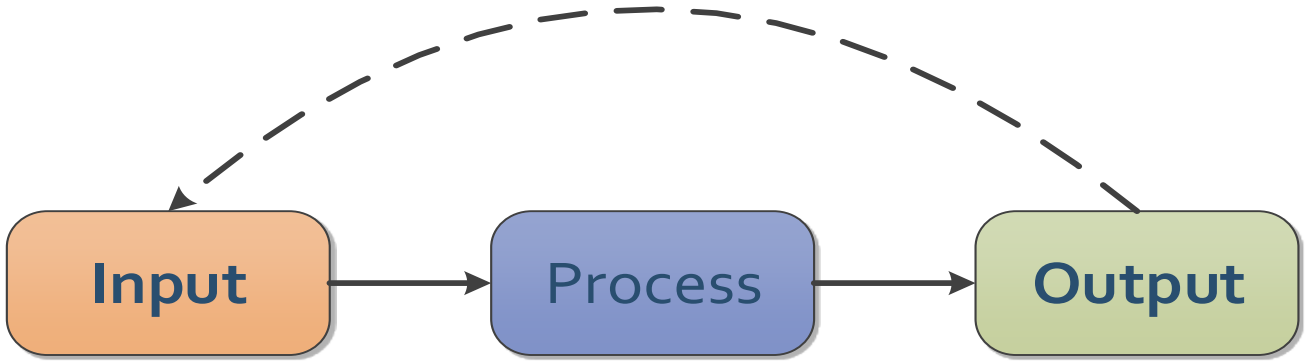
CS2311 Computer Programming

LT01: Basic Concept of Programming

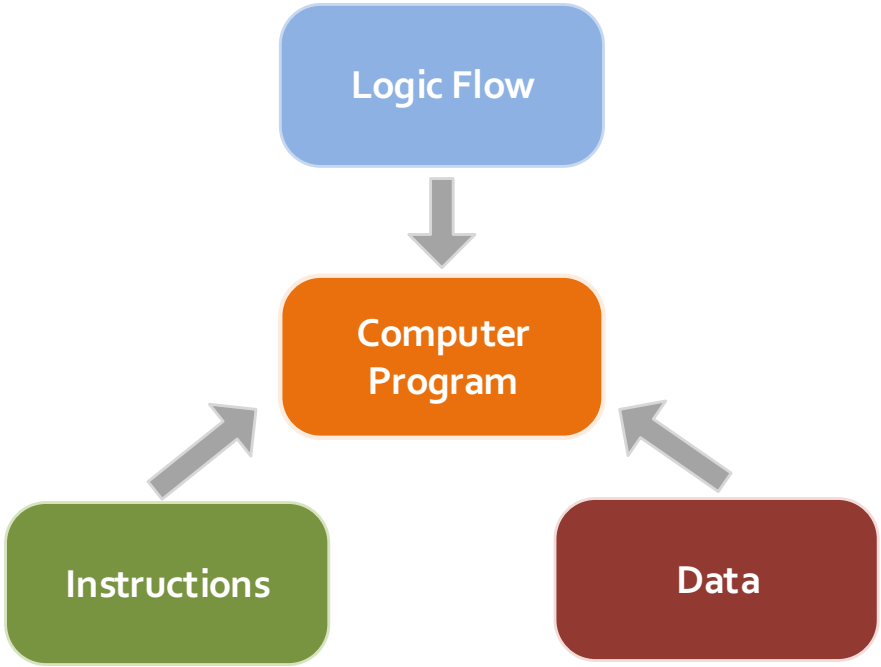
Computer Program (External View)

- Basic elements of a Computer Program
  - Input
  - Process
  - Output



Computer Program (Internal View)

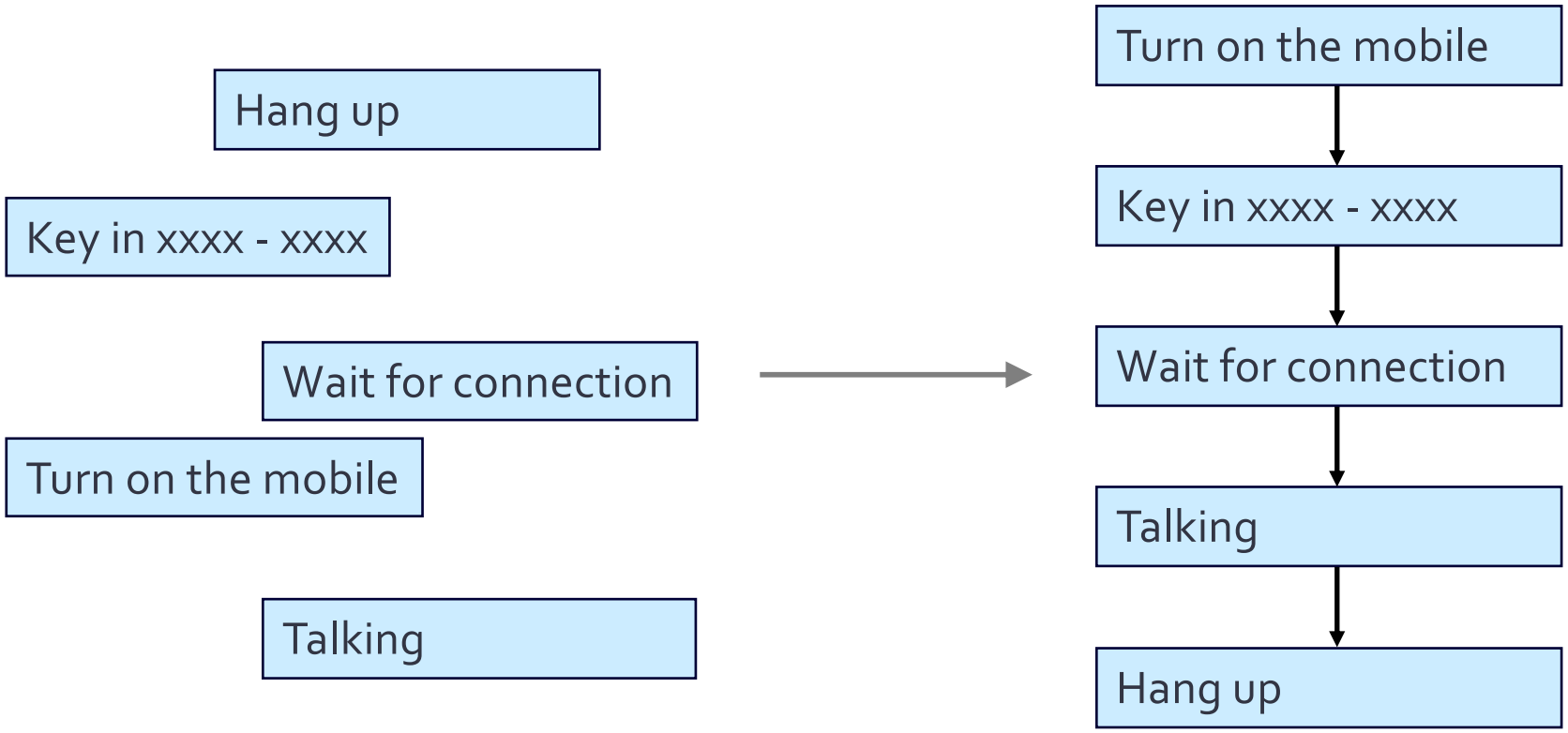
- A list of instructions ordered logically
- Usually involve data access



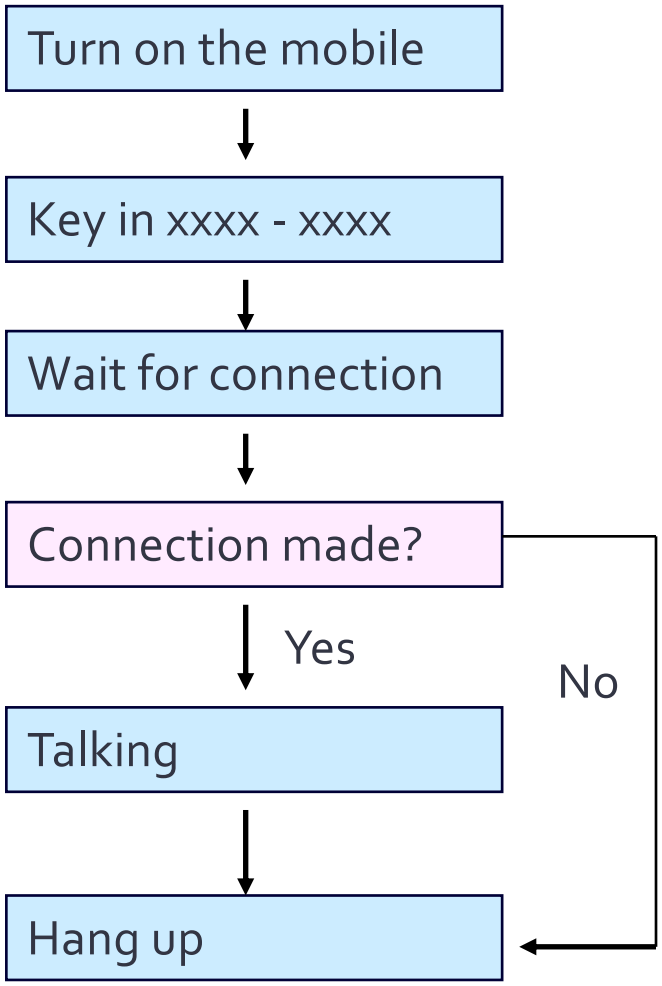
Computer Program

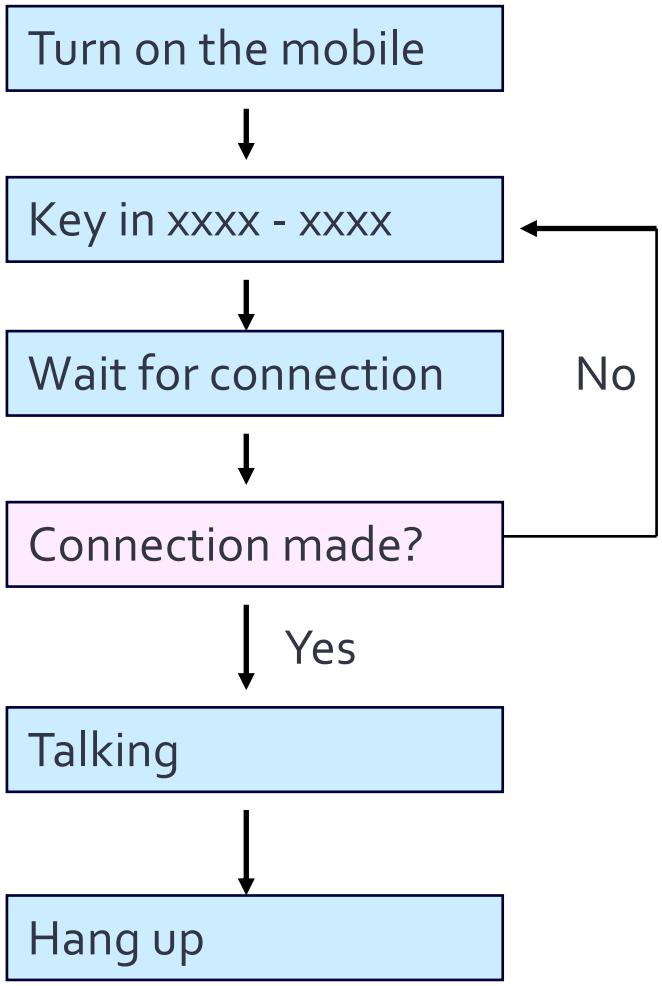
- Instructions
  - A set of predefined actions that a computer can perform
  - E.g. addition, subtraction, read , write
- Logic Flow
  - Arrangement of Instructions
  - E.g. Calculate BMI (Body Mass Index)
    - Read weight from keyboard
    - Read height from keyboard
    - Weight x weight/height
    - Write BMI to screen
- Data
  - Variable (data)
    - A space for temporarily store value for future process
  - Constant (data)
    - A value that will not be changed for the whole processing

Logic Flow


Hang up
Key in xxxx - xxxx
Wait for connection
Turn on the mobile
Talking
Turn on the mobile
Key in xxxx - xxxx
Wait for connection
Talking
Hang up

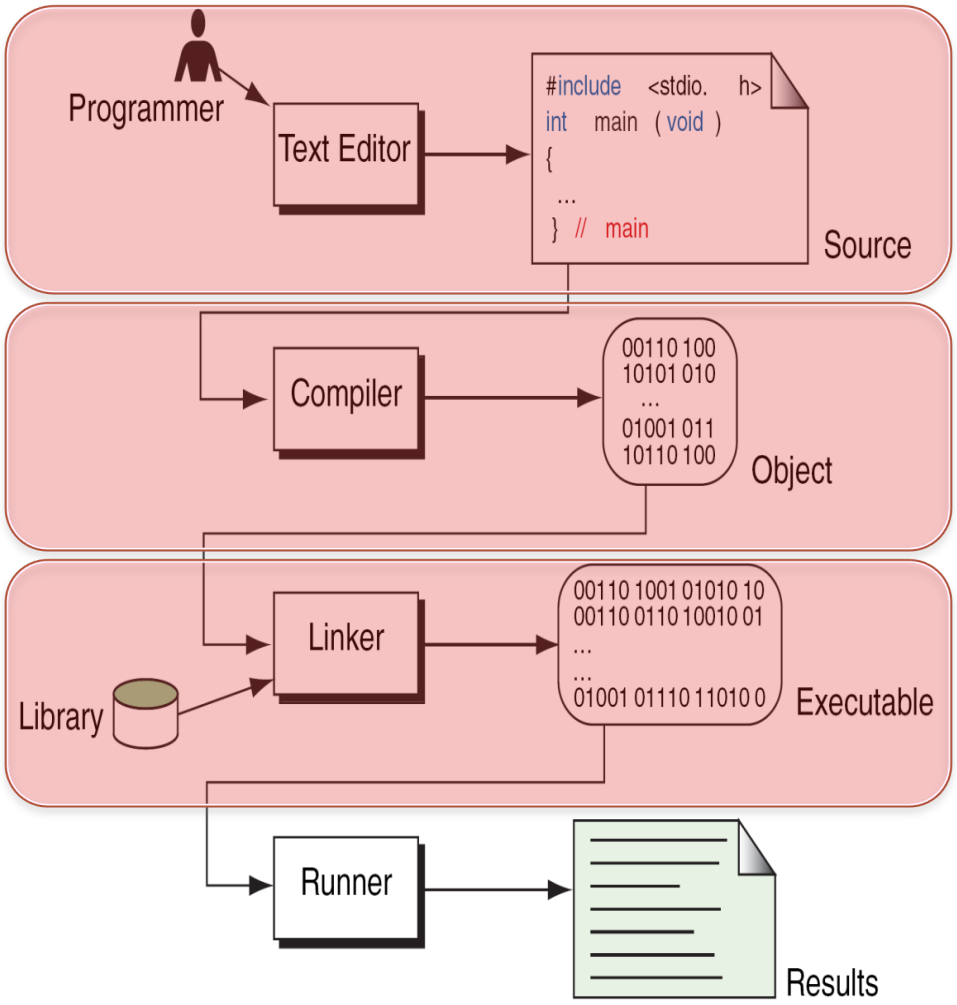
Logic Flow


Turn on the mobile
Key in xxxx - xxxx
Wait for connection
Connection made?
Yes
No
Talking
Hang up

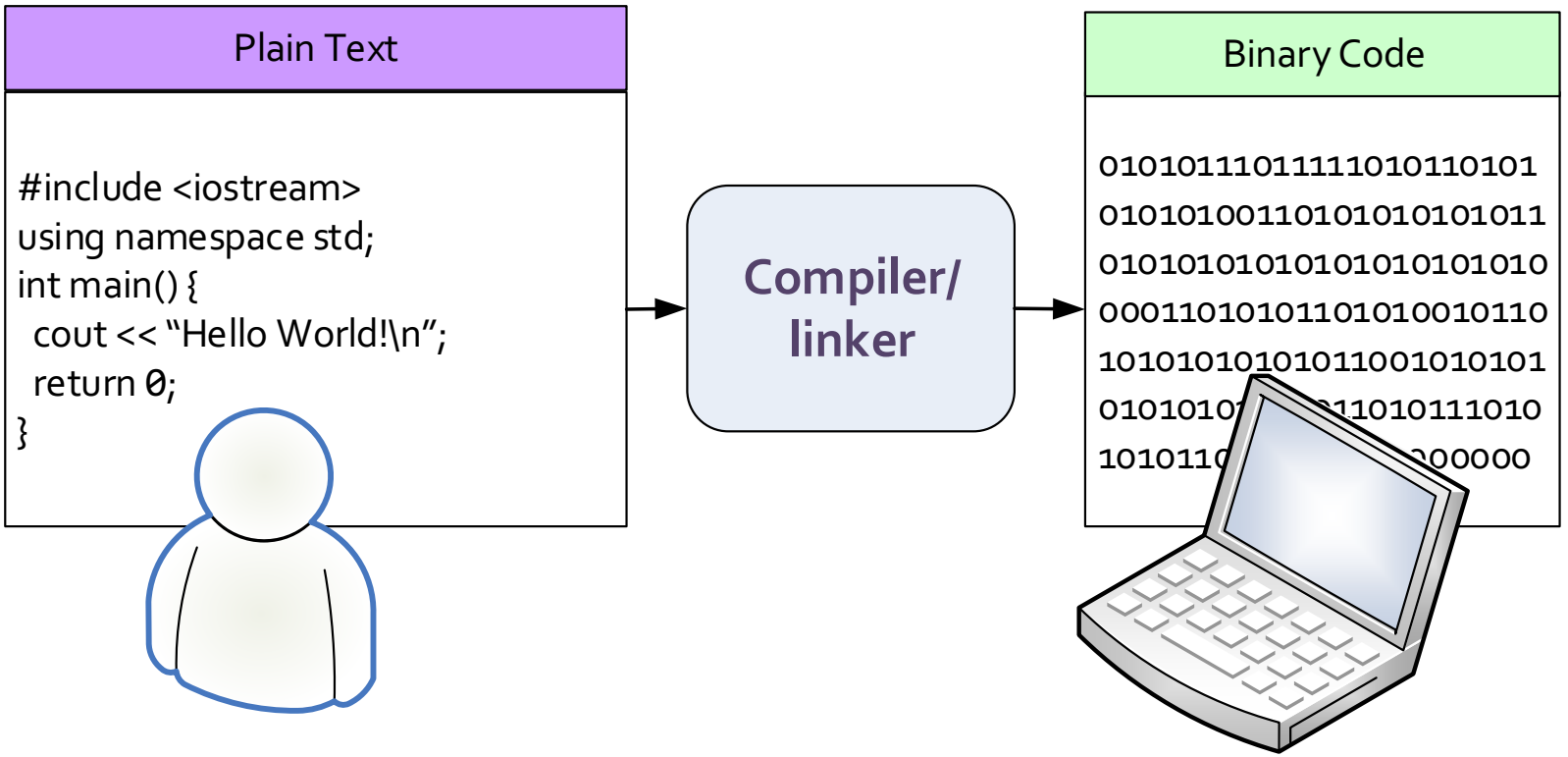

Turn on the mobile
Key in xxxx - xxxx
Wait for connection
No
Connection made?
Yes
Talking
Hang up

Building a C++ Program

- **Writing** source code in C++
  - e.g. hello.cpp
- **Preprocessing**
  - Processes the source code for compilation
- **Compilation**
  - Checks the grammatical rules (syntax)
  - Source code is converted to object code in machine language (e.g. hello.obj)
- **Linking**
  - Combines object code and libraries to create an executable (e.g. hello.exe)
  - Library: common functions (input, output, math, etc.)



Sample Program (Framework)



A Simple Program

```
/* The first program in honor of Dennis Ritchie who invented C at Bell Labs in
   1972 */

#include <iostream>
using namespace std;

int main() {
  cout << "Hello, world!\n";
  return 0;
}
```

Function – main

- **`int main()`**
  - **`int`** means the return value of the function is an integer
  - **`main`** is the **name** of the function
  - **No** semi-colon after **`main()`**
  - C++ is case sensitive:
  - **`void main()`** works for some compilers
    - **`void`** means there is **NO return** value
  - E.g., `void Main()`, `VOID main()` are **incorrect**
  - In this course, we stick to **`int main()`**

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

int main() {
   cout << "Hello, world!\n";
   return 0;
}
```

- **{}**
  - Braces: left brace begins the body of a function. The corresponding right brace must end the function
- **`return 0`**
  - The **`main()`** function has to return an integer upon completion
  - **`0`** is returned to indicate the program exits successfully

Functions – the main function

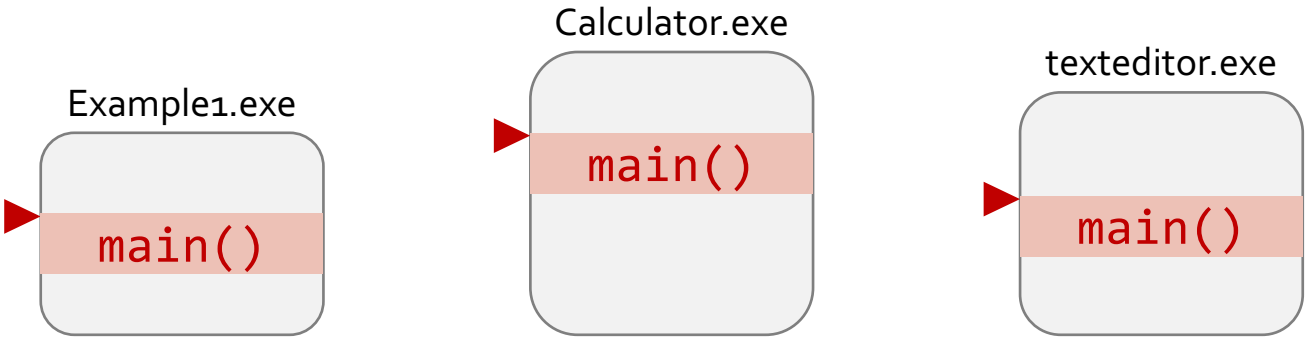
```
int main() {

   return 0;
}
```

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

int main() {
   cout << "Hello, world!\n";
   return 0;
}
```

- The starting point of program
  - the first function called by the computer



Simple Program

```
/* The traditional first program in honor of
   Dennis Ritchie who invented C at Bell Labs
   in 1972 */

#include <iostream>
using namespace std;

int main() {

   cout << "Hello, world!\n";
   return 0;
}
```

Library / SDK /Package

- Normally, we won't write a program all by ourselves. Instead, we will reuse the code written by ourselves / other developers. Especially for the repeating tasks or low-level operations like disk I/O
- The reusing code is well designed and is packed as a library / SDK / Package
- Standard C++ program comes with a set of package to make programmer task easier
- ***iostream*** is one example

Object – cout

```
cout << "Hello, world!\n";
```

- `cout`: "Console OUTput" allows our program to output values to the standard output stream (the screen)
- `cout`: object provided by iostream library (package) for screen (console) output (we will elaborate this concept in future classes)
- `<<`: output (also called insertion) operator that outputs values to an output device. In this case, the output device is `cout` (the screen)
- The value on the right hand side of the operator (`"Hello, world!\n"`) is the string you want to output
  - Any literal (character string) that is to be output must be in between a pair of double quotes

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

int main() {
   cout << "Hello, world!\n";
   return 0;
}
```

Object – cout

- **\n**
  - **escape sequence**: the character following **** is NOT interpreted in the normal way
  - represents a **newline** character: the effect is to advance the cursor on the screen to the beginning of the next line
  - newline: position the character to the beginning of next line
- ****
  - backslash: Insert the backslash character **** in a string
- **\"**
  - double quote: Insert the double quote character **"** in a string
- **endl**
  - Same as the string "**\n**".
  - No **** before **endl**

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

int main() {
   cout << "Hello, world!\n";
   return 0;
}
```

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

int main() {
   cout << "Hello, world!" << endl;
   return 0;
}
```

Syntax Errors

- A simple C++ program will have

```
#include <iostream>    //A preprocessor
using namespace std;   //namespace declarati

int main() {
     /* the starting point of program executio
     cout << Hello, world! \n;
     cout << Hello, world Again! < endl;
     return 0;
}
```

The texts to output should be placed in a pair of double quotes "text".

< is not an operator of **cout**. We need to use **<<**.

We need **;** at the end of each statement

Preprocessor Directives

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

int main() {
  cout << "Hello, world!\n";
  return 0;
}
```

```
using namespace std;
```

- Standard (std) *namespace* is used such that we can use a shorthand name for the element cout
  - `std::cout` <=> `cout`

```
#include <iostream>
```

- Include library **`iostream`** into the program as it contains the definition of **`cout`**, which is used to print something to the screen.
- Load contents of a certain file / library
- **NO** semi-colon at the end of the include directive

Comments

```
/* The traditional first program in honor of
   Dennis Ritchie who invented C at Bell Labs
   in 1972 */
```

- Enclosed by "**/***" and "***/**" OR begin with "**//**"

- **//** single line comments

```
// this is a single line comment
// each line must begin with "//" sign
```

Summary

- Basic components of a computer program are:
  - Instructions
  - Logic Flow
  - Variables and Constants
- A correct logic is important in programming
- Programmer usually reuse code written by the others and the code is commonly in form of library / SDK / packages

Summary

- A simple C++ program will have

```
#include <iostream>      // a preprocessor directive
using namespace std;     // namespace declaration
int main() {
     /* the starting point of program execution  */

     return 0;
}
```