Chapter 02

C++ Program Structure and its Components

Lê Thành Sách

Content

- A Sample C++ Program
- Coding Styles
- Data and standard Output Function
- Standard Input Function
- Data Processing: Simple Example

A Sample C++ Program

```
#include <iostream>
#include <stdlib.h>

/*
This is a simple program.
The objective of this program is to print the following information on different
lines
1. University's name
2. Course's name
*/
void main(){
     // “std::cout" is a function, which is defined in <stdio.h>
     std::cout << "Ho Chi Minh city University of Technology\n";
     std::cout << "Programming Fundamentals\n";

     // "pause" causes the program pause before executing the next statement.
     // “system" is a function, which is defined in <iostream>
     system("pause");
}
```

A Sample C++ Program

Components

- Declare libraries to be used
  - The **#include** directive:
    - std::cout: standard output stream
    - std::cout: **defined** in `<iostream>`, so we need to `#include <iostream>`
    - "system": a function, for calling system commands, e.g., "pause", "cls" ….
    - "system": **defined** in `<stdlib.h>` so we need to `#include <stdlib.h>`

```
#include <iostream>
#include <stdlib.h>

void main(){
    std::cout << "...";
    std::cout << "...";

    system("pause");
}
```

A Sample C++ Program

Components

- The statement is executed first
  - The first statement inside the main function
  - Each c++ program must have a main function

```
void main(){


}
```

The "{" bracket marks the beginning of the body of the `main function`

The "}" bracket marks the end of the body of the `main function`

A Sample C++ Program

Components

- Complete prototype for function “main”
  - argc: number of arguments in executable command line, including program’s name.
  - argv: list of arguments

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


}
```

A Sample C++ Program

Components

- The statement is executed first
  - The first statement inside the main function
  - Each c++ program must have a main function

```
int main(){


}
```

This is the return type of function (main)
`int`: Function "main" returns the status code: 0 = no error; 1 = error
EXIT_SUCCESS = 0; EXIT_FAILURE = 1

A Sample C++ Program
Components

- Comment
  - Comments are used to explain code; it is not required to write comments in C++ grammar, because compilers will not analyse them
  - There are two type of comment available in c++:
    - Multi-line comments: Start with “/*” and end with “***/***”
    - Single-line comments: Start with “***//***” and continue until the end of the line

```
#include <iostream>
#include <stdlib.h>


/*
The objective of  ...
 ...
This is  ...
*/
void main(){
      std::cout("..."); // “std::cout" is  ...
      std::cout("...");

      system("pause"); // "pause" causes  ...
}
```

Coding Styles

- The program is as well as a text in natural language.
- Therefore, it should be written in a way such that it is easy for reading, understanding and detecting errors.
- In large projects, programming style is very important.

Coding Styles

- The list of topics can be found in the following pages
- Students must practice the coding style during the learning process, through practice.

- See more in:

  http://users.ece.cmu.edu/~eno/coding/CCodingStandard.html

  http://www.cs.swarthmore.edu/~newhall/unixhelp/c_codestyle.html

Coding Styles
Suggestions

- Guidelines for naming something:
  - Function
    - Starts with a verb, because the function is a processing unit (i.e., action execution), follows by some other words, connects words by a underscore(_)
    - The function's name must reflect the meaning/purpose of that function.

    Example:

    Should use: check_for_errors()

    Instead of: error_checking() or dump_data_to_file()

Coding Styles
Suggestions

- Guidelines for naming something:
  - Function
    - Should use the following “suffix” or “prefix” to increase its semantic.
    - Prefixes
      - is — Check if a condition is satisfied
      - get — Get value
      - set — Asign value

Coding Styles
Suggestions

- Guidelines for naming something:
  - Variable
    - Starts with a noun
    - All characters in the name are lowercase
    - Connect words together with underscore, "_"

    Example:

    unit32 timeout_msec;

    Time time_of_error

Coding Styles
Suggestions

- Guidelines for naming something:
  - Variable
    - Starts with a noun
    - All characters in the name are lowercase
    - Connect words together with underscore, "_"
  - Variable type pointer
    - Put a * next to the variable name
    - Suffix with "ptr" if possible

    Example:

    ```
    char     *name;
    Student *student_ptr
    ```

Coding Styles
Suggestions

- Guidelines for naming something:
  - Global variable
    - Prefix with "g_", to distinguish in code

    Example :

    ```
    Logger  g_logger;
    Logger  g_logger_ptr;
    ```
  - Global constant
    - Use uppercase letters

    Example:

    ```
    const int A_GLOBAL_CONSTANT = 5;
    ```

Input and Output

- Output:
  - Use output stream
- Input:
  - Use input stream

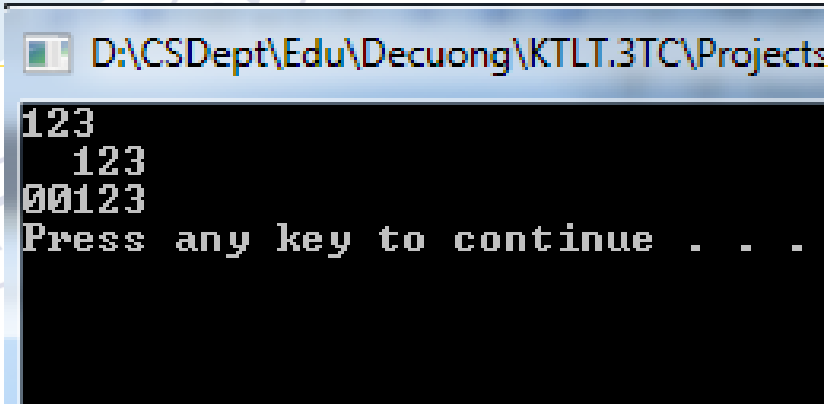
Input and Output

- Output stream
  - Standard output stream: “std::**cout**”, defined in <iostream>
  - String stream: **ostringstream**, **stringstream**, defined in <sstream>
  - Stream created from files: **ofstream**, defined in <iostream>
- Input stream
  - Standard output stream: “std::**cin**”, defined in <iostream>
  - String stream: **istringstream**, **stringstream**, defined in <sstream>
  - Stream created from files: **ifstream**, defined in <iostream>

Data and Standard Output Function in C++

- Integer type
  - "int": a data type, more in next chapter.
  - The value written in the code (literal constant):
    - Example: 15, -20, 40, etc
  - "setw", "setfill": manipulators to control the format of the output

```
#include <iostream>
#include <stdlib.h>
#include <iomanip>

using namespace std;

void main(){
     cout <<  123 << endl;
     cout << setw(5) << 123 << endl;
     cout << setfill('0') << setw(5) <<
123 << endl;
     system("pause");
}
```



Data and Standard Output Function in C++

Coding style: practice

```
#include <iostream>
#include <stdlib.h>
#include <iomanip>

using namespace std;

void main(){
	cout <<  123 << endl;
	cout << setw(5) << 123 << endl;
	cout << setfill('0') << setw(5) << 123 << endl;

	system("pause");
}
```

Use TAB to align the code

Should!

Data and Standard Output Function in C++

Practice programming style

```
#include <iostream>
#include <stdlib.h>
#include <iomanip>

using namespace std;

void main(){
cout <<  123 << endl;
cout << setw(5) << 123 << endl;
cout << setfill('0') << setw(5)
<< 123 << endl;
system("pause");
}
```

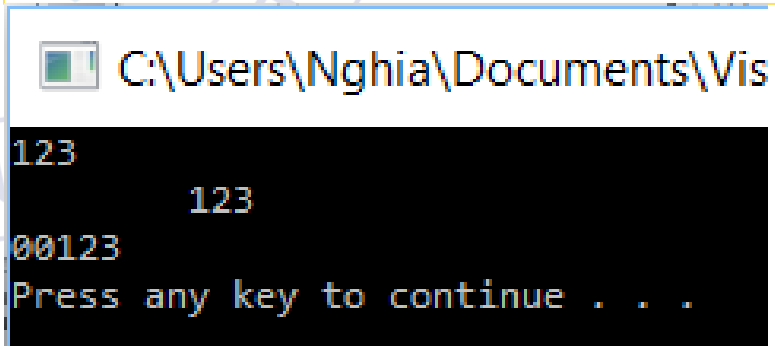
```
#include <iostream>
#include <stdlib.h>
#include <iomanip>

using namespace std;

void main(){
  cout <<  123 << endl;
    cout << setw(5) << 123 <<
endl;
cout << setfill('0') << setw(5)
<< 123 << endl;
  system("pause");
}
```

Should not!

Data and Standard Output Function in C++

- Special characters
  - \n (newline)
  - \t (tab)
  - \v (vertical tab)
  - \f (new page)
  - \b (backspace)
  - \r (carriage return)
  - \n (newline)

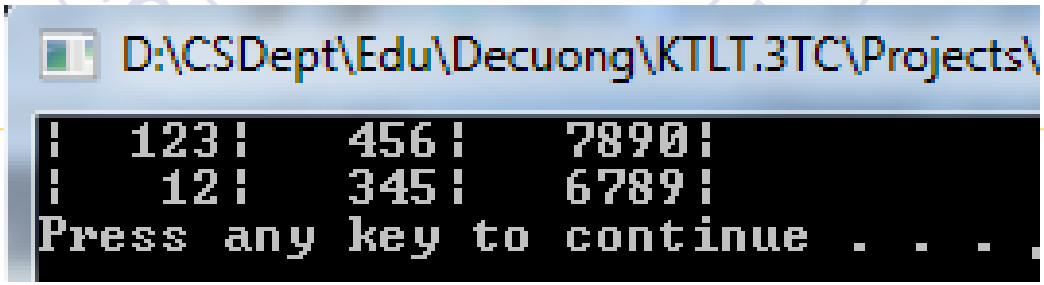
```
#include <iostream>
#include <stdlib.h>
#include <iomanip>
using namespace std;

void main(){
     cout <<  123 << "\n";
     cout << "\t" << 123 << endl;
     cout << setfill('0') << setw(5)
<< 123 << endl;
     system("pause");
}
```



Data and Standard Output Function in C++

- Integer type
  - Print multiple number using “cout” only one time

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

void main(){
        cout << "|" << setw(5) << 123<< "|" << setw(5) <<
456 << "|" << setw(5) << 7890 << "|"<< endl;
        cout << "|" << setw(5) << 12<< "|" << setw(5) <<
345 << "|" << setw(5) << 6789 << "|"<< endl;
        system("pause");

}
```



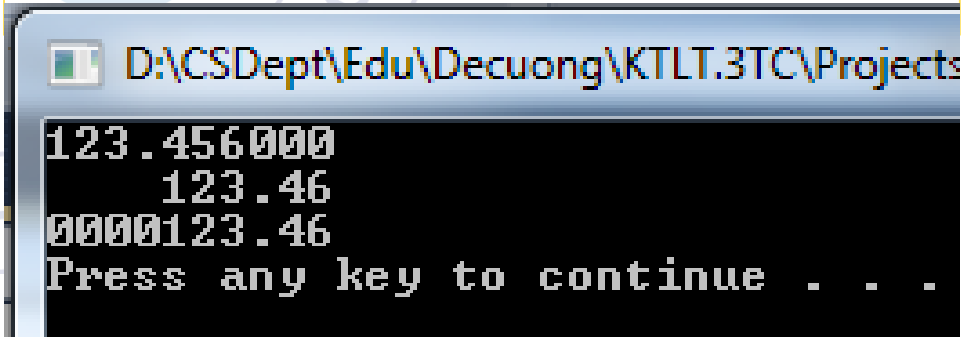
Data and Standard Output Function in C++

- Real type
  - Single precision: type name "float"
  - Double precision: type name "double"
  - Will discuss more in the next chapter
  - The value written in C++ code (literal constant):
    - Example:

    float: 15.5f

    double: 15.5

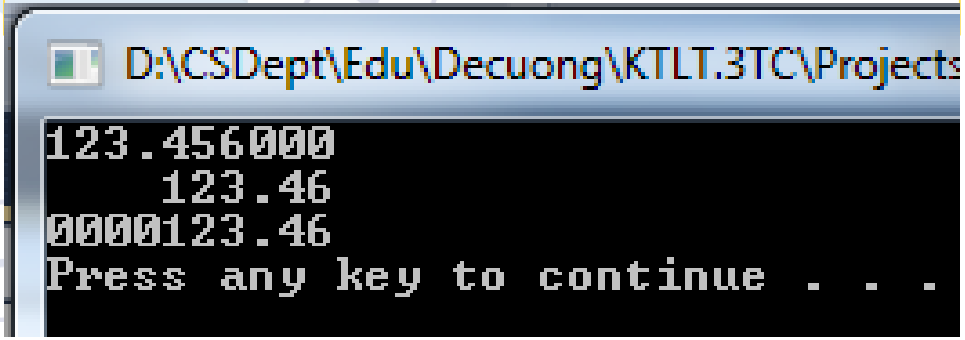
```
#include <stdio.h>
#include <iomanip>
using namespace std;

void main(){
    cout << setw(10) << setfill('0') << left
<< 123.456 << endl;
    cout << setw(10) << setfill(' ') <<
right << 123.46<< endl;
    cout << setw(10) << setfill('0') <<
123.46<< endl;

    system("pause");
}
```



Data and Standard Output Function in C++

- Real type
  - Question: How to print only three numbers after "." ?
  - Question: How to control the width of the whole number in screen, including dot "."?

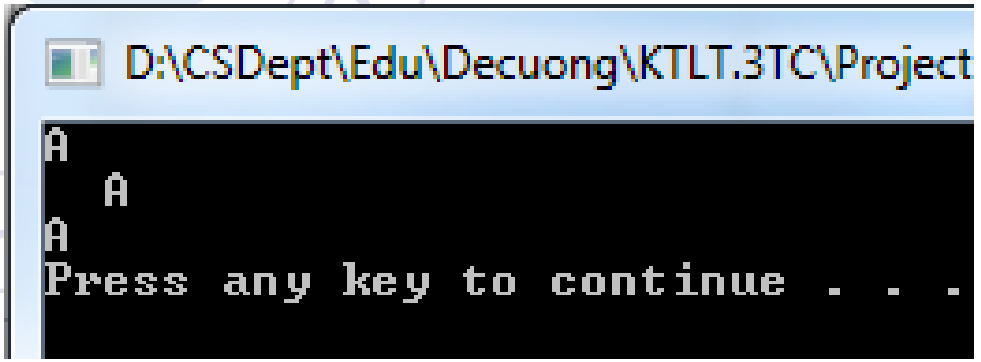
```
#include <stdio.h>
#include <iomanip>
using namespace std;

void main(){
    cout << setw(10) << setfill('0') << left
<< 123.456 << endl;
    cout << setw(10) << setfill(' ') <<
right << 123.46<< endl;
    cout << setw(10) << setfill('0') <<
123.46<< endl;

    system("pause");
}
```



Data and Standard Output Function in C++

- Char type
  - Type name: "char", we will discuss more in the next chapter.
  - The value written in C++ code (literal constant):
    - Example: 'A', 'a', etc.

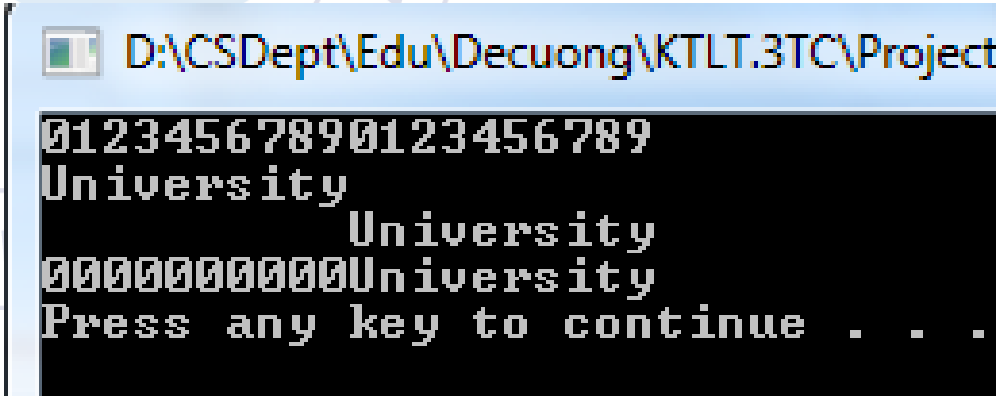
```
#include <iostream>
#include <iomanip>
using namespace std;

void main(){
    cout << 'A' << endl;
    cout << setw(4) << 'A' << endl;
    cout << 'A' << endl;
    system("pause");
}
```



Data and Standard Output Function in C++

- String type
  - Type name: "std::string", we will discuss more in the next chapter.
  - The value written in C++ code (literal constant):
    - Example: "This is a string", "abcdef", etc.

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

void main(){
     cout << "01234567890123456789" << endl;
     cout << "University" << endl;
     cout << setw(20) << "University" << endl;
     cout << setw(20) << setfill('0') <<
     "University" << endl;

     system("pause");
}
```



Data and Standard Output Function in C++

- Summary:
  - "**cout**": standard output stream in C++ program
  - To output something (variables, constants, experessions, functions): use operator "<<"
  - To format the output, use manipulator in `<iomanip>` l
    - fixed, left, right, …
    - setw: width of the next values in screen
    - setprecision: number of digits after decimal symbol (.)
    - setfill: prefix character in output

Data and Standard Output Function in C++

- Special characters:
  - \' single quote
  - \“double quote
  - \? question mark
  - \\ backslash
  - \a audible bell
  - \b backspace
  - \f form feed - new page
  - \n line feed - new line
  - \r carriage return
  - \t horizontal tab
  - \v vertical tab

Standard Input Function in C++

- Rule
  - When reading data from the keyboard, program stores it into a area somewhere in virtual memory space.
    - Such the area must be specified by a type, called **data type**.
    - The area has a name. That name is called **variable**.

Standard Input Function in C++

- Standard Input Function
  - "std::cin": input stream
  - Syntax:

```
std::cin      >>                    ;
```

Extraction operator

Variable or the name of memmory area.

Can use multiple ">> __" to get input multiple time;

Standard Input Function in C++

- Integer number

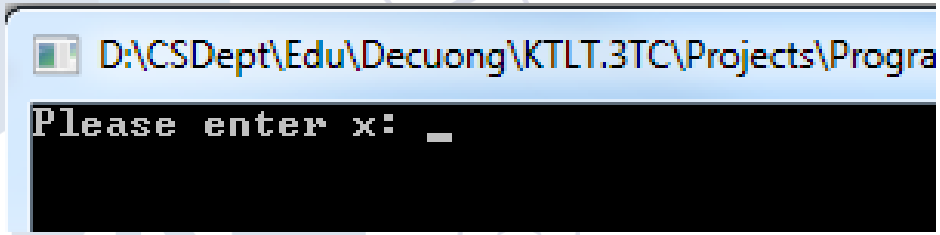
```
#include <iostream>
using namespace std;

void main(){
    int x;
    cout << "Please enter x: ";
    cin >> x;
    cout << "x = " << x << endl;

    system("pause");
}
```

Which is a datatype?
Which is a variable?

Standard Input Function in C++

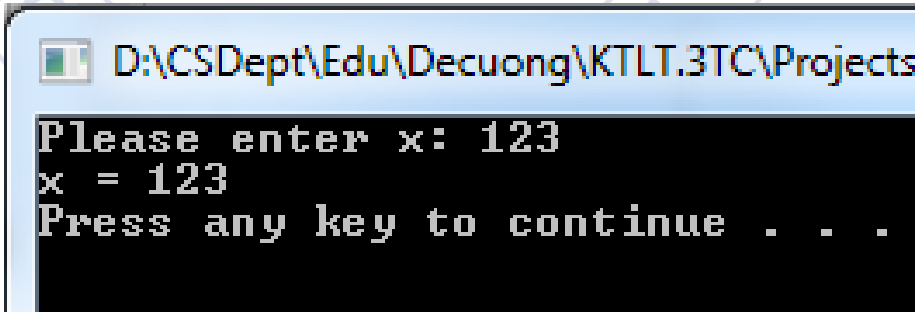
- Integer number

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

void main(){
	int x;
	cout << "Please enter x: ";
	cin >> x;
	cout << "x = " << x << endl;

	system("pause");
}
```



(The program is waiting for the user to enter an integer number)

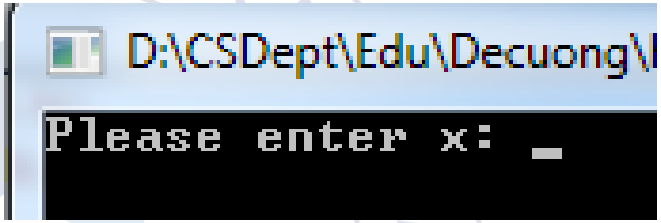
The screen after entering value 123 and press ENTER key:



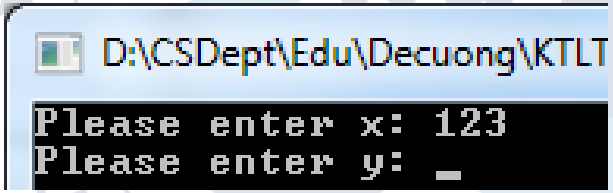
Standard Input Function in C++

- The program reads two integers

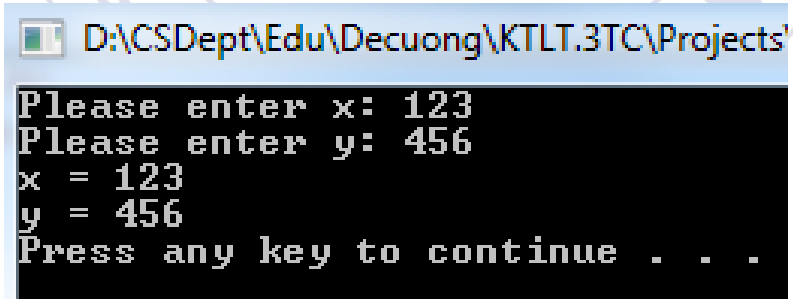
The screen at the beginning:



The screen after enter value 123 and pressing ENTER key:



The screen after enter the value 456 and pressing the ENTER key:



Standard Input Function in C++

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

void main(){
    int x;
    int y;
    cout << "Please enter x: ";
    cin >> x;
    cout << "Please enter y: ";
    cin >> y;

    cout << "x = " << x << endl;
    cout << "y = " << y << endl;

    system("pause");
}
```

Standard Input Function in C++

- The code is shorter but has the same function
  - Two memory areas are declared on the same line
  - “std::cin”: reads two values x and y
  - “std::cout”: prints x and y
  - Use control characters “\n” in correct place

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

void main(){
     int x, y;
     cout << "Please enter x and y: ";
     cin >> x >> y;

     cout << "x = " << x << "\ny = " << y << endl;

     system("pause");
}
```

Standard Input Function in C++

- std::cin, see more in following pages:
  - http://www.cplusplus.com/reference/iostream/cin/
  - https://www.tutorialspoint.com/cplusplus/cpp_basic_input_output.htm

Data Processing: Simple Example

- The program requirements:
  - Read two numbers x and y as coordinates in 2D space.
  - Calculate and print the distance from the center of coordinate axes to points (x, y)

```
#include <iostream>
#include <math.h>
using namespace std;

void main(){
    float x, y;
    cout << "Please enter coordinates x and y: ";
    cin >> x >> y;

    cout << "(x,y) = (" << x << ", " << y << ")" << endl;
    cout << "Distance to the origin = " << sqrt(x*x + y*y) << endl;
    system("pause");
}
```

Data Processing: Simple Example

- We need to include the file <math.h> to use the math library
- sqrt(.): Calculates the square root of the value passed to it and returns the result

```
#include <iostream>
#include <math.h>
using namespace std;

void main(){
    float x, y;
    cout << "Please enter coordinates x and y: ";
    cin >> x >> y;

    cout << "(x,y) = (" << x << ", " << y << ")" << endl;
    cout << "Distance to the origin = " << sqrt(x*x + y*y) << endl;
    system("pause");
}
```