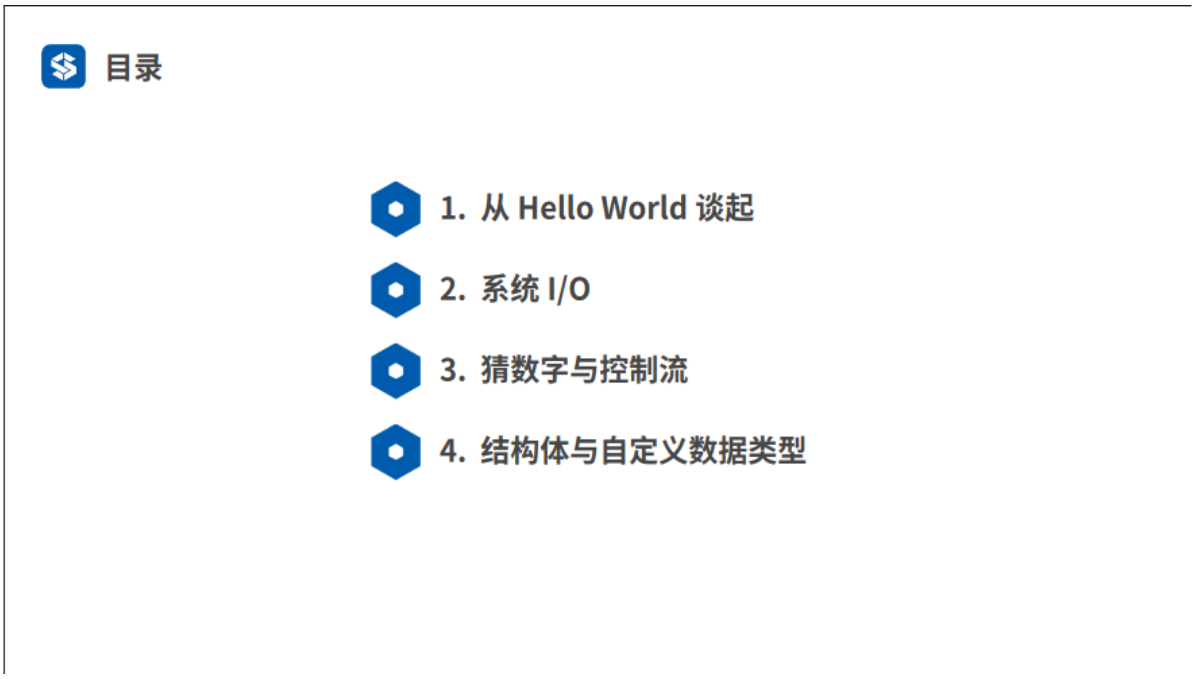
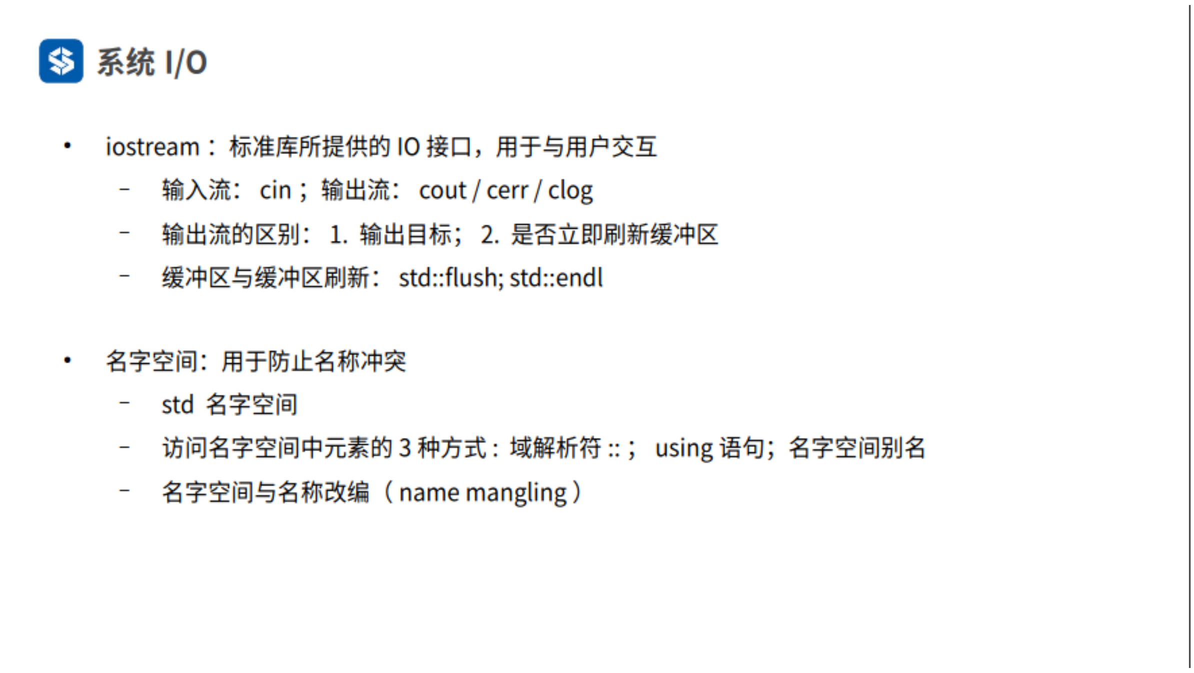
chapter 1 C++ 初探



目录

1、hello world

2、系统IO

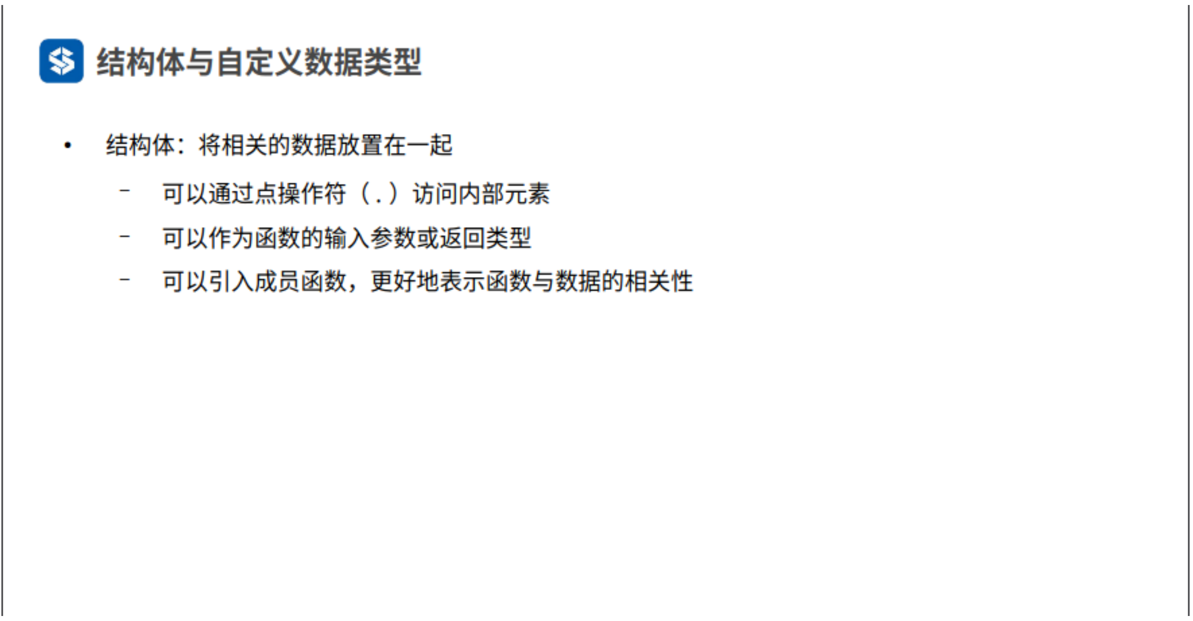


系统 I/O

- iostream：标准库所提供的 IO 接口，用于与用户交互
  - 输入流： cin；输出流： cout / cerr / clog
  - 输出流的区别： 1. 输出目标； 2. 是否立即刷新缓冲区
  - 缓冲区与缓冲区刷新： std::flush; std::endl
- 名字空间：用于防止名称冲突
  - std 名字空间
  - 访问名字空间中元素的 3 种方式：域解析符 :: ； using 语句；名字空间别名
  - 名字空间与名称改编（name mangling）

```
/*******************  输乳流: cin  ******************************************
 * 用法：先定义一个变量 a ，然后使用 cin 指向这个变量，最后程序运行的时候，
 * 就会提示您输入一个数字给该变量赋值：
 * 如： cin >> a ;
*****************************************************************************/

/********************** 输出流: cout / cerr / clog ****************************
 *    std::cout 标准输出流，可以被重定向.
*    std::cerr 标准错误输出流  对应与C 语言中的 stderr ， 用于显示错误信息，其会立即刷新缓冲区，无需等待缓冲区满，因此比较可以及时地输出一些关键信息
*    std::clog  标准日志输出流 ， 其不会及时刷新缓冲区，因此当程序意外终止的时候，可能来不及想屏幕或者重定向文件输出 日志 log.
*
*******************************************************************************/

/****************** 主动刷新 输入缓冲区 : std::flush / std::endl ****************************
*
*    std::flush : 强制刷新缓冲区 ，
*
*    std::endl:   换行 + 强制刷新缓冲区 ，
*
***********************************************************************************************/
```

3、结构体



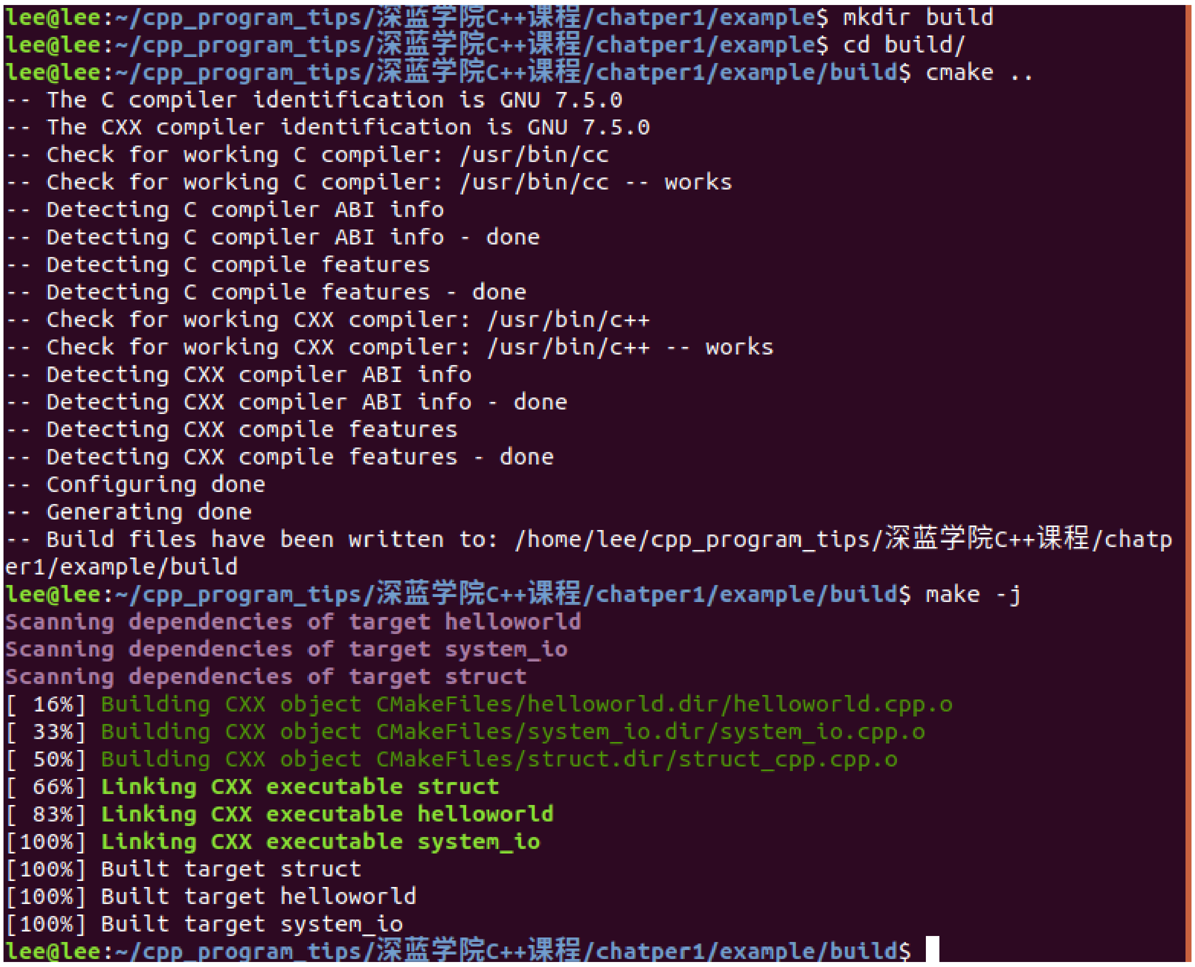
结构体与自定义数据类型

- 结构体：将相关的数据放置在一起
  - 可以通过点操作符（.）访问内部元素
  - 可以作为函数的输入参数或返回类型
  - 可以引入成员函数，更好地表示函数与数据的相关性

```
#include <iostream>

// 结构体通常使用 . 操作符号来访问该元素 。
// c++ 之中，结构体也是可以引入 函数的，如下所示：
struct Point{
  int x;
  int y;
  void change_x()
  {
    x = x + 1 ;
    y = y + 1 ;
  }
};

int main()
{
  Point P;
  P.x = 4 ;
  P.y = 5 ;
```

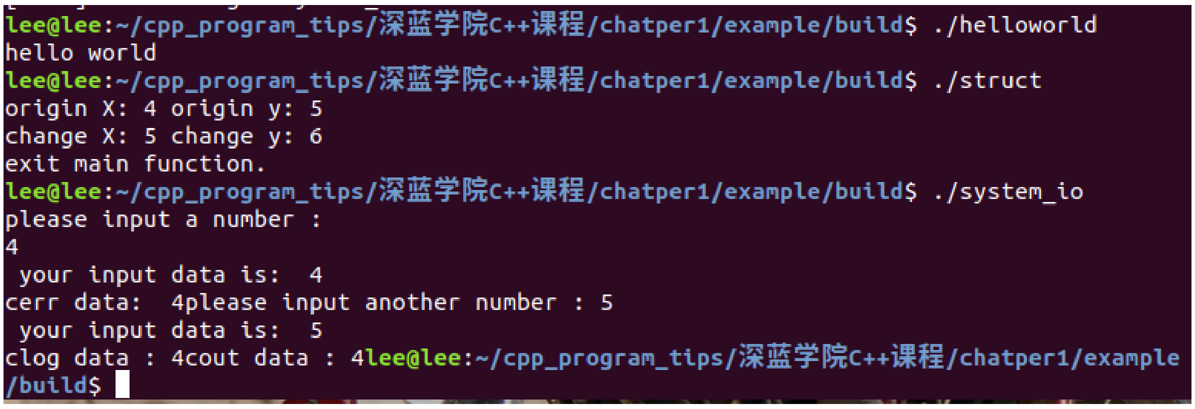
```
    std::cout<< "origin X: "<<P.x << " origin y: "<<P.y << std::endl;
    P.change_x();

    std::cout<<"change X: "<<P.x<< " change y: "<<P.y<<std::endl;
    std::cout<<"exit main function. "<<std::endl;
    return 0;
  }
```

编译方法:



```
lee@lee:~/cpp_program_tips/深蓝学院C++课程/chatper1/example$ mkdir build
lee@lee:~/cpp_program_tips/深蓝学院C++课程/chatper1/example$ cd build/
lee@lee:~/cpp_program_tips/深蓝学院C++课程/chatper1/example/build$ cmake ..
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/lee/cpp_program_tips/深蓝学院C++课程/chatp
er1/example/build
lee@lee:~/cpp_program_tips/深蓝学院C++课程/chatper1/example/build$ make -j
Scanning dependencies of target helloworld
Scanning dependencies of target system_io
Scanning dependencies of target struct
[ 16%] Building CXX object CMakeFiles/helloworld.dir/helloworld.cpp.o
[ 33%] Building CXX object CMakeFiles/system_io.dir/system_io.cpp.o
[ 50%] Building CXX object CMakeFiles/struct.dir/struct_cpp.cpp.o
[ 66%] Linking CXX executable struct
[ 83%] Linking CXX executable helloworld
[100%] Linking CXX executable system_io
[100%] Built target struct
[100%] Built target helloworld
[100%] Built target system_io
lee@lee:~/cpp_program_tips/深蓝学院C++课程/chatper1/example/build$
```

```
# 打开终端
cd 进入 ............/cpp_program_tips/深蓝学院C++课程/chatper1/example
mkdir build
cd build
cmake ..
make -j

# 分别运行 .
./helloworld
./struct
./system_io
```



```
lee@lee:~/cpp_program_tips/深蓝学院C++课程/chatper1/example/build$ ./helloworld
hello world
lee@lee:~/cpp_program_tips/深蓝学院C++课程/chatper1/example/build$ ./struct
origin X: 4 origin y: 5
change X: 5 change y: 6
exit main function.
lee@lee:~/cpp_program_tips/深蓝学院C++课程/chatper1/example/build$ ./system_io
please input a number :
4
 your input data is:  4
cerr data:  4please input another number : 5
 your input data is:  5
clog data : 4cout data : 4lee@lee:~/cpp_program_tips/深蓝学院C++课程/chatper1/example
/build$ 
```