

作业练习1：C++基础语法与知识点

面向对象程序设计 (C++)

WHUT-CS 2023 Spring

I. 作业目的

在编写设计良好的程序时，开发人员通常遵循以下几个步骤：

- **A.** 程序的目标任务、需求和约束
- **B.** 为实现目标任务设计适当的算法和数据结构
- **C.** 通过专门的程序设计语言编写程序实现所需的算法和数据结构

本次作业主要关注上述步骤中的**(C)**部分，其它部分将通过其他作业练习完成。作业练习中将针对不同的知识点 (Knowledge Items) 进行考察，通过编写实现指定功能点的程序巩固课程内容所涵盖的知识范围。

注意：作业练习的考察范围涵盖本课程的全部学习内容，不完全遵循授课章节的顺序，也不完全遵循课堂讲授范围，部分知识点的学习和运用需根据教材或参考书内容加以自学

II. 作业要求

编写完成的程序需满足以下要求：

- **A.** 尽量使用C++语法风格的编程方式实现程序功能需求，避免使用C语法风格。例如：
 - 对于输入/输出操作，应使用C++中的 `cout` 和 `cin` 格式，而不应使用C中的 `printf` 和 `scanf` 格式
 - 包含头文件时，应使用C++中的 `cstring` 代替C中的 `string.h`
- **B.** 严格保持整洁的代码格式，并使用适当的注释规则对程序代码进行标注。例如：
 - 对于作业内容中的每条编程任务，在程序中应使用对应序号在相关代码处进行标注，格式为 `/* <n> */`

例如：对于第5项作业任务

```
5: Declare a variable of type int
```

程序中的对应代码及相关标注可以如下编写

```
/* <5> */  
int x;
```

- 如果某条编程任务需要通过程序中的若干代码片段完成，则对应序号只需要在每个源程序中标注一次
- 如果某行或某段代码同时实现了多条编程任务，则可以将多个对应任务序号标注在一起

例如：如果一段表达式同时实现了任务4和任务6的需求，则可按照如下格式标注

```
/* <4> <6> */  
// the implementation statements for task 4 and task 6
```

III. 相关知识点

本次作业练习内容主要涵盖以下知识点

- C++/C++2011程序的命令行编译
- C++程序结构及.cpp与.h文件之间的关联关系
 - 如何引用在其他文件中定义的标识符
- 基本数据类型、常量、变量、与数组
- 复合数据类型
- 结构、联合、枚举
- 变量与数组的初始化
- 在不同的内存区域中创建并存储数据
 - 静态内存区域
 - 调用栈区域
 - 内存堆区域
- 指针与引用
- 函数的定义、调用与参数传递
 - 在函数调用中传递数组参数
 - 使用函数解决简单的计算问题
- 递归函数的使用
- 循环语句
 - C++中新增的for循环语句风格
- 选择语句
- switch语句
- 输入与输出
 - cin 与 cout 的使用
 - 用户输入状态
 - 清除输入队列
- 文件的读写
- 命名空间的使用
- 模板方法的概念

IV. 作业内容

本节以任务点方式列出需要完成的设计任务，最终编写完成的程序中应涵盖所有任务功能点，并按照前述要求进行标注。部分任务中还标注了相关知识点和编程提示供参考。

1. 为完成本次作业的编程任务，需创建5个代码文件，`hmk1.cpp`，`tools.cpp`，`data.cpp`，`tools.h`，and `info.txt`。

[KI] 程序中使用多个源文件及其相互关系

[Design]

- hmk1.cpp是整个程序的入口文件，它包含程序的main()函数。
- tools.cpp中声明了程序中要用到的工具类函数。
- tools.h是与tools.cpp对应的头文件。
- data.cpp 中声明了一些在其他源代码文件中要用到的数据内容。
- info.txt用于在程序中进行数据的输入和输出操作。

2. 为能够正常使用 `cout` 和 `cin`，在 `hmk1.cpp` 和 `tools.cpp` 中包含必要的头文件。
3. 为能够正常调用库函数 `isspace` and `strlen`，在 `tools.cpp` 中包含必要的头文件引用。
4. 在 `tools.cpp` 和 `hmk1.cpp` 中包含头文件 `tools.h`
5. 在 `hmk1.cpp` 中声明主函数 `main()`，该文件中应该仅包含这一个函数的声明。
6. 除了主函数之外，其它的所有函数都应该被声明在 `tools.cpp` 中。
7. 除了需要被隐藏的函数，对于本作业后续要在 `tools.cpp` 中编码实现的所有函数，都应该将其函数原型声明在 `tools.h` 文件中。

[KI] content of head files.

[Hint] you can decide which functions defined in `tools.cpp` are "hidden" (internal linkage).

8. 在 `tools.h` 文件中使用两种不同的方式定义两个符号常量 `TITLE_LEN` 和 `NAME_LEN`，它们的值分别被初始化为 80 和 50。

[Hint] One by `#define` and the other by a `const` definition.

9. 在 `hmk1.cpp` 文件的 `main` 函数中，按照下列要求声明若干局部变量
 - 为每一种char类型声明一个变量. 在C++中至少有5种不同的字符类型。
 - 为每一种整数类型声明一个变量。
 - 为每一种浮点类型声明一个变量。

[KI] Data types, constants, and storage holders

10. 为上一步中声明的每一个变量进行初始化赋值。

[Hint] You can either initialize a variable by the same statement that declares it, or by a separate statement.

11. 在 `tools.h` 文件中声明一个结构体 `Book`，并且包含以下的几个数据域：
 - `title`，字符数组类型，以第9题中声明的 `TITLE_LEN` 作为其数组长度。
 - `authors`，字符数组类型，以第9题中声明的 `NAME_LEN` 作为其数组长度..
 - `price`，浮点类型。
 - `id`，联合类型，包含以下两个数据域（可以选择使用嵌套的匿名联合类型或单独定义的具名联合类型来声明该字段）
 - `isbn` . 用于保存一个字符串

[Hint] For its implementation, you can choose a char array or a string object

- `library_id` , 用于保存一个整型数据.

[Design] A book needs some id, which is either some ISBN number or a number chosen by the library, but not both. Therefore, for saving memory space, we can let id has an union type.

12. 在 `tools.h` 中声明一个枚举类型 `Weekday` , 在其中用不同的枚举值 (例如: `Mon ... Sun`) 表示一周中的每一天.
13. 在 `tools.h` 中声明一个枚举类型 `Month` , 在其中用不同的枚举值 (例如: `Jan ... Dec`) 表示一年中的每一月.
14. 在 `tools.h` 中声明一个结构体 `Date` , 在其中包含如下3个数据域:
 - `year` , 无符号整型
 - `month` , `Month` 类型 .
 - `weekday` , `Weekday` 类型 .
15. 在 `hmk1.cpp` 中创建一个 `Date` 类型的全局变量 `today` , 且该变量应具有外部链接属性(external linkage), 能够在其它文件中被使用.
16. 在 `data.cpp` 中创建一个容量为 `NAME_LEN` 的全局字符数组 `myName` , 它应该能够在其它文件中被使用.
17. 在 `data.cpp` 中创建一个 `Book` 类型的全局变量 `bestBook` , 且该变量也具有外部链接数据 .

[Design] The global names `today` , `myName` and `bestBook` are defined in two different files, and they all can be used by some other files.

18. 在主函数 `main()` 中, 请求用户通过键盘输入数据来为 `today` , `myName` 和 `bestBook` 进行赋值, 并在屏幕上打印出赋值后的结果.
19. 在主函数 `main()` 中声明一个整型常量 `x` .
20. 在主函数 `main()` 中声明一个整型指针 `xp` , 并满足下列要求:
 - 该指针的值能够改变, 即指针的指向能够改变.
 - 不能通过该指针改变其所指向的目标变量的值.
21. 声明另一个指针 `xp2` , 并满足下列要求:
 - 该指针应指向一个包含5个元素的 `int` 型数组.
 - It cannot be used to modify the data that it points to.
22. 声明一个包含5个元素的数组 `ar1` , 该数组的每个元素应用是一个整型变量的地址.
23. 声明一个函数指针 `fp1` , 并且初始化该指针指向库函数 `strlen` . 在该声明中不要使用关键字 `auto` .
24. 声明一个函数指针 `fp2` , 并且初始化该指针指向库函数 `strncpy` . 在该声明中可以使用关键字 `auto` .
25. 基于所使用的 g++/clang/msvc 编译器, 写出将本程序编译为可执行程序命令行编译指令, 并在文件 `hmk1.c` 的头部用注释格式标注出该条指令.

[KI] Compilation

[Hint] All the .cpp files of the program are needed by the command line. Some features are supported only C++11, like the usage of `auto`. To allow c++11 features, some special option is needed in a command to use the specified compiler to compile the program. *

26. 在一条单独的表达式中, 利用 `cout` 打印输出至少3个不同变量的值.

[KI] input and output

27. 利用系统中的符号常量打印输出系统支持的最大的 `int` 值和最小的 `long long` 值.

28. 编写一条C++表达式, 在声明一个整型变量的同时对该变量进行初始化, 且该表达式在C语言中是不合法的表达式.(即该表达式的语法应为C++中新增的变量初始化方式)

29. 在屏幕上打印输出第10题中声明的所有变量, 且不允许使用 `printf` 函数.

[Hint] to print out a constant of each character type, see page 89 and 90 of the book "C++ Primer Plus (edition 6)". Do not use `printf`.

30. 声明一个整型变量, 并通过 `cout` 打印输出其十进制、八进制和十六进制的值.

31. 编写一条C++表达式, 通过 `cout` 打印输出字符 'A', 及其对应的整型值.

32. 通过 `sizeof` 运算符计算并打印输出一个数组所占用的内存字节数.

33. 基于C语言中的字符数组完整保存用户输入的一个字符串, 该字符串应是由多个单词和空格组成的一条英文语句.

34. 基于C++中的字符串类 `string` 完整保存用户输入的一个字符串, 该字符串应是由多个单词和空格组成的一条英文语句.

35. 用户输入一个字符并将其保存在一个 `char` 型变量中, 这个字符可能是空格或回车键.

36. 在主函数 `main` 中完成下列功能:

- 打开文件 `info.txt`.
- 将变量 `myName` 中保存的字符串内容输出到文件 `info.txt` 中保存,
- 读取文件 `info.txt` 中的内容, 并将其打印输出到屏幕上.

[Hint] This is to check whether the operating of writing to file is successful. You can do it by reading the characters one by one and print them, or you can read the file content into some string (object or character array) and print the string.

37. 在文件 `tools.cpp` 中定义一个函数 `print_my_name`, 该函数打印输出在文件 `data.cpp` 中声明的变量 `myName` 的值.

[KI] In a file, how to use some name defined in another file?

38. 在文件 `tools.cpp` 中定义一个函数 `get_my_name`, 该函数请求用户通过键盘输入一个名字, 并将用户输入保存在文件 `data.cpp` 中声明的变量 `myName`. 在主函数 `main()` 中调用该函数.

[KI] Input multiple words from the keyboard.

[Hint] You can decide the return type of this function.

39. 在文件 `tools.cpp` 中定义一个函数, 该函数可以将一个字符数组中保存的字符串进行逆序存放, 并在主函数中调用该函数.

40. 在文件 `tools.h` 定义一个内联函数，该函数返回两个整型参数中的较大值，并在主函数中调用该函数。
41. 在文件 `tools.cpp` 中定义一个函数，该函数基于循环结构返回一个整型数组中的最大值，并在主函数中调用该函数。

[hint] Define the function in `tools.cpp` and test it in the `main()` function in `hmkl.cpp`.

42. 在文件 `tools.cpp` 中定义一个函数，该函数基于递归结构返回一个整型数组中的最大值，并在主函数中调用该函数。

[hint] it is like the above task, but the difference is that it is recursive.

43. 在文件 `tools.cpp` 中定义一个函数 `sum_recursive()`，该函数包含一个非负整数 `n` 作为其参数，并基于递归结构计算从 0 到 `n` 的整数序列的和，并在主函数中调用该函数。
44. 在文件 `tools.cpp` 中定义一个函数 `sum_while()`，该函数包含一个非负整数 `n` 作为其参数，并基于 `while` 循环结构计算从 0 到 `n` 的整数序列的和，并在主函数中调用该函数。
45. 在文件 `tools.cpp` 中定义一个函数 `sum_for()`，该函数包含一个非负整数 `n` 作为其参数，并基于 `for` 循环结构计算从 0 到 `n` 的整数序列的和，并在主函数中调用该函数。

- Use some syntax of the `for` statement that is not allowed by C but allowed by C++.

46. 在文件 `tools.cpp` 中定义一个函数 `week_plans()` 并在主函数中调用该函数，该函数包含一个 `Weekday` 类型的参数 `day`，并为每一天打印输出一条不同的句子，例如

- if `day` is monday, it prints "Focus on study".
- if `day` is Friday, it prints "Have some fun".
- ...
- you have to use the `switch` statement to implement the function.

47. 在文件 `tools.cpp` 中定义一个函数 `hello` 并在主函数中调用该函数，该函数应具备如下功能：

- 该函数没有参数。
- 该函数打印输出字符串 `hello`，并紧接着输出变量 `myName` 中的值。
- 该函数通过其函数体中的局部变量记录其被调用的次数，并将该值作为函数返回值。

[hint] Can a local variable have a duration that is longer than a function call?

48. 在文件 `tools.cpp` 中定义一个函数 `grades()` 并在主函数中调用该函数，该函数包含一个整型参数 `n`，并且能够根据以下条件输出相应字符：

- 如果 $n \geq 90$ ，打印输出 A
- 如果 $80 \leq n < 90$ ，打印输出 B
- 如果 $70 \leq n < 80$ ，打印输出 C
- 如果 $60 \leq n < 70$ ，打印输出 D
- 否则，if $N < 60$ ，打印输出 F
- 请使用 `if - else` 语句编写该函数的功能。

49. 在文件 `tools.cpp` 中定义一个函数 `nine_roots()` 并在主函数中对其调用。该函数接收一个整型参数 `n` 作为参数，并基于 `cout` 表达式以九宫格形式打印输出 `n` 到 `n+8` 共九个数字的平方根。每一个平方根打印输出 3 位小数。该函数应完成下列功能要求：

- 正确设置 `cout` 的输出宽度和精度属性。
- 输出结束后，应重置 `cout` 的输出格式到函数调用前的状态。

[Hint] There is some sample program in Chapter 8 of "C++ Primer Plus (edition 6)"

For example, when `nine_roots(10)` is called, it will print the square roots from 10 to 18, as follows:

	one	two	three
1	3.162	3.316	3.464
2	3.605	3.741	3.872
3	4.000	4.123	4.242

50. 在文件 `tools.cpp` 中定义三个同名函数 `together`，并在主函数中对这些函数分别调用：

- Version 1: 接收两个整型参数，并返回两个参数的和。
- Version 2: 接收两个C格式的字符串参数 (即字符指针), 并以两个字符串的拼接内容作为返回值。
- Version 3: 接收两个C++格式的字符串参数(即String类型的对象)，并返回一个新的字符串对象，该字符串是两个参数字符串的拼接。

[KI] Function overloading

V. 作业提交

- 所有同学请独立完成本次作业，不允许共同完成和抄袭
- 通过理工大在线课堂提交所有作业代码
- 截止日期: 2023 October 31 11:59pm

VI. 参考文献

[1] C++ Primer Plus (edition 6), Stephen Prata, ISBN: 978-0321-77640-2, Pearson

[2] C++语言程序设计 (第5版), 郑莉, 董渊, ISBN: 978-7302-56691-5, 清华大学出版社