

实践7 随机字母问题

- 常规方法

```
#include<cstdlib>
```

```
#include<ctime>
```

```
srand(time(0));
```

```
chUp  ='A'+ rand()%26;
```

```
chLow = 'a'+ rand()%26;
```

```
if(rand()%2)    chMix ='A'+ rand()%26;
```

```
else           chMix ='a' + rand()%26;
```

实践7 随机字母问题

- 一种简单方案

```
char alphabet[53]={ “AB...Zab...z”      };
```

```
srand(time(0));
```

```
numUp   = rand()%26;
```

```
numLow  = 26+rand()%26;
```

```
numMix  = rand()%52;
```

```
cout<<alphabet[numXXX];
```

实践7小结

- 复杂任务的程序实现
 - 划分功能块，规划各部分代码的任务(东大①→南京站②→厦门站③→厦大)。
 - 细化任务(①：摆渡车→地铁→南京站)。
 - 各模块相对独立，减少交织，耦合。
- 结构化程序设计——“自顶向下(Top down)”
 - 算法(Algorithms)
 - 数据结构(Data Structure)

实践7小结

- 函数的**定义**(definition)与**声明**(declaration/prototype)
 - **不可**在函数内**定义(define)**函数，即函数不可以嵌套定义。
 - **不要**在函数内**声明(declare)**函数，此声明作用域在函数内。
 - **同一个函数**可多次声明，但不可以多次定义。
- 函数可以嵌套**调用(call)**
 - 如**割线法**函数中，嵌套调用**待求根方程**的函数。
 - 如求PI函数中，嵌套调用pow函数。
 - 如求一元二次方程函数中，嵌套调用sqrt函数。
 - 如计算器函数，嵌套调用 $+-\times\div$ 函数。

实践7小结

- 程序的可读性（对程序员友好）
 - 函数的返回
 - 字符 vs. ASCII码值
- 外部输入的三种方式的使用
 - `cin`, `cin.get`, `cin.getline`
 - 注意使用单字符输入函数`get`前清空输入缓冲区。
- 编程风格：命名，缩进与对齐！
 - 函数名也是一种标识符，同一作用域内函数名与变量名不可重复。

实践7小结

- 大型工程设计中，每个实现复杂功能的程序，都可以被当作一系列小功能片段的组合，每个小功能片段可以抽象为特定功能的函数。每次需要完成相同功能，如打印表头，计算公约数，就不需要重写代码，只要直接调用写好的函数。
- 函数功能的权衡(trade-off)
 - 过粗，一个函数几百行，几屏都翻不完——恐怖
 - 过细，每一行代码都被拆散成一个函数——痛苦

Tips: 起一个好的函数名减少程序员的痛苦。

20 ± 行的函数比较符合人脑思维内存容量。

实践7小结

- 函数的好处!
- 以计算GCD (the Greatest Common Divisor)为例!
 - GCD函数可以被其他代码调用，提高代码的可复用性(reusability)，功能改变时也便于扩展(extensibility)。
 - 把计算GCD的函数与main函数的其他代码分隔开，逻辑更清晰(simplicity)，代码更容易阅读(readability)。
 - 如果计算GCD的过程有错误，错误就局限在GCD函数内部，缩小调试范围，提高代码的可维护性(maintainability)可测试性(testability)。

实践8 函数提高

- 实验内容：
 - 函数的内联
 - 函数的重载
 - 带默认参数的函数

实验九，第1题，学习作用域的含义时，允许使用全局变量。

本次其他与函数相关实验中，均不允许使用全局变量传递参数或返回。

实践8：

- 定义和使用：内联函数、重载函数、带默认参数的函数。
- 掌握
 - 标识符的作用域，变量的存储类型，生命期。
 - 多文件组织方式的实践。
- 实践内容：实验九，课本习题3.11~3.14。

本周代码量较少较简单，尽量全部完成，**重难点在理解！**
熟练掌握并理解局部变量，静态变量的**作用域、生命期**。
熟练掌握并理解函数的**重载和默认参数**。
理解内联函数与常规函数调用的差异。

实践8：提交程序清单1

- 实验九-范例3，静态局部变量的理解。
- 要求：务必！！！！
 - 先读程序写出运行结果；
 - 后执行程序，检验你的分析结果。
- 目的：理解变量作用域和生命期。
 - 同名全局变量 vs. 局部变量(实验九.范例3)。
 - 局部变量 vs. 静态局部变量(实验九.范例3; 例3.10)。
 - 全局变量 vs. 静态局部变量(实验九.范例3)。

实践8：提交程序清单2

- 函数重载实践

- 实验九-5，不同数据类型输出函数：output()

- 要求：调用函数时，自适应地输出：整型数，实型数，字符串。
 - 要求：当参数为整型时，设置默认参数，使得函数在默认情况下以十进制方式输出(该形参可采用枚举类型定义)
 - 要求：当参数为实型时，设置默认参数，设定默认输出精度为6位。

- 熟练掌握理解重载函数的声明，定义，调用。
- 熟练掌握理解函数默认参数的使用，含义。

实践8：提交程序清单3

- 函数重载实践

- 课本习题3.13，求余函数：remainder()

- 要求：当参数为整型时，返回两数相除的余数；
 - 要求：当参数为实型时，返回四舍五入取整后两数相除的余数。
 - 思考：对实型数进行四舍五入round()，向下取整floor()，向上取整ceil()函数的实现方式。注意区分正负数的处理。

- 要求熟练理解重载函数的声明，定义，调用。

- 附加：当输入参数，一个是整型，一个是实型时，产生什么情况？分析原因。

实践8：提交程序清单4

内联函数实践，课本习题3.11~3.12。

要求：用函数实现自己的表头，程序在控制台输出表头，计算自己做的题号： $(\text{学号后二位} - 1) \% 2 + 11$

目的：理解内联函数相关语法与内联的含义。

Tips: 实际是否内联展开，与编译器的设置有关。

实践8：提交程序清单5*

多文件组织方式，实验九-2。

注意：各自用学号区分**头文件**。本题将提交两个文件：

Exp08_学号_area.cpp

Exp08_学号_area.h

注意：**定义**性质代码写在cpp文件中，**声明**性质代码写在h文件中。

Tips: 本次实验，main函数中为area函数的调用和测试代码。若为了简便，这些代码可以放在area.cpp文件内。也可以另定义一个源文件 Exp08_学号_areaMain.cpp，在其中定义main函数，在所建工程中同时包含两个源文件。

重要！！！！

提交的文件名格式：

Exp08_学号_实验名.cpp

注意:1) 下划线

2) 学号别漏掉

3) 各种名字(变量名, 文件名)的可读性!

4) 多文件实验要求提交两个文件, 程序源文件(.cpp文件)和头文件(.h文件)

附加题也用**Exp08_学号_实验名.cpp**格式提交!