小马加编信息学教案(十四)

C++函数(一)

一. 课程内容

1. 函数的概念与用途
2. 函数的定义
3. 函数的调用

二. 知识讲解

1. 函数的概念与用途

1.1 函数的概念与用途

> **一个 `C++` 程序无论大小，都由一个或者多个"函数"组成，而且其中必须有且只有一个函数 `main()` ，称之为 "主函数"**，由函数 `main()` 调用其他函数来完成程序的特定功能
> 当然，其他函数之间也可以按照规则互相调用。

> `C++` 中的函数由一段相对独立的代码组成，这段代码能实现某一项具体、独立、完整的功能。

> 函数在程序设计中的作用主要有两个，一是 **"代码重用"**；二是 **"问题分解"**

- **代码重用**是保证同一个函数可以被一个或多个函数调用任意多次，从而减少重复代码的编写。
- **问题分解**可以保证一个大的程序（或者说软件），按照模块化编程思想，由大化小，分解成若干个结构清晰、功能独立、调试方便的函数，甚至给若干人合作完成，从而提高开发效率。

1.2 常用库函数

> C++ 提供了很多常用的系统函数，如输入单个字符的函数 `getchar()` 等。但是有些函数，必须要加上相关头文件才能使用
> 例如 整数取绝对值的函数 `abs()` 、求算术平方根的函数 `sqrt()` 等

> C++ **常用数学库函数**

函数格式	含义	举例
int abs(int i)	返回整型参数 i 的绝对值	abs(−5)=5
double fabs(double x)	返回双精度参数 x 的绝对值	fabs(−2.5)=2.5
double ceil(double x)	返回不小于 x 的最小整数(上取整)	ceil(2.5)=3.0 ceil(−2.5)=−2.0
double floor(double x)	返回不大于 x 的最大整数(下取整)	floor(2.5)=2.0 floor(−2.5)=−3.0
double pow(double x,double y)	返回 x 的 y 次幂的值	pow(2,3)=8
double sqrt(double x)	返回 x 的平方根	sqrt(9)=3
double log(double x)	返回 ln(x)的值(以 e 为底)	log(2.71)≈1 log(8)/log(2)=3
double log10(double x)	返回 lg(x)的值(以 10 为底)	log10(100)=2

2. 函数的定义

2.1 函数的定义

> C++ 要求函数必须先定义、后使用。

> 定义函数，就是要说明函数的**返回值类型**、**函数名**、**函数参数**，以及**完成特定功能的语句组合（函数体）**。

> 定义函数的格式如下：

```
返回值类型 函数名（参数列表）{
    函数体
}
```

> 其中，第一行称为**函数头部**。**函数名**是标识这个函数的合法标识符。

> **返回值类型**是指一个函数结束后返回给调用者的一个“返回值”的数据类型。
> 有些函数的功能是执行一系列操作，而不返回任何值，**这种情况下，*返回值类型*是关键字void**。

> **参数列表**是当函数被调用时，调用者向函数传递的各种“参数”，此处的参数称为**形式参数**
> 参数列表包括**参数的数据类型和参数名**，参数是可选的，没有参数就是“无参”函数，但是**括号不能省略**。

> 大括号之间的部分称为**函数体**，主要包括变量说明语句、表达式语句等。
> **如果有返回值，则函数体内至少有一条语句" `return` 表达式"**。
> 在执行函数体的过程中，一旦遇到 `return` 语句，执行完就立刻退出函数，不再执行后续的语句。
> 无返回值函数不需要 `return` 语句。

> ex: 实现返回传入的形参 a, b 之和 $a + b$ 的函数

```
int add (int a, int b) {
    int c = a + b;
    return c;
}
```

2.2 函数的声明

> 函数声明由**函数返回类型、函数名和形参列表**组成。
> 形参列表必须包括形参类型，但是**不必对形参命名**

2.3 注意

> **如果函数的定义放在调用之后，则必须先声明再调用**
> **如果函数的定义在调用之前，则可不必声明**

> 函数不能**嵌套定义**
> 强调一点，C语言不允许函数嵌套定义；也就是说，**不能在一个函数中定义另外一个函数，必须在所有函数之外定义另外一个函数**。
> main() 也是一个函数定义，也不能在 main() 函数内部定义新函数。

3. 函数的调用

- **在程序中以任何方式对函数的使用**，都称为函数的调用。

> 函数调用是通过“函数名”进行的，一般格式为：
> 函数名（参数列表）

> 此处的**参数列表称为 "实际参数"** ，是传递给调用函数的，必须严格对应函数定义时函数头部的形式参数列表，包括**参数个数、参数顺序、数据类型**。
> **调用无参函数时参数列表可以没有，但括号不能省略**。如果参数列表包含多个参数，则各参数间用逗号隔开。

- 函数调用的方式

> 以函数在程序中出现的位置和形式来看，函数调用方式分为三种。
> (1) **函数调用作为一条独立语句，完成一件事情(一系列操作)，没有任何返回值**。例如：`print (n); doit(dep,total); input();`
> (2) **函数调用的结果作为表达式的一部分**。例如：`int t = compute(i,j) + i*j;`
> (3) **以实参形式出现在其他函数调用中**。例如：`number = min(sum(-5,100),n); num = max(max(a,b), c);`

- 阅读理解下面的程序内容，并尝试写出运行结果

```
#include  <bits/stdc++.h>
using namespace std;

void maxnum(int x,int y) {
    int w = x > y ? x : y;
    cout << w << endl;
}

int main() {
    int a = 5, b = 22;
    maxnum(a, b);    // 函数调用作为一条独立语句
    return 0;
}
```

```
#include  <bits/stdc++.h>
using namespace std;

int fac(int n) {
    int z = 1;
    for(int i = 1; i <= n; i++)
        z = z * i;
    return z;
}

int main() {
    int x = fac(5) + fac(4);    // 函数调用出现在表达式中
    cout << x << endl;
    return 0;
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;

int big(int x,int y);// 函数的提前声明

int main() {
    int x, y, z;
    cin >> x >> y >> z;
    cout << big(big(x, y), z) << endl;  // 函数调用的返回值又作为其他函数调用的实际参数
    return 0;
}

int big(int x, int y) { // 函数定义
    if(x > y)
        return x;
    else
        return y;
}
```

三. 经典例题

- (请使用函数完成以下各题)

1. 任性输出

> 请编写一个函数，该函数的作用是输出三句 `Hello xmjb` 。
> 并调用这个函数 n 次

> 输入格式
> 一行一个正整数 n ，表示调用次数

> 输出格式
> $3*n$行，每行都是一句 `Hello xmjb` 。

样例输入	样例输出
1	Hello xmjb Hello xmjb Hello xmjb

> 数据范围
> $1 \leq n \leq 1000$

2. 素数统计

> 输入一个正整数n,输出$1 \sim n$中的素数个数

> 输入格式
> 一行一个正整数n

> 输出格式
> 输出一个正整数，表示1 ~ n中的素数个数

样例输入	样例输出
10	4

3. 统计闰年

> 输入两个年份 x 和 y，统计并输出公元 x 年到公元 y 年之间的所有闰年数(包括 x 年和 y 年)。

> 输入格式
> 一行两个正整数表示 x 和 y，之间用一个空格隔开。

> 输出格式
> 一行一个正整数，表示公元 x 年到公元 y 年之间的所有闰年数。

样例输入	样例输出
2000 2004	2

> 数据范围
> $1 \leq x \leq y \leq 3000$

四. 提高巩固

1. 曼哈顿距离

> 平面直角坐标系中位于坐标$(x1，y1)$的 i 点与位于坐标$(x2，y2)$的 j 点的曼哈顿距离为$d(i，j) = |x1 - x2| + |y1 - y2|$。
> 请编程输入两个点的坐标，输出它们之间的曼哈顿距离。

> 输入格式
> 一行四个整数(100 以内)，分别表示两个点的坐标$(x1，y1)$和$(x2，y2)$

> 输出格式
> 一行一个整数，表示两个点之间的曼哈顿距离。

样例输入	样例输出
10 5 6 20	19

> 数据范围
> 输入的整数都在100以内

2. 数的分离

> 定义一函数 $digit(n,k)$ 分离出整数 n 从右边数第 k 个数字。如 $digit(2076,1)$ 等于 6，而 $digit(2076,5)$ 等于 0。
> `main` 函数输入 n 和 `k`，调用 $digit(n,k)$ 输出答案，
> 输入格式
> 一行两个整数分别表示 n 和 k，之间用一个空格隔开。

> 输出格式
> 一行一个整数，表示整数 n 从右边数第 k 个数字。

样例输入	样例输出
31859 3	8

> 数据范围
> n 在 $long\ long$ 范围内。

3. 回文数个数

> 输入一个正整数 n ，求 1 ~ n 之间“回文数”的个数。
> 回文数是指一个数倒过来和原数一样，如 12121、11、1221、1 是回文数，而 1231 不是回文数。

> 输入格式
> 一行一个正整数 n，$1 \leq n \leq 10000$。

> 输出格式
> 一行一个正整数，表示 1 ~ n 之间回文数的个数。

样例输入	样例输出
12	10