C++知识总结

C++简明教程，可以用来复习知识点、巩固基础知识、备忘提示等作用。

目录

1. c++程序结构

```
#include <iostream>
using namespace std;
int main(){
    //你的代码放在这里
    return 0;
}
```

按照惯例，用helloworld开启我们的c++之旅。

```
//包含头文件
#include <iostream>
//使用std命名空间，
using namespace std;
//程序开始的地方，main函数主函数
int main(){

    //输出Hello World
    cout<<"Hello World";

    //返回0
    return 0;
}
```

2. 注释的使用

```
/*
    c++ 支持两种格式的注释
        多行注释
        单行注释
*/
// 这个是单行注释
```

3. 标准输入和输出

> cin和cout定义在iostream头文件中，iostream 是 Input Output Stream 的缩写，意思是“输入输出流”。

3.1 cin输入

```
int a,b;
//cin与流提取运算符 >> 配合使用
//获取用户输入的值
cin>>a;
//也可以一次输入多个数，中间用空格隔开，或者每行一个数
cin>>a>>b;
```

3.2 cout输出

```
//cout与流插入运算符 << 结合使用
cout<<a;
//也可以直接输出多个内容
//endl(end of line) 一行输出结束，输出下一行
cout<<"结果是："<<a<<endl;

//输出格式设置
int n = 141;
//1) 分别以十六进制、十进制、八进制先后输出 n
cout << "1)" << hex << n << " " << dec << n << " " << oct << n << endl;
double x = 1234567.89, y = 12.34567;
//2)保留5位有效数字
cout << "2)" << setprecision(5) << x << " " << y << " " << endl;
//3)保留小数点后面5位
cout << "3)" << fixed << setprecision(5) << x << " " << y << endl;
//4)科学计数法输出，且保留小数点后面5位
cout << "4)" << scientific << setprecision(5) << x << " " << y << endl;
//5)非负数显示正号，输出宽度为12字符，宽度不足则用 * 填补
cout << "5)" << showpos << fixed << setw(12) << setfill('*') << 12.1 << endl;
//6)非负数不显示正号，输出宽度为12字符，宽度不足则右边用填充字符填充
cout << "6)" << noshowpos << setw(12) << left << 12.1 << endl;
//7)输出宽度为 12 字符，宽度不足则左边用填充字符填充
cout << "7)" << setw(12) << right << 12.1 << endl;
//8)宽度不足时，负号和数值分列左右，中间用填充字符填充
cout << "8)" << setw(12) << internal << -12.1 << endl;
cout << "9)" << 12.1 << endl;
```

3. 数据类型

类型	关键字
布尔型	bool

类型	关键字
字符型	char
整型	int
浮点型	float
双浮点型	double
无类型	void
宽字符型	wchar_t

3.1 布尔型

```
//布尔型（bool 类型）
//bool类型只有两个取值，true和false，true表示“真”,false表示“假”。
//一般用在条件判断上，例如if语句、switch语句、while语句等
bool flag=true;
cout<<flag;  //输出1

//如果需要输出true或false，需要进行转化
cout<<(flag?"true":"false");
```

3.2 整型

数据类型	数据类型说明符	缩写	大小	范围
整型	int	int	4	-2147483648至147483647
无符号整型	unsigned int	unsigned	4	0至4294967295
短整型	short int	short	2	-32768至32767
无符号短整型	unsigned short int	unsigned short	2	0 至 65535
长整型	long int	long	4	-2147483648至147483647

数据类型	数据类型说明符	缩写	大小	范围
长整型	long long int	long long	8	-9223372036854775808至9223372036854775807

```
//数值类型的极值与平台相关，不同平台可能不一样
//获取int类型所占字节数，最大值和最小值
//需要引入limits头文件，#include <limits>
cout << "int数据类型 \n" << "所占字节数：\t" << sizeof(short);
cout << "\n最大值：\t" << (numeric_limits<short>::max)();
cout << "\n最小值：\t" << (numeric_limits<short>::min)() << endl;

//整型运算时需要注意
cout<<5/9;       //0
cout<<5.0/9;     //0.555556
```

3.3 浮点型

数据类型	类型说明符	大小	范 围
单精度浮点数	float	4字节	数字介于 ±3.4E-38 和 ±3.4E38 之间
双精度浮点数	double	8字节	数字介于 ±1.7E-308 和 ±1.7E308 之间
高双精度浮点数	long double	8字节	数字介于 ±1.7E-308 和 ±1.7E308 之间

```
//浮点型
```

3.4 字符型

字符型只能存储一个字符，在计算机中以数字形式存储。

数据类型	类型说明符	大小	范围
字符型	char	1 个字节	-128 到 127 或者 0 到 255
无符号字符型	unsigned char	1 个字节	0 到 255
有符号字符型	signed char	1 个字节	-128 到 127

```
//字符型
char a='a';
cout<<a<<endl;                //输出a
cout<<a-32<<endl;             //输出65，字符型参加数学运算自动转换为数字型
cout<<char(a-32)<<endl;       //输出A

//如果是小写字母则转换为大写字母
char b;
cin>>b;
b=(b>='a'&&b<='z')?b-32:b;
cout<<b;
```

4. 变量和常量

4.1 变量

- 变量是存放数据的容器。
- 变量定义
    - 必须先定义再使用。
    - 变量定义时，系统会根据变量类型给变量开辟空间来存储数据。

```
//定义变量
//语法：数据类型 变量名;
int a;

//定义变量的时候赋值
float b=3.0;

//同时定义多个变量，但是只能同时定义同一种数据类型
double c,d=2.12;
bool flag;
char a;
```

数据类型	初始化默认值
int	0
char	'\0'
float	0
double	0
pointer	NULL

```
/*
    变量：全局变量和局部变量
*/

//1.全局变量
#include <iostream>
using namespace std;
//全局变量声明
int studentId=1;
int main(){
    cout<<"全局变量在全局范围内有效："<<studentId;
    return 0;
}

//2.局部变量
#include <iostream>
using namespace std;
int main(){
    //局部变量声明
    int studentId=1;
    cout<<"局部变量在局部范围内有效："<<studentId;
    return 0;
}
```

4.2 常量

- 常量就像是常规的变量，只不过常量的值在定义后不能进行修改。
- 常量可以是任何的基本数据类型，整型、浮点型、字符、字符串和布尔值。

定义常量

```
/*
```

```
    定义常量
        1. 使用#define 预处理器
        2. 使用const关键字
*/

//1. 使用#define 预处理器
#include <iostream>
using namespace std;
//定义常量
#define PI 3.1415926
int main(){
    float r,s;
    cin>>r;
    s=PI*r*r;
    cout<<s;
    return 0;
}

//2. 使用const关键字
const PI=3.1415926;
```

5. scanf和printf输入和输出

5.1 scanf使用

```
//--scanf("输入控制符", 输入参数);
//%d 输入控制符，表示输入一个整数
//&i 表示变量 i 的地址，&是取地址符
scanf("%d", &i);

//--scanf("非输入控制符+输入控制符", 输入参数);
//尽量不用非输入控制符，因为所有的非输入控制符都要"原样输入"
scanf("i = %d", &i);

//--当然，也可以一次输入多个变量
scanf("%d%c",&a,&b);
```

- 注意事项
  - 参数的个数一定要对应
  - 输入的数据类型一定要与所需要的数据类型一致
  - 在使用 scanf 之前最好使用 printf 提示输入

5.2 printf使用☆

```
/*
printf 几种使用方式
    直接输出字符串
    输出变量
    输出多个变量
    混合输出字符串和变量
*/
//1. 输出字符串
//printf("字符串\n");
//\n是转义符，代表换行输出，建议使用以提高用户输入体验
printf("Hello World!\n");

//2. 输出变量
//printf("输出控制符", 输出参数);
printf("%d\n", i);

//3. 输出多个变量
//printf("输出控制符1 输出控制符2", 输出参数1, 输出参数2);
//printf是原样输出
printf("%d %d\n", i, j);
printf("i = %d, j = %d\n", i, j);

//4. 混合输出字符串和变量
//根据编程需要或题目需求进行输出
//例如，要求输出"2018年不是闰年！",只需要原样复制到printf的头一个参数里，然后替换变量就行了。
printf("%d年不是闰年！",year);
```

5.3 格式控制字符串详解

```
%[flags][width][.prec][length]type
//参数翻译成中文
%[标志][最小宽度][.精度][类型长度]类型

//type 类型，唯一的必选参数
printf("%d",a);

//width 最小宽度，可选
printf("%10d",a);
//如果实际位数超过指定宽度，按实际位数输出
//如果实际位数少于指定宽度，则补空格或补零输出
printf("%6d",1000); //输出:  1000
printf("%06d",1000);//输出:001000

//.prec 精度，可选
printf("%.3f",12.3456);  //输出12.346

//flag 标志，可选
```

```
printf("%5d %-5d\n",100,100);        //默认右对齐，使用-左对齐,右边补空格
printf("%+d %+d\n",1000,-1000);      //输出正负号
printf("% d % d\n",1000,-1000);      //正号用空格替代，负号输出
printf("%x %#x %o %#o\n",25,25,25,25);  //十六进制、八进制输出，加#可以输出进制前缀
printf("%05d\n",100);                //加0，位数不足时补零

//length 类型长度，可选，指明统一数据类型的不同长度
```

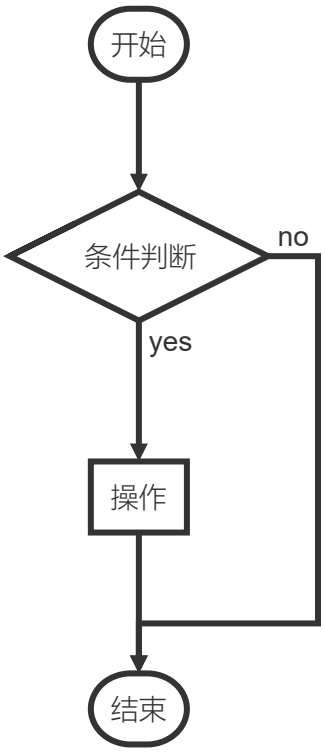
类型字符

格式控制符	格式控制符含义
%c	单个字符
%d	十进制整数(int)
%ld	十进制整数(long)
%u	无符号十进制数(DWORD)
%f	十进制浮点数(float)
%lf	十进制浮点数(double)，输出可以用%f，输入必须用%lf
%o	八进制数
%x	十六进制数(0x00000)
%s	字符串

6 判断

6.1 if语句 ☆

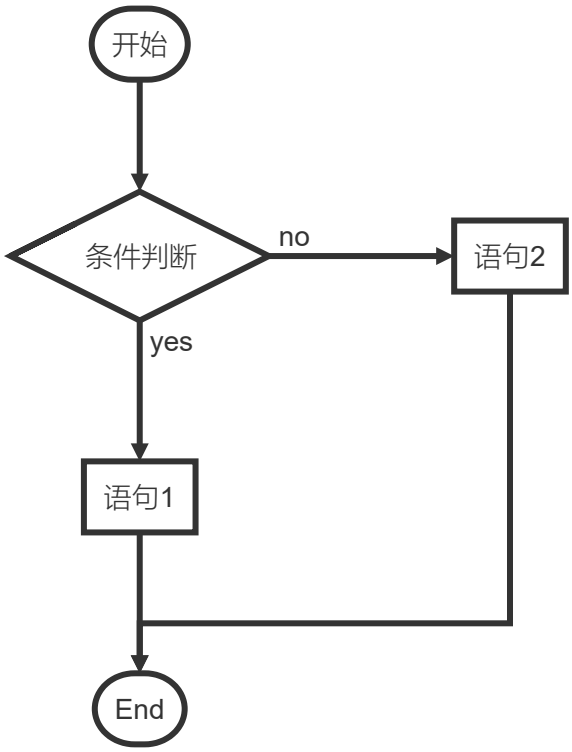
```
//if语句
//如果执行的操作只有一句，则
if(条件判断) 操作;
//一般情况下，需要执行的语句换行，tab缩进比较美观。
if(条件判断)
    操作;
//如果包括多个语句，需要加上大括号
if(条件判断){
    多个语句;
}
```




```
//判断偶数
int num=12;
if(num%2==0){
    printf("%d是个偶数",num);
}
```

6.2 if-else语句

```
if(条件判断){
    //条件判断为真
    语句1;
}else{
    //条件判断为假
    语句2;
}
```




```
//根据判断某年是不是闰年，分别输出xx年是闰年，xx年不是闰年。
if(year%4==0&&year%100!=0||year%400==0)
    printf("%d年是闰年");
else
    printf("%d年不是闰年");
```

6.3 条件表达式☆

```
/*
c++里提供了if-else的替代形式，条件运算符
格式如下：(条件判断)?表达式1:表达式2;
执行流程：
    先进行条件判断：条件为真，执行表达式1
                    条件为假，执行表达式2
*/
(year%4==0&&year%100!=0||year%400==0)?printf("%d年是闰年"):printf("%d年不是闰年");

//使用条件表达式的值
```

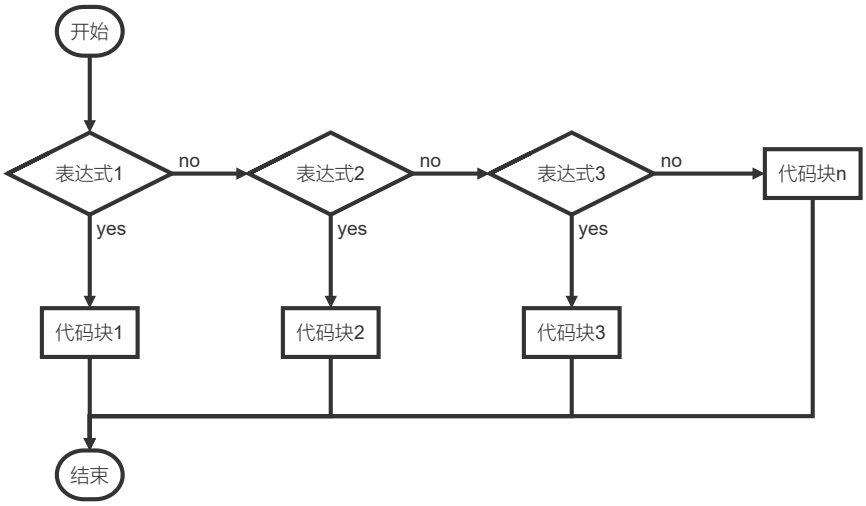
```
//判断数字大小,if-else语句
 if(a > b){
      max = a;
 }else{
      max = b;
 }
//判断数字大小，条件表达式
max=(a>b)?a:b;

//输入一个字符，判别它是否为大写字母，如果是，将它转换成小写字母；如果不是，不转换。
ch=(ch>='A' && ch<='Z')?(ch+32):ch;   //判别ch是否大写字母，是则转换

//条件运算符自右向左结合
a?b:c?d:e;
//等同于
a?b:(c?d:e);
```

6.4 if-else if语句

```
//if-else if语句用来实现多分支
if(表达式1){
       执行代码块1;
}else if(表达式2){
       执行代码块2;
}else if(表达式3){
       执行代码块3;
}
……
else{
       执行代码块n;
}
```




```
/**
小玉家的电费
    月用电量在150千瓦时及以下部分按每千瓦时0.4463元执行
    月用电量在151~400千瓦时的部分按每千瓦时0.4663元执行
    月用电量在401千瓦时及以上部分按每千瓦时0.5663元执行

根据题意模拟，分成三种情况。
    a<=150
    a>=151 && a<=400
    其他情况
*/
double a;//计算开double
scanf("%lf",&a);//输入,注意double型用%lf
if (a<=150){//判断即可
    printf("%.1lf",a*0.4463);
}
else if (a>=151 && a<=400){
    printf("%.1lf",150*0.4463+(a-150)*0.4663);
}
else {
    printf("%.1lf",150*0.4463+250*0.4663+(a-400)*0.5663);
}
```

6.5 switch语句

```
/*
输入成绩判断等级，分别为优秀、良好、中等、及格、不及格，其他情况输出"输错了"
```

```
*/
int s;
cin>>s;
switch(s/10){
    case 10:
    case 9:
        cout<<"优秀";
        break;
    case 8:
        cout<<"良好";
        break;
    case 7:
        cout<<"中等";
        break;
    case 6:
        cout<<"及格";
        break;
    case 5:
    case 4:
    case 3:
    case 2:
    case 1:
    case 0:
        cout<<"不及格";
        break;
    default:
        cout<<"输错了";
}
```