C语言基础知识学习经典入门

目录

[1 本课概要 1](#_Toc449785437)

[2 C语言入口方法 1](#_Toc449785438)

[3 C语言标准输入与输出 1](#_Toc449785439)

[4 C语言方法调用 2](#_Toc449785440)

[5 C语言宏定义 – 在编译阶段 2](#_Toc449785441)

[6 C语言宏方法定义 – 在编译阶段，无类型 3](#_Toc449785442)

[7 C语言条件判断 3](#_Toc449785443)

[8 C语言循环 4](#_Toc449785444)

[9 C语言结构体 – 存储数据的集合 – 结构体的赋值，直接是内存的拷贝 5](#_Toc449785445)

[10 C语言结构体指针 - 同一块内存 5](#_Toc449785446)

[11 C语言函数指针 6](#_Toc449785447)

[12 C语言typedef关键字 6](#_Toc449785448)

[13 C语言面向对象 – 思维方式 7](#_Toc449785449)

[14 C语言引入自定义头文件 9](#_Toc449785450)

[15 C语言字符串操作 - sprintf 10](#_Toc449785451)

[16 C语言文件操作 – fprintf、 fread 10](#_Toc449785452)

[17 C语言实现猜数字小游戏 11](#_Toc449785453)

# 1 本课概要

# 2 C语言入口方法

#include **<stdio.h>  
  
int** main(**int** argc, **const char** \* argv[]) { //argc从控制台传进来参数的数量，argv[]从控制台传入参数的值  
 **int** i;  
 **for** (i = 0; i < argc; i++){  
 printf(**"%s\n"**,argv[i]);  
 }  
  
 **return** 0; //C语言程序异常退出的代码，是返回给操作系统的，0表示正常退出  
}

C:\Users\Administrator\.CLion12\system\cmake\generated\45b5718a\45b5718a\Debug>**C01HelloC.exe** //当前应用程序的文件名

C01HelloC.exe //当前控制台所传入的第一个参数

C:\Users\Administrator\.CLion12\system\cmake\generated\45b5718a\45b5718a\Debug>C01HelloC.exe Hello C **"Hello World"**

C01HelloC.exe

Hello

C

Hello World

# 3 C语言标准输入与输出

#include **<stdio.h>  
  
int** main() {  
 printf(**"Hello %d"**,100);  
 printf(**"Hello %d\n"**,50);  
  
 puts(**"Hello C"**);  
 puts(**"Hello world"**);  
  
 **char** buf[100];  
 gets(buf);  
 puts(buf);  
  
 **int** a;  
 scanf(**"%d"**,&a);  
 printf(**"%d\n"**,a);  
  
 **return** 0;  
}

# 4 C语言方法调用

#include **<stdio.h>  
  
int** max(**int** a, **int** b){  
 **if** (a>b){  
 **return** a;  
 }**else**{  
 **return** b;  
 }  
}  
  
**int** main() {  
 printf(**"%d\n"**, max(10,12));  
  
 **return** 0;  
}

# 5 C语言宏定义 – 在编译阶段

#include **<stdio.h>**#define **MATH\_PI** 3.14 //在编译阶段进行替换，因此在运行过程中很快

**int** main() {  
 printf(**"%f\n"**, **MATH\_PI**);  
  
 **return** 0;  
}

# 6 C语言宏方法定义 – 在编译阶段，无类型

#include **<stdio.h>**#define **MAX**(**A**,**B**) \  
 **A**>**B**?**A**:**B  
  
int** main() {  
 printf(**"%d\n"**,**MAX**(20,10));  
  
 **return** 0;  
}

# 7 C语言条件判断

#include **<stdio.h>  
  
void** test(**int** score){  
 **if** (score >= 90){  
 printf(**"优秀\n"**);  
 } **else if** (score >= 80) {  
 printf(**"良好\n"**);  
 } **else if** (score >= 60) {  
 printf(**"及格\n"**);  
 } **else** {  
 printf(**"不及格\n"**);  
 }  
}  
  
**void** test1(**int** score){  
 **switch** (score/10) {  
 **case** 9:  
 puts(**"优秀"**);  
 **break**;  
 **case** 8:  
 puts(**"良好"**);  
 **break**;  
 **case** 7:  
 **case** 6:  
 puts(**"及格"**);  
 **break**;  
 **default**:  
 puts(**"不及格"**);  
 **break**;  
 }  
}  
  
**int** max(**int** a,**int** b){  
 **return** a>b?a:b;  
}  
  
**int** main() {  
 test(88);  
 test1(35);  
 printf(**"%d"**, max(20,10));  
  
 **return** 0;  
}

# 8 C语言循环

#include **<stdio.h>  
  
int** main() {  
 **int** i,j;  
  
 **for** (i = 0; i < 100; i++){  
 printf(**"%d\n"**,i);  
 }  
  
 **for** (j = 0; j < 100 ; printf(**"%d\n"**,j++)) {  
 puts(**"我先执行"**);  
 }  
  
 **int** k = 0;  
 **while** (k<100){  
 printf(**"%d\n"**,k);  
 ++k;  
 }  
  
 **int** a = 0;  
 **do** {  
 printf(**"%d\n"**,a);  
 a++;  
 }**while** (a<100);  
  
 **return** 0;  
}

一般情况，for循环用的最多，while用在死循环中。

# 9 C语言结构体 – 存储数据的集合 – 结构体的赋值，直接是内存的拷贝

#include **<stdio.h>  
  
struct** People{  
 **int** age;  
 **char**\* name;  
};  
  
**int** main() {  
 **struct** People p;  
 p.age = 10;  
 p.name = **"yuxiang"**;  
 printf(**"age:%d\n"**,p.age);  
  
 **struct** People p1 = p; //内存的拷贝  
 p.age = 20;  
 printf(**"age:%d\n"**,p1.age);  
  
 **return** 0;  
}

# 10 C语言结构体指针 - 同一块内存

#include **<stdio.h>**#include **<stdlib.h>  
  
struct** People {  
 **int** age;  
};  
  
**int** main() {  
 **struct** People \*p = malloc(**sizeof**(**struct** People));  
 p->age = 10;  
 printf(**"%d\n"**,p->age);  
  
 **struct** People \*p1 = p; //p1和p指向同一块内存  
 p1 -> age = 20;  
 printf(**"%d\n"**,p1->age);  
   
 free(p);  
  
 **return** 0;  
}

# 11 C语言函数指针

#include **<stdio.h>  
  
void** sayHello(){  
 printf(**"Hello C\n"**);  
}  
  
**int** main() {  
*// sayHello();* **void** (\*p)(); *//通过一个函数指针* p = sayHello; *//指向sayHello方法* p(); *//通过指针来执行方法* **return** 0;  
}

# 12 C语言typedef关键字

#include **<stdio.h>  
  
typedef struct** {  
 **int** age;  
} People;  
  
**typedef void** (\*Func)();  
  
**void** sayHello(){  
 printf(**"type define function\n"**);  
}  
  
**int** main() {  
 People p;  
 p.age = 10;  
 printf(**"%d\n"**,p.age);  
  
 Func f = sayHello;  
 f();  
  
 **return** 0;  
}

# 13 C语言面向对象 – 思维方式

代码一：

**typedef struct** {  
 **int** age;  
 **void** (\*sayHello)();  
} People;  
  
**void** PeopleSayHello(){  
 printf(**"Hello\n"**);  
}  
  
People \* PeopleCreate(){  
 **return** malloc(**sizeof**(People));  
}  
  
**void** PeopleDelete(People \*p){  
 free(p);  
}  
  
People\* PeopleInit(People\* p,**int** age){  
 p->age = age;  
 p->sayHello = PeopleSayHello;  
 **return** p;  
}  
  
**int** main() {  
 People \*p = PeopleInit(PeopleCreate(),20);  
 p->sayHello();  
 PeopleDelete(p);  
  
 **return** 0;  
}

代码二：

#include **<stdio.h>**#include **<stdlib.h>***//object begin>>>>>>>*#define **ObjectField**\  
 **void**(\*onDelete)(**void**\*);  
  
**typedef struct** Object {  
 **ObjectField**}Object;  
  
#define **ObjectCreate**(**TYPE**) malloc(**sizeof**(**TYPE**))  
  
#define **ObjectDelete**(**obj**) {\  
 **obj**->onDelete(**obj**);\  
 free(**obj**);\  
}  
  
**void** ObjectOnDelete(**void** \*obj){  
 printf(**"Object on delete\n"**);  
}  
  
Object\* ObjectInit(Object \*obj){  
 obj->onDelete = ObjectOnDelete;  
 **return** obj;  
}  
*//object end<<<<<<<<<***typedef struct** {  
 **ObjectField  
 int** age;  
 **void** (\*sayHello)();  
} People;  
  
**void** PeopleSayHello(){  
 printf(**"Hello\n"**);  
}  
  
**void** PeopleOnDelete(People \*p){  
 ObjectOnDelete(p);  
 printf(**"People on delete\n"**);  
}  
  
People\* PeopleInit(People\* p,**int** age){  
 ObjectInit((Object\*)p);  
 p->age = age;  
 p->sayHello = PeopleSayHello;  
 p->onDelete = PeopleOnDelete;  
 **return** p;  
}  
  
**int** main() {  
 People \*p = PeopleInit(**ObjectCreate**(People),20);  
 p->sayHello();  
 **ObjectDelete**(p);  
  
 **return** 0;  
}

# 14 C语言引入自定义头文件

//hello.h

#ifndef **C13INCLUDE\_HELLO\_H** //防止头文件重复引入#define **C13INCLUDE\_HELLO\_H  
void** sayHello();  
#endif *//C13INCLUDE\_HELLO\_H*

// hello.c

#include **<stdio.h>** //系统库，第三方库#include **"hello.h"** //自定义库 **void** sayHello(){  
 printf(**"Hello C"**);  
}

//main.c

#include **"hello.h"  
  
int** main() {  
 sayHello();  
 **return** 0;  
}

# 15 C语言字符串操作 - sprintf

#include **<stdio.h>**#include **<string.h>  
  
int** main() {  
 **char** buf[100];  
 memset(buf,0,100);  
 sprintf(buf,**"Hello %d,%f,%s"**,100,1.2,**"yuxiang"**); //将格式化的字符串打印到缓冲区中

printf(**"%s\n"**,buf);  
  
 **return** 0;  
}

# 16 C语言文件操作 – fprintf、 fread

#include **<stdio.h>  
  
  
int** main() {  
*// int i;  
// FILE \*f = fopen("data.txt","w");  
// if (f!=NULL) {  
// for (i=0; i<100; i++){  
// fprintf(f,"Hello %d\n",i);  
// }  
// }  
// fclose(f);* FILE \*f = fopen(**"data.txt"**,**"r"**);  
 fseek(f,0,**SEEK\_END**);  
 **long** size = ftell(f);  
 **char** buf[size+1];  
 fseek(f,0,**SEEK\_SET**);  
 fread(buf, **sizeof**(**unsigned char**),size,f);  
 buf[size] = **'\0'**;  
 fclose(f);  
  
 printf(**"%s"**,buf);  
  
 **return** 0;  
}

# 17 C语言实现猜数字小游戏

#include **<stdio.h>**#include **<time.h>**#include **<stdlib.h>  
  
  
int** main() {  
 srand((**int**)time(**NULL**)); *//使用当前系统的时间，指定一个随机的种子* **int** randNum = rand()%10;  
  
 printf(**"输入一个0到10的数值\n"**);  
  
 **int** userInput;  
 **while** (1){  
 scanf(**"%d"**,&userInput);  
  
 **if** (userInput<randNum) {  
 printf(**"数值过小\n"**);  
 }**else if** (userInput>randNum) {  
 printf(**"数值过大\n"**);  
 }**else**{  
 printf(**"正确\n"**);  
 **break**;  
 }  
 }  
  
 printf(**"exit\n"**);  
  
 **return** 0;  
}