

C++面试题

面向对象技术的基本概念是什么，三个基本特征是什么？

基本概念：类、对象、继承； 基本特征：封装、继承、多态。

初级

1. `#include<file.h>` 与 `#include "file.h"`的区别？

前者是从标准库路径寻找和引用file.h，而后者是从当前工作路径搜寻并引用 file.h。

2. `const` 有什么用途？

在C/C++中，（1）可以定义const常量，（2）修饰函数的返回值和形参；在C++中，还可以修饰函数的定义体，定义类的const成员函数。被const修饰的东西受到强制保护，可以预防意外的变动，提高了程序的健壮性。

3. `const`和`#define`有什么区别？

（1）const和#define都可以定义常量，但是const用途更广。

（2）const 常量有数据类型，而宏常量没有数据类型。编译器可以对前者进行类型安全检查。而对后者只进行字符替换，没有类型安全检查，并且在字符替换可能会产生意料不到的错误。

（3）有些集成化的调试工具可以对const 常量进行调试，但是不能对宏常量进行调试。

4. `sizeof`计算的是在栈中分配的内存大小。如何计算：

若指定了数组长度，则不看元素个数，总字节数=数组长度*`sizeof`（元素类型） 若没有指定长度，则按实际元素个数类确定

5. `sizeof`与`strlen`的区别？

（1）`sizeof`的返回值类型为`size_t`（`unsigned int`）；

(2) sizeof是运算符，而strlen是函数；

(3) sizeof可以用类型做参数，其参数可以是任意类型的或者是变量、函数，而strlen只能用char*做参数，且必须是以'\0'结尾；

(4) 数组作sizeof的参数时不会退化为指针，而传递给strlen是就退化为指针；

(5) sizeof是编译时的常量，而strlen要到运行时才会计算出来，且是字符串中字符的个数而不是内存大小；

6. 数组和指针的区别？

(1) 数组要么在全局数据区被创建，要么在栈上被创建；指针可以随时指向任意类型的内存块；

(2) 修改内容上的差别：

(3)用运算符sizeof 可以计算出数组的容量（字节数）。sizeof(p),p 为指针得到的是一个指针变量的字节数，而不是p 所指的内存容量。C++/C 语言没有办法知道指针所指的内存容量，除非在申请内存时记住它。注意当数组作为函数的参数进行传递时，该数组自动退化为同类型的指针。

7. 什么是多态？多态有什么作用？

多态就是将基类类型的指针或者引用指向派生类型的对象。多态通过虚函数机制实现。

多态的作用是接口重用。

8. C++空类默认有哪些成员函数？

默认构造函数、析构函数、复制构造函数、赋值函数

9. 头文件中的ifndef/define/endif有什么作用？

这是C++预编译头文件保护符，保证即使文件被多次包含，头文件也只定义一次。

10. 评价一下C/C++各自的特点

C语言是一种结构化语言，面向过程，基于算法和数据结构，所考虑的是

如何通过一个过程或者函数从输入得到输出； C++是面向对象，基于类、对象和继承，所考虑的是如何构造一个对象模型，让这个模型能够契合与之对应的问题，通过获取对象的状态信息得到输出或实现过程控制。

11. main函数执行之前会执行什么？执行之后还能执行代码吗？

全局对象的构造函数会在main函数之前执行； 可以，可以用_onexit 注册一个函数，它会在main 之后执行; 如果你需要加入一段在main退出后执行的代码，可以使用atexit()函数，注册一个函数。语法：

```
#include <stdlib.h>
#include <stdio.h>
int atexit(void (*function))(void));
void fn1( void ), fn2( void ), fn3( void );
int main( void )
{
    atexit(fn1);
    atexit( fn2 );
    printf( "This is executed first.\n" );
}
void fn1()
{
    printf( " This is\n" );
}
void fn2()
{
    printf( " executed next." );
}
结果：
This is executed first.
This is executed next.
```

12. 关键字static的作用是什么？

这个问题很少有人能回答完全。在C语言中，关键字static有三个明显的作用：

- 1). 在函数体，一个被声明为静态的变量在这一函数被调用过程中维持其值不变。
- 2). 在模块内（但在函数体外），一个被声明为静态的变量可以被模块内所用函数访问，但不能被模块外其它函数访问。它是一个本地的全局变量。

比如：在文件ClassA.cpp中声明 `static int g_id=0`，那么g_id就不能被ClassB.cpp定义的函数使用。

中级

1. 指针和引用的区别？

指针和引用都提供了间接操作对象的功能。

(1) 指针定义时可以不初始化，而引用在定义时就要初始化，和一个对象绑定，而且一经绑定，只要引用存在，就会一直保持和该对象的绑定；

(2) 赋值行为的差异：指针赋值是将指针重新指向另外一个对象，而引用赋值则是修改对象本身；

(3) 指针之间存在类型转换，而引用分const引用和非const应用，非const引用只能和同类型的对象绑定，const引用可以绑定到不同但相关类型的对象或者右值

2. C++中有malloc/free，为什么还有new/delete？

malloc/free是C/C++标准库函数，new/delete是C++运算符。他们都可以用于动态申请和释放内存。对于内置类型数据而言，二者没有多大区别。malloc申请内存的时候要制定分配内存的字节数，而且不会做初始化；new申请的时候有默认的初始化，同时可以指定初始化；

对于类类型的对象而言，用malloc/free无法满足要求的。对象在创建的时候要自动执行构造函数，消亡之前要调用析构函数。由于malloc/free是库函数而不是运算符，不在编译器控制之内，不能把执行构造函数和析构函数的任务强加给它，因此，C++还需要new/delete。

3. 继承层次中，为什么基类析构函数是虚函数？

编译器总是根据类型来调用类成员函数。但是一个派生类的指针可以安全地转化为一个基类的指针。这样删除一个基类的指针的时候，C++不管这个指针指向一个基类对象还是一个派生类的对象，调用的都是基类的析构函数而不是派生类的。如果你依赖于派生类的析构函数的代码来释放资源，而没有重载析构函数，那么会有资源泄漏。

4. 如果虚函数是有效的，那为什么不把所有函数设为虚函数？

不行。首先，虚函数是有代价的，由于每个虚函数的对象都要维护一个虚函数表，因此在使用虚函数的时候都会产生一定的系统开销，这是没有必要的。

5. 为什么构造函数不能为虚函数？

虚函数采用一种虚调用的方法。虚调用是一种可以在只有部分信息的情况下工作的机制。如果创建一个对象，则需要知道对象的准确类型，因此构造函数不能为虚函数。

6. C++如何阻止一个类被实例化？一般在什么时候将构造函数声明为private？

(1) 将类定义为抽象基类或者将构造函数声明为private；

(2) 不允许类外部创建类对象，只能在类内部创建对象

7. C++里面是不是所有的动作都是main()引起的？如果不是，请举例。

全局变量的初始化，就不是由main函数引起的

高级

8. 重载和覆盖有什么区别？

虚函数是基类希望派生类重新定义的函数，派生类重新定义基类虚函数的做法叫做覆盖；

重载就在允许在相同作用域中存在多个同名的函数，这些函数的参数表不同。重载的概念不属于面向对象编程，编译器根据函数不同的形参表对同名函数的名称做修饰，然后这些同名函数就成了不同的函数。

重载的确定是在编译时确定，是静态的；虚函数则是在运行时动态确定。

9. 请描述进程和线程的区别？

(1) 进程是程序的一次执行，线程是进程中的执行单元；

(2) 进程间是独立的，这表现在内存空间、上下文环境上，线程运行在进

程中；

(3) 一般来讲，进程无法突破进程边界存取其他进程内的存储空间；而同一进程所产生的线程共享内存空间；

(4) 同一进程中的两段代码不能同时执行，除非引入多线程。

10. 进程间如何通信？

信号、信号量、消息队列、共享内存

11. TCP和UDP有什么区别。

TCP——传输控制协议,提供的是面向连接、可靠的字节流服务。当客户和服务器彼此交换数据前，必须先双方在双方之间建立一个TCP连接，之后才能传输数据。TCP提供超时重发，丢弃重复数据，检验数据，流量控制等功能，保证数据能从一端传到另一端。

UDP——用户数据报协议，是一个简单的面向数据报的传输层协议。UDP不提供可靠性，它只是把应用程序传给IP层的数据报发送出去，但是并不能保证它们能到达目的地。由于UDP在传输数据报前不用在客户和服务端之间建立一个连接，且没有超时重发等机制，故而传输速度很快。

TCP协议和UDP协议的一些特性区别如下：

- 1.TCP协议在传送数据段的时候要给段标号；UDP 协议不需要。
- 2.TCP协议可靠；UDP协议不可靠。
- 3.TCP协议是面向连接；UDP协议采用无连接。
- 4.TCP协议负载较高,采用虚电路；UDP协议低负载。
- 5.TCP协议的发送方要确认接受方是否收到数据段(3次握手协议)。
- 6.TCP协议采用窗口技术和流控制。

12. 请讲述堆和栈的区别。

(1) 申请方式不同。栈上有系统自动分配和释放；堆上有程序员自己申请并指明大小；

(2) 栈是向低地址扩展的数据结构，大小很有限；堆是向高地址扩展，是不连续的内存区域，空间相对大且灵活；

(3) 栈由系统分配和释放速度快；堆由程序员控制，一般较慢，且容易产生碎片；

socket

socket调用库函数主要有：

创建套接字

Socket(*af,type,protocol*)

建立地址和套接字的联系

bind(*sockid, local aDDR, addrlen*)

服务器端侦听客户端的请求

listen(*Socketid ,quenlen*)

建立服务器/客户端的连接 (面向连接TCP)

客户端请求连接

CONnect(*sockid, destaddr, addrlen*)

服务器端等待从编号为Socketid的Socket上接收客户连接请求

newsockid=accept(*Socketid, Clientaddr, paddrlen*)

发送/接收数据

面向连接： **send(sockid, buff, buflen)**

recv()

面向无连接： **sendto(sockid,buff,...,addrlen)**

recvfrom()

释放套接字