

一、选择题

1、在一个类中可以对一个操作符进行 (D) 重载。

A、1种 B、2种以下 C、3种以下 D、多种

2、在C++中执行以下4条语句后输出rad值为： (C)

```
1 static int hot=200; //hot = 200
2 int &rad=hot;      // rad point to 200(hot)
3 hot = hot + 100;   //hot = 300
4 cout<< rad << endl; //now rad = 300
```

A、100 B、200 C、300 D、400

二、简答题

1. const关键字与宏定义的区别是什么？

- 宏定义是源码级别的抽象，在预处理时展开。const在编译运行时才使用。
- 宏定义没有类型，仅展开，不做任何类型检查。const具有具体的类型，在编译运行的时候会执行类型检查
- 宏定义仅仅是展开，有多少地方使用就有多少处展开。const会分配内存，可以是栈也可以是堆空间

2. malloc的底层实现是怎样的？free是怎么回收内存的？

- malloc基本的实现底存是一个内存空闲链表，当申请内存空间时，搜索内存空闲链表，找到适配的空闲内存空间，然后将空间分割成两个内存块，一个变成分配块（分配给用户），一个变成新的空闲块（闲置）。如果没有搜索到，那么就会用sbrk()才推进brk指针来申请内存空间。

即，将内存中的小块合并成需要的大小。如果实在没有找到空间则返回NULL。

- free回收内存的时候，直接将内存接到空闲块上

3. new/delete与malloc/free的区别与联系是什么？(面试常考)

- new/delete是C++关键字，需要编译器支持C++。malloc/free是库函数，需要头文件引入
- 使用new操作符申请内存分配时无须指定内存块的大小，编译器会根据类型信息自行计算。malloc则需要显式地指出所需内存的尺寸。
- 通常new操作符内存分配成功时，返回的是对象类型的指针，类型严格与对象匹配，无须进行类型转换。malloc内存分配成功则是返回void *，需要通过强制类型转换将void*指针转换成我们需要的类型。
- new操作符从自由存储区（free store）上为对象动态分配内存空间，而malloc函数从堆上动态分配内存。

4. 区分以下概念：内存泄漏、内存溢出、内存踩踏、野指针？(面试常考)

- 内存泄漏：堆区的内存由程序员分配和释放，如果只分配没有释放，则一定时间之后会造成堆空间被占满造成内存泄漏。
- 内存溢出：定义变量时候内存空间不够分配该类型的变量，此时为内存溢出。
- 内存踩踏：没懂
- 野指针：释放指针所指向的内存空间，和指针无关。释放之后需要指向NULL。

5.引用与指针的区别是什么？并且将"引用"作为函数参数有哪些特点？在什么时候需要使用"常引用"？

- 引用的本质是给变量取别名，指针是指向内存地址的一个特殊的变量
- 传引用和传指针的效果是一样的，但是传引用直接对实参进行操作，在内存中不会产生实参的副本

三、写出下面程序的运行结果。

```
1  #include <iostream>
2  using namespace std;
3  void f2(int &x, int &y)
4  {
5      int z = x;
6      x = y;
7      y = z;
8  }
9
10 void f3(int *x, int *y)
11 {
12     int z = *x;
13     *x = *y;
14     *y = z;
15 }
16
17 int main()
18 {
19     int x, y;
20     x = 10; y = 26;
21     cout << "x,y=" << x << ", " << y << endl;
22     f2(x, y);
23     cout << "x,y=" << x << ", " << y << endl;
24     f3(&x, &y);
25     cout << "x,y=" << x << ", " << y << endl;
26     x++; y--;
27     f2(y, x);
28     cout << "x,y=" << x << ", " << y << endl;
29     return 0;
30 }
31 /*result:
32 **x,y=10, 26
33 **x,y=26, 10
34 **x,y=10, 26
35 **x,y=25, 11
36 */
```