

2013-2014 C++期末考试

第一大题：

1 请说明 **new**、**delete**、**malloc**、**free** 分别在 **c++** 程序中的作用和使用场合，以及它们之间的不同之处

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**。

2. 结构 (**struct**) 和联合 (**union**) 两者有何区别？

(1). 结构和联合都是由多个不同的数据类型成员组成，但在任何同一时刻，联合中只存放了一个被选中的成员（所有成员共用一块地址空间），而结构的所有成员都存在（不同成员的存放地址不同）。

(2). 对于联合的不同成员赋值，将会对其它成员重写，原来成员的值就不存在了，而对于结构的成员赋值是互不影响的。

3. 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。

4. 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。但是一个衍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。

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。

第二大题：程序阅读理解

1.

```
char* get_SO2_String_1() {  
    char gas[] = "SO2";  
    return gas;  
}
```

```
char* get_SO2_String_2() {  
    char *gas = "SO2";  
    return gas;  
}
```

```
char* get_NO2_String_1() {  
    char gas[] = "NO2";  
    return gas;  
}
```

```
char* get_NO2_String_2() {  
    char *gas = "NO2";  
    return gas;  
}
```

```
void print_air_condition(int m[][2], int day, int gas_index) {  
    cout<< "\t" << day + 1;  
    cout<< "\t" << (gas_index == 0 ? get_SO2_String_1() : get_NO2_String_1());  
    cout<< "\t" << m[day][gas_index] << endl;  
}
```

```
void print_special_day(int m[][2], int day, int gas_index) {  
    cout<< "\t" << day + 1;  
    cout<< "\t" << (gas_index == 0 ? get_SO2_String_2() : get_NO2_String_2());  
    cout<< "\t" << *(m[day] + gas_index) << endl;  
}
```

```

int main() {
    cout<< "\tDay\t" << "Gas\t" << "Number" <<endl;
    int SO2_NO2[7][2] = {{95}, {102,133}, 163, 94, 89, 76, 133, 54, {69}, 76};
    print_air_condition(SO2_NO2, 0, 1);
    print_air_condition(SO2_NO2, 3, 0);
    print_air_condition(SO2_NO2, 5, 1);

    print_special_day(SO2_NO2, 2, 1);
    print_special_day(SO2_NO2, 4, 0);

    return 0;
}

```

2.

```

#include <Iostream>
using namespace std;
class Vehicle
{
public:
    virtual void run(int number = 10){
        cout << "we do not know how to run\n";
    }
    virtual void stop(){
        cout << "we do not know how to stop\n";
    }
    void announce(){
        cout << "this is a vehicle\n";
    }
};
class Car:public Vehicle
{
public:
    void run(int number = 60){
        cout << "driving at " << number << " km/h\n";
    }
    void stop(){
        cout << "brake to stop\n";
    }
    void announce(){
        cout << "this is a car\n";
    }
};
void main()
{

```

```

Vehicle v1,*v2;
Car c1,*c2;

v1.run();
c1.announce();

v2 = &c1;
v2->run();
v2->stop();
v2->announce();

c2 = (Car*)&v1;
c2->run();
c2->announce();
}

```

答案:

```

we do not know how to run
this is a car
driving at 10km/h
brake to stop
this is a vehicle
we do not know how to run
this is a car

```

3.

```

int get_int(){
    int i;
    if(!(cin>>i))
        throw Invalid_Input("Was expecting an integer");
    return i;
}

```

利用上述信息，实现 Invalid_Input 类，并写出 main 函数。要求 main 函数中，利用 try/catch 结构和 get_int() 函数，获取控制台输入的整数，并能 catch 非整数输入异常，输出异常信息。

答案

```

class Invalid_Input {
public:
    Invalid_Input(char * in_ptr) : msg_ptr(in_ptr) { }
    const char * const msg_ptr;
};

```

```

void main()
{
    try{
        get_int();
    }
    catch (Invalid_Input& except){
        cout << "Invalid input - " << except.msg_ptr << endl;
    }
}

```

4. 下面的递归程序，输出结果是如下模式的星状图形。请将程序空白部分补充完整：

```

*
* *
* * *
* * * *

int printEachLine(int j){
    if (_____)
        return 1;
    else{
        _____
        _____
        return 1;
    }
}

int print(int i){
    if (_____)
        return 1;
    else{
        _____
        _____
        return 1;
    }
}

/*****/

int main(){
    print(4); // recursion
    return 0;
}

```

答案：

```

int printEachLine(int j){
    if (j==0) return 1;
}

```

```

        else{
            printEachLine(j-1);
            cout<<"* ";
            return 1;
        }
    }

int print(int i){
    if (i==0)
        return 1;
    else{
        print(i-1);
        printEachLine(i);
        cout<<endl;
        return 1;
    }
}

```

第三大题：设计题

（1）第一部分

银行 ATM 随处可见，使人们的生活更便捷。ATM 系统提供的主要功能包括**登录、查询余额、存款、取款**。ATM 系统提供一系列的操作界面供用户使用这些功能，并将信息存入数据库。

请你给出 ATM 系统的设计类图，要求写清各个类的属性及方法，并说明每个类的作用。

（2）第二部分

我们的生活中有太多的银行，可是他们各自的 ATM 一般只对自己的用户提供最大的方便，跨行操作会收取手续费。

联合公司致力于提供一个跨银行的平台，方便人们的生活。他们最近与三家大银行——小工银行（ICBank）、小农银行（ABank）、小建银行（CBank）联合推出了一款新型 ATM，在这台 ATM 上，人们可以选择进入上述三家银行其中一家，之后的操作就仿佛在这家银行自有的 ATM 上进行一样。

该 ATM 系统提供的功能有**登录、查询余额、存款、取款**，这三家银行对这些功能的实现有各自不同的方法，对应有不同的界面以及数据库。

请你给出该 ATM 系统的设计类图，要求写清各个类的属性及方法，并说明每个类的作用。