C到C++ 1

一、bool类型

1、知识点介绍

1、bool是一种数据类型

2、取值为true(真)或false(假)

3、定义： bool isCollide=false;

4、内存大小占1字节

2、注意

1、该类型的取值不仅仅只能用true或false，还是可以是一个数值，如果是一个数值，这个值，遵循非0为真的原则。

二、内联函数

1、知识点介绍

1、内联函数是一个函数，是通过内存膨胀减少函数的跳转时间，以空间换时间。
1、内存膨胀：指的是代码区的代码变多，因为使用内联函数之后，那么这个函数的调用就不会在栈区申请内存来运行，也就是少了入栈出栈的时间，如果是一个普通函数在调用的时候会有入栈和出栈。

2、内联函数的定义

```
inline int getNum()
{
    return 1;
}
```

1、定义一个内联函数也就是在定义函数的时候在返回值类型前面加上一个inline修饰函数

3、内联函数的特点

1、他是一个函数，调用与普通函数一致
2、通过内存膨胀来减少函数的跳转
3、函数体代码过长，或者函数体中有循环，不建议使用内联
4、函数体是简单的赋值语句或者返回语句，而且使用频率高，建议使用内联

4、注:

内联函数在这里类似与宏替换，但是他们是有区别的，内联函数是一个函数，是在程序执行期间运行的，没有函数的入栈和出栈，宏替换是在预处理阶段执行的，本质是替换，内联函数有形参类型，有返回值类型和返回值，带参宏的形参是没有类型，没有返回值类型也没有返回值

三、引用

1、知识点介绍

1、引用是对一个变量或者对象取的别名

2、引用的定义

1、既然是对一个变量或者对象取别名，那就先得有变量或者对象，不能凭空取一个别名也就是定义引用必须初始化

```
int gameControllerNum=10;
定义别名的方式：  真名的类型 & 别名 = 真名;//别名也要符合名字规则
int & gcn=gameControllerNum;
```

gcm是gameControllerNum的别名，那么我们就可以用这个别名，用这个别名就相当于是在用真名，对别名修改同时别名所对应的真名的值也会改变。

```
gcn=20;
```

那么相对于的gameControllerNum的值也会别改成20

3、引用的使用

1、要看你引用的是什么类型的变量，如果是引用的整形变量那么就这个引用就像一个整形变量使用，如果引用的是一个指针就引用就像指针一样使用，是结构体变量，就像结构体变量一样使用

4、引用的注意事项

1、引用是别名，所以定义引用的时候别需要初始化
2、对引用的操作和对引用对应的变量的操作是完全等价的
3、& 在引用这里不是取地址，而是起到标志的作用
4、引用的类型必须和其所对应的变量的类型相同
5、引用不是定义新的变量或对象，因此不会为引用开辟新的空间内存

5、指针和引用的区别

在效率上是没有多大区别的。
1、引用是别名，不会被分配空间。指针是实体，会被分配空间
2、引用在定义时必须初始化，而且不是被改变，指针定义时可以不用初始化，也可以改变值
3、指针有多级指针，但是没有多级引用
4、引用是直接访问，指针是间接访问

四、函数重载

1、知识点介绍

1、函数重载指的是，在同一个项目中定义的函数名字可以重复，也就是可以有同名的函数，但是要有条件才能同名

2、构造函数重载的条件

```
1、函数名必须一致
2、函数的参数列表不同
   函数的参数列表不同：
        Ⅰ、参数列表个数不同
        Ⅱ、相对应位置的类型不同
 注意：重载函数重载满足上述条件就可以了，函数的返回值不同不在条件内容
```

1、以下这两个函数不满足函数重载，那么这样定义就会报错

```
int fun(){}
void fun(){}
```

2、以下这两个函数满足函数重载，参数列表个数不同，那么在调用的函数的就像函数调用一样，就比如有两个参的fun函数那么就是传2个参调用，要要调用下面一个参的函数，就穿一个对应的参，和函数调用是一致的

```
void fun(int a,int b){}
void fun(int a){}
void fun(int a,double b,char c){}
```

3、以下这两个函数虽然参数个数一样，但是参数对应位置类型不同，也满足函数重载 ，在调用就穿对应的参数的调用就好了

```
void fun(int a,int b){}
void fun(float a,float b){}
```

4、函数重载容易导致的二义性及解决

```
int fun(int a){}
int fun(float a){}
```

1、从函数重载的重载条件来看，构成重载，编译不会有问题，但在调用时，如果调用方式如：fun(2.3); 这样由于函数的实参为double，而该函数并无该类型的重载，在编译时会自动将double类型强转去匹配函数，这时double可以强转为int，也可以强转为float，这样就会出现二义性的问题。

5、解决方法：

1、加入新的重载函数 int fun(double a){}

2、明确调用时实参强转类型 如：fun((int)2.3);或者传一个不会进行转换的参数进去

注：编译器总是会帮助我们把基本数据类型大的转为小的

五、函数的参数缺省

1、知识点介绍

```
1、顾名思义，在声明函数的某个参数的时候为之指定一个默认值，在调用该函数的时候如果采用默认值，就无须指定该参数。
```

2、函数的缺省参数定义

函数参数的缺省定义：

```
  void fun(int a,int b=2){}
```

函数调用：

```
  fun(1);
  fun(1,3);
```

这个函数的形参有缺省的值了，那么就可以不用给它传参了，不给它传参，那么形参用的值就是这里默认的值，如果是传参了，就是用的传参了的值

3、函数的缺省参数注意事项

1、缺省参数只能由后往前依次缺省

2、缺省值必须是常量

3、函数如果只有定义，需在定义的参数列表中指明缺省参数。如果函数有声明和定义，则缺省参数只需在声明只指明即可。

```
void myFun(int a,int b = 0);

void myFun(int a,int b){}
```

4、函数的缺省参数与函数重载产生的二义性，及解决方法

```
int myAdd(int a,int b,int c = 0);

int myAdd(int a,int b);
```

分析：

上述两个函数满足函数重载，编译可通过，在调用第2个函数时就会出现二义性问题。也就是第1个函数也是满足调用条件，第2个函数同样满足。只有调用myAdd函数并且给出3个实参才不会出现二义性。

解决办法：

1）、把第1个函数的缺省参数去掉

2）、不实现函数重载

作业

1、实现功能，通过调用fun函数，能完成int+int double+double类型的变量，并返回他们的值，输出显示

项目打包发给2829114166(QQ邮箱)
格式：期数+名字+作业名字（例：2012-易木-第一节课作业）