**1. 一个C++应用程序，由三个cpp文件和若干头文件组成,cpp 文件分别为MyMain.cpp, func1.cpp,func2.cpp,**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**//MyMain.cpp 为**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <iostream.h>

int main(int argc,char \* argv[])

{

cout<<g2(10)<<endl;

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**//Func1.cpp 为**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int f1(int num)

{

return g1(num\*2);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**//Func2.cpp 为**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int g1(int num)

{

return num\*num;

}

int g2(int num)

{

return f1(num\*3);

}

**1）请添加必要的头文件，使此程序能编译通过并执行，理解包含警戒的作用。**

**2）如果main,f1,g1,g2或更多的函数之间有更为复杂的调用关系，头文件一般按怎样的规律写呢？**

**3）请总结出cpp文件与.h文件的关系，以及头文件中存放的内容。**

**4）若main,f1,g1,g2函数中都用到常量YEARDAYS(365),和WON(1),TUE(2),WED(3),THU(4),FRI(5),STA(6),SUN(7)**

**如何写头文件？**

**2.以max函数为例，实现不同参数传递方式（传值/传引用），不同参数常量性，以及不同的返回值类型（值返回，引用返回，常量引用返回）等情况下,max函数的参数虚实结合和max的各种使用和它们间的异同。注意参数不同形式时，哪些形式的调**用是可以的？哪些算重载函数？哪些是完全等价的？

int max(int,int);

int max(const int, const int);

const int max(int,int);

int max(int&,int&);

int& max(int&,int&);

int& max(const int&,const int&);

const int& max(int&,int&);

const int& max(const int&,const int&);

...

**Main函数中，可能有如下的调用：**

int retValue01 = max(1,2);

int retValue02 = max(a,b);

max(a,3) = 5;

max(a,b)=max(c,d );

**等形式。**

**3.练习类的定义与多文件实现，类A的定义在头文件a.h中，**

**class A {**

public:

void Display() const

{

cout<<"The argment is NULL"<<endl;

}

void Display(const char \* str) const

{

cout<<"The argment is "<<str<<endl;

}

void OtherFunc()

{

}

};

**main函数写在mainApp.cpp文件中，**

int main(int argc,char \* argv[])

{

A aA;

if (argc>1)

{

aA.Display(argv[1]);

}

else

{

aA.Display();

}

cout<<sizeof(A)<<endl;

return 0;

}

a)查看上述程序的执行结果。理解命令行参数、对象的大小、函数重载。

b)现将Display函数分别改成以inline形式和纯外联实现，试一试。理解内联/外联实现 在编程中的注意事项。

c)试一试在a.h中，写或不写包含警戒，程序是否正确执行。

d)给类A增加一个整型的**类变量aStaticInt和一个类方法**，在main函数中调用该方法。 掌握类变量的声明和定义。进一步，试一试，**类变量是一个数组呢？**

e)现再增**加一个类B，并放入b.h中**

**class B**

{

public:

void Func()

{

cout<<"This is member function Func()"<<endl;

}

private:

A aA;

};

并在主函数中调用B中的成员函数Func.

f)针对e),重新按c)中的要求试一试。若将Func的实现改成外联实现，按c)中的要求,再试一试。

**准确理解包含警戒的作用，以及各cpp文件独立编译的含义。**

g)将e)中的B改成：

**class B**

{

public:

B(A& aA) { pA = &aA; }

void Func()

{

cout<<"This is member function Func()"<<endl;

}

void NewFunc()

{

pA->OtherFunc();

}

private:

A \*pA;

};

试一试，让程序编译通过并执行。

h)将B中的NewFunc函数以外联的形式实现，b.h可以怎么写？理解class A;与#include "a.h"的含义差别，理解外联实现的好处。

**4.练习将实际生活中的事物，抽象成类，并给出类的设计**(找出数据成员和成员函数并用C++的类表示出来即可)

1）笔、钢笔

2）文件、目录

3）打印机、显示器

4）太阳、月亮

**5．练习读懂类的代码，练习使用已有的类。**

下面的TRandom类是**用线性调和算法**，实现的一个**伪随机数发生器**,仔细阅读代码，理解其含义。

**线性调和算法的基本原理是**：对于给定的N、M和K，任给一个X，不断用X=((N\*X+M) mod K )进行迭代计算，那么得到的无穷多个x值，近似于在（０，K）上的均匀分布．其中N,M,K为较大的数(N最好是素数)。

#include <limits.h> //声明INT\_MAX和ULONG\_MAX常量

#include <windows.h> //声明GetTickCount()函数,其返回从开机到现在经过的毫秒数

//#include <stdlib.h> //声明rand和srand函数

//#include <time.h> //声明time函数

class TRandom

{

public:

//缺省使用系统时间(开机后经过的毫秒数)为种子

TRandom (long seed=0) { mSeed=(seed?seed: GetTickCount()); }

//也可以定义自己的种子

void Seed(long seed=0) { mSeed=(seed?seed: GetTickCount( )); }

//取得一个随机的整数

int Integer() { return Next();}

//取得一个在指定范围内的随机整数

int Integer(int min,int max) { return min+Next()%(max-min+1);}

//取得一个随机的（0到1之间的）小数

double Real() {return double(Next())/double(INT\_MAX);}

private:

//使用调和算法

void Change() {mSeed=(314159265\*mSeed+13579)%ULONG\_MAX;}

//取得伪随机数发生器的随机数序列中的下一个随机整数

int Next()

{

int loops=mSeed%3;

for (int i=0;i<=loops;i++)

Change ();

return int(mSeed/2);

}

//随机数发生器的种子

unsigned long mSeed;

};

现有一个大小为54的整型数组，其元素分别为0,1,2,…,53.

１) 注意类中的public和private, 理解访问控制的作用．

２）请利用给出的TRandom类,**完成此数组的”洗牌”过程**，**并输出洗牌后的顺序**。写出相应的完整C++程序。

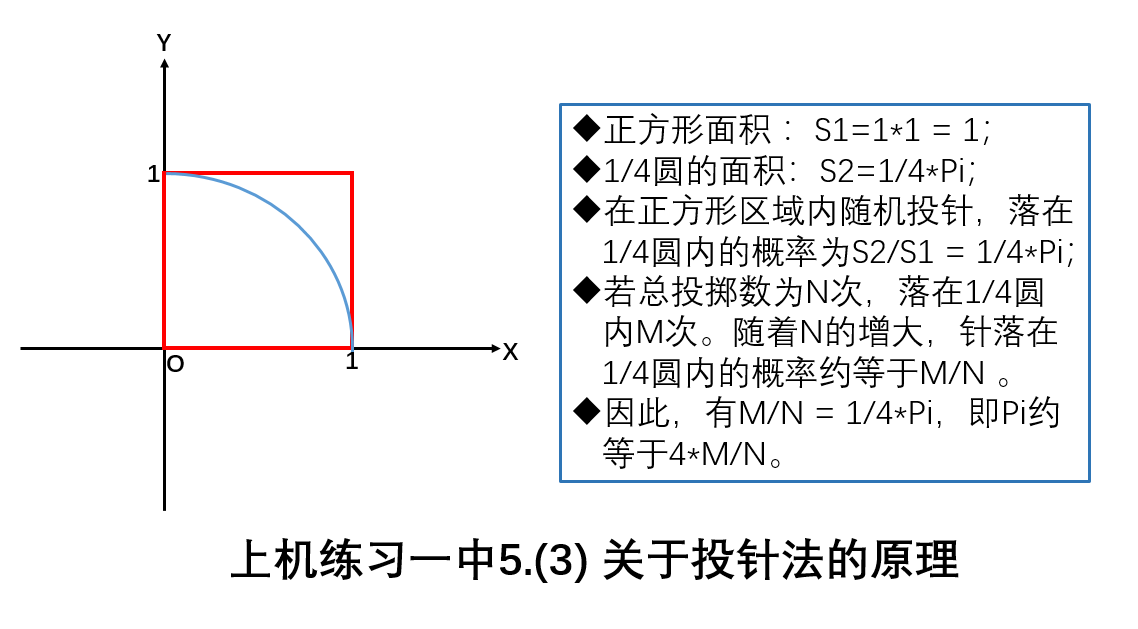
思考：用TRandom类，洗牌后,54张牌的排列大约多少种可能?现实中54张可有多少种排列方法?

3)利用给出的TRandom，使用投针法，估算PI值。

投针法：在半径为1的1/4圆内，随机投针，若落在圆内，则圆内计数加1；

这样，若总投掷数为N次，落在圆内M次，随着N的增大，

M/N约等于PI/4,即PI约等于4\*M/N.



6．一个骰子(dice)有6个面，各面的点数分别为1，2，3，4，5，6，但各面之间的相对位置是固定的。

请实现Dice类，其主要完成

1) 模拟掷骰子过程（Cast），并返回掷得到点数；

2）报告当前各面的点数。

3) 模拟一次掷两个或多个骰子，并返回得到的总点数。

实现该类时，取得随机数可使用系统函数srand和rand，也可使用上例中的TRandom类。