

C++ 程序设计 实验指导与课程设计

郭有强 王一宾 张怡文 叶家鸣 侯传宇 编著

安徽大学出版社

内 容 简 介

本教材是安徽省高等学校“十一五”省级规划教材——《C++程序设计》的配套实验教材。

本书自成体系,是《C++程序设计》的配套实验及课程设计指导教材,也可与其他相关教材配套使用。

“C++程序设计”是一门实践性很强的课程,上机实验和课程设计是本课程不可缺少的实践环节。

本教材给出了11个基础实验、8个课程设计,基本覆盖了C++程序设计的主要知识点;给出了4套模拟试题及相关解答。附录部分介绍了常见编译和连接错误、程序调试方法和技巧,以帮助读者上机练习;最后给出了实验报告参考样本和课程设计报告参考样本,以规范文档格式。

本教材中所有的程序都在 Visual C++ 6.0 开发环境中测试通过。

图书在版编目(CIP)数据

C++程序设计实验指导与课程设计 / 郭有强,王一宾,
张怡文,叶家鸣,侯传宇编著. —合肥:安徽大学出版社,
2008.5

ISBN 978-7-81110-454-7

I. C… II. ①郭…②王…③张…④叶…⑤侯 III. C语言
—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 061812 号

C++程序设计实验指导与课程设计

郭有强 王一宾 张怡文 叶家鸣 侯传宇 编著

出版发行	安徽大学出版社 (合肥市肥西路3号 邮编 230039)	经 销	新华书店
联系电话	编辑室 0551-5106428 发行部 0551-5108397	印 刷	合肥创新印务有限公司
E-mail	ahdxchps@mail.hf.ah.cn	开 本	787×1092 1/16
责任编辑	李镜平	印 张	7.875
特约编辑	罗 耀	字 数	192千
封面设计	孟献辉	版 次	2008年6月第1版
		印 次	2008年6月第1次印刷

ISBN 978-7-81110-454-7

定价 12.00 元

如有影响阅读的印装质量问题,请与出版社发行部联系调换

前 言

Foreword Foreword Foreword Foreword

本教材是安徽省高等学校“十一五”省级规划教材——《C++程序设计》的配套实验教材。

“C++程序设计”是一门实践性很强的课程,上机实验和课程设计是本课程不可缺少的实践环节。上机实验的目的是使学生熟悉面向对象语言 C++ 编程的思路及解题的全过程,加深对 C++ 语言的理解,得到面向对象程序设计基本方法和技巧的训练,从而巩固和深化所学的知识,真正能用 C++ 语言这个强有力的编程工具去解决实际问题。同时,通过上机实验及其随后的课程设计,期望学生能够熟练掌握 C++ 集成开发环境的使用。

本实验教材共分为 4 个部分:第 1 部分给出了 11 个基础实验,基本覆盖了 C++ 程序设计的主要知识点,每个实验都包含“实验目的”、“实验准备”、“实验内容”和“思考与练习”,并给出了基础实验部分“思考与练习”的相关解答。第 2 部分给出了 2 个课程设计样例(基于 C++ 控制台的应用程序、基于 MFC 的 Windows 编程应用)和 6 个课程设计题目(供参考使用),通过综合训练,期望读者能够对面向对象编程思想有进一步理解,进而提升实际应用编程能力。第 3 部分给出了 4 套模拟试题及解答。第 4 部分介绍了常见编译和连接错误、程序调试方法和技巧、标准 ASCII 码表,以帮助读者上机练习;还给出了实验报告参考样本和课程设计报告参考样本,以规范文档格式。

本书内容丰富,结构紧凑,选题典型丰富,注重操作步骤及细节,具有很强的可操作性。

实验的软件环境为 Visual C++、Turbo C++ 3.0 或 Borland C++。教材中所有的程序都在 Visual C++ 6.0 开发环境中测试通过。

本教材是《C++程序设计》的配套实验及课程设计指导教材。由于本书自成体系,也可与其他相关教材配套使用。全部例题的源代码放在网站 www.bbxy.edu.cn 上,供教师下载参考。

参加本书编写和文字工作的还有王一宾、张怡文、叶家鸣、侯传宇,在此深表感谢。

感谢读者选择使用本书,对书中内容存在的问题,敬请读者批评并提出修改建议,作者将不胜感激。在使用本书时如遇到什么问题需要与作者商榷,或想索取其他相关资料,请与作者联系。

电子邮件地址:bbxyguo@163.com.

郭有强

2008 年 5 月

目 次

Catalog Catalog Catalog Catalog

第 1 部分 基础实验

实验 1	Visual C++ 6.0 集成开发环境和简单 C++ 程序.....	(1)
实验 2	C++ 对 C 语言的扩充(1)	(6)
实验 3	C++ 对 C 语言的扩充(2)	(9)
实验 4	类和对象(1)	(12)
实验 5	类和对象(2)	(17)
实验 6	继承与派生	(20)
实验 7	多态性与虚函数	(24)
实验 8	运算符重载与类型转换	(29)
实验 9	类模板	(32)
实验 10	I/O 流	(36)
实验 11	利用 MFC 开发 Windows 应用程序	(39)
	【思考与练习】参考答案	(42)

第 2 部分 课程设计

1	概 述	(52)
2	总体要求	(52)
2.1	系统分析与系统设计	(52)
2.2	详细设计与编码	(52)
2.3	上机调试和测试	(53)
2.4	课程设计报告	(53)
3	课程设计样例	(54)
	课程设计 1	(54)
	课程设计 2	(61)

4	课程设计题目	(76)
	题目 1 模拟计算器程序	(76)
	题目 2 排课程序	(77)
	题目 3 图书馆管理系统	(77)
	题目 4 有理数运算	(78)
	题目 5 银行账户管理程序	(79)
	题目 6 水电煤气管理系统	(79)

第 3 部分 模拟试题

	模拟试题(1)	(81)
	模拟试题(2)	(84)
	模拟试题(3)	(90)
	模拟试题(4)	(96)
	参考答案	(99)

第 4 部分 附 录

	附录 A 常见编译、连接错误	(110)
	附录 B 程序调试方法和技巧	(114)
	附录 C 标准 ASCII 码表	(117)
	附录 D 实验报告参考样本	(118)
	附录 E 课程设计报告参考样本	(119)

第 1 部分 基础实验

实验 1 Visual C++ 6.0 集成开发环境和简单 C++ 程序

【实验目的】

1. 掌握使用 VC++ 集成开发环境开发程序的过程, 熟悉常用的功能菜单命令, 学习使用 VC++ 环境的帮助。
2. 学习并理解简单的 C++ 程序结构。

【实验准备】

1. 了解 C++ 程序的基本结构。
2. 了解简单的程序输入输出流。

【实验内容】

1. 初步熟悉 C++ 语言的上机开发环境。
2. 编程输出: This is my first practice !
3. 使用 Visual C++ 6.0 集成环境来编辑、编译并运行下面简单的 C++ 程序:

```
#include <iostream. h>
int max(int,int);          //声明自定义函数
void main()                //主函数
{
    int a,b,c;
    cout<<"input two number:\n";
    cin>>a>>b;
    c=max(a,b);           //调用 max 函数,将得到的值赋给 C
    cout<<"max="<<c<<endl;
}
int max(int x,int y)      //定义 max 函数,函数值为整型,形式参数 x,y 为整型
{
    int z;                //max 函数中的声明部分,定义本函数中用到的变量 z 为整型
```

```

if(x > y)
    z=x;
else
    z=y;
return(z);          //将 z 的值返回,通过 max 带回调用处
}

```

【实验步骤】

1. 启动集成开发环境

选择菜单【开始 | 程序 | Microsoft Visual Studio 6.0 | Microsoft Visual C++ 6.0】，进入 Visual C++ 6.0 用户界面，如图 1.1 所示。

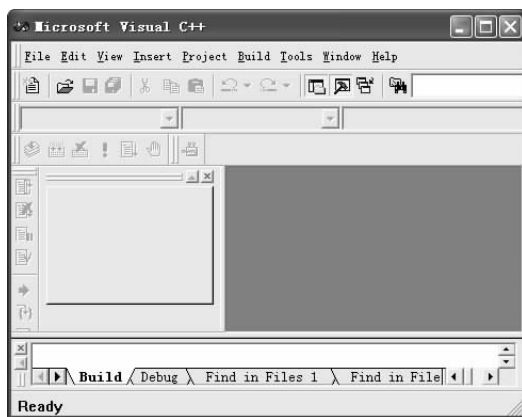


图 1.1 Visual C++ 6.0 用户界面

2. 创建新工程

(1) 选择【File】菜单下的【New】菜单项，出现“New”对话框，选择【Projects】标签，如图 1.2 所示。

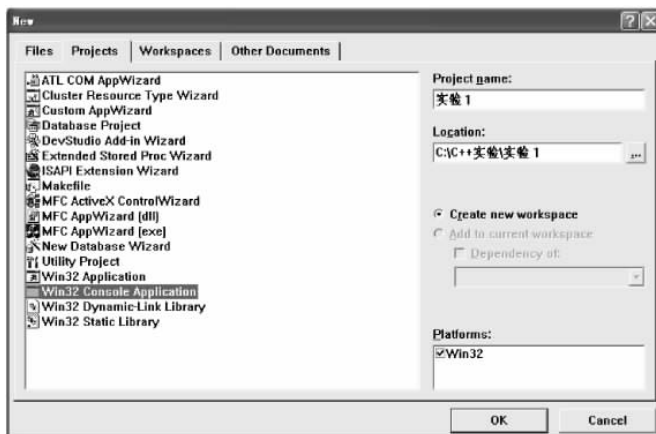


图 1.2 “New”对话框

(2) 在列表栏中,选择“Win32 Console Application”(Win32 控制台应用程序)。在右侧的 Projects name 编辑框中输入工程的名称(如“实验 1”)。在 Location 编辑框中指定该工程文件的保存路径,如图 1.2 中所示,然后单击【OK】按钮。

(3) 在“Win32 Console Application—Step 1 of 1”对话框中选择 An empty project 项。然后单击【Finish】按钮,如图 1.3 所示。

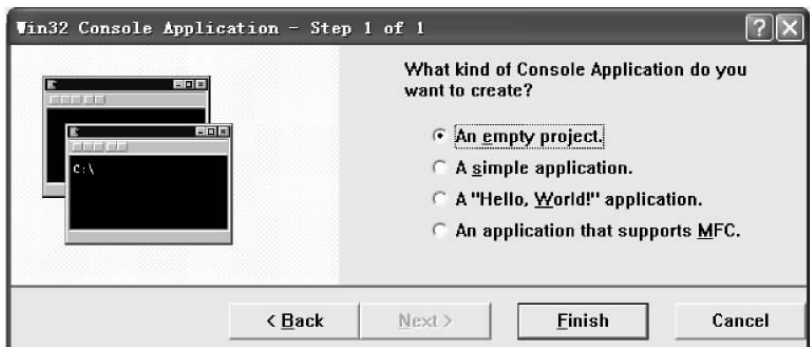


图 1.3 “Win32 Console Application—Step 1 of 1”对话框

(4) 在“New Project Information”对话框中单击【OK】按钮,完成工程创建过程。

3. 创建 C++ 源程序文件

(1) 选择【File】菜单下的【New】,显示“New”对话框,选择“Files”标签页面,如图 1.4 所示。在列表栏中,选择“C++ Source File”,然后在右边的文件输入框中输入源程序的文件名。

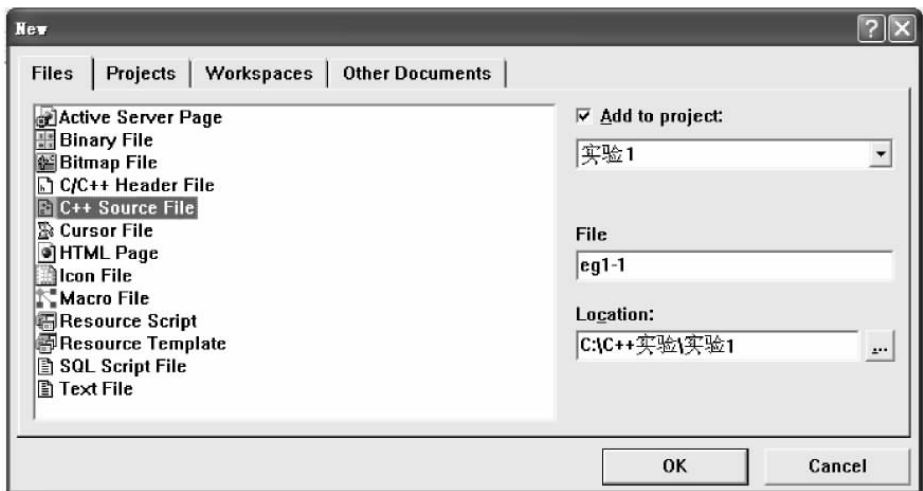


图 1.4 “New”窗口

(2) 单击【OK】按钮,则创建完成了一个源程序文件,并出现代码编辑窗口,如图 1.5 所示。

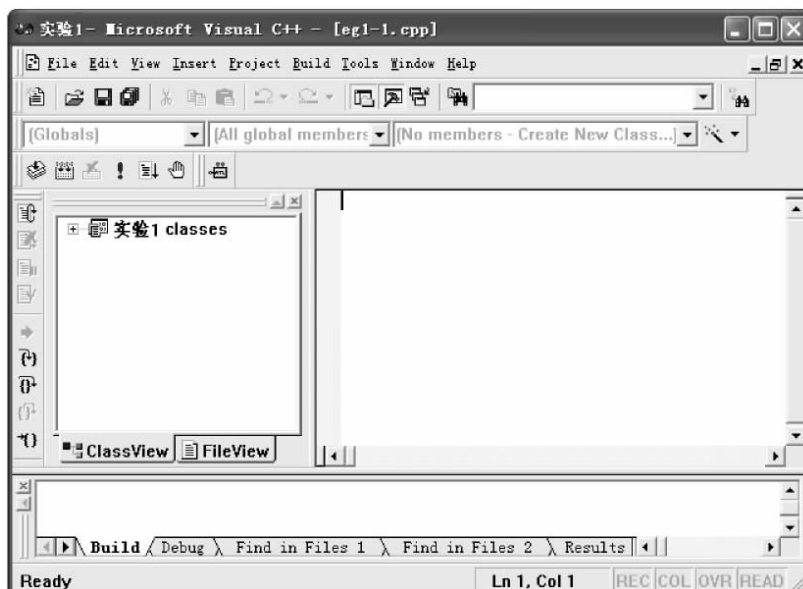


图 1.5 “C++源程序编辑”

(3) 在“C++源程序编辑”窗口下编辑 C++源程序,如图 1.6 所示。

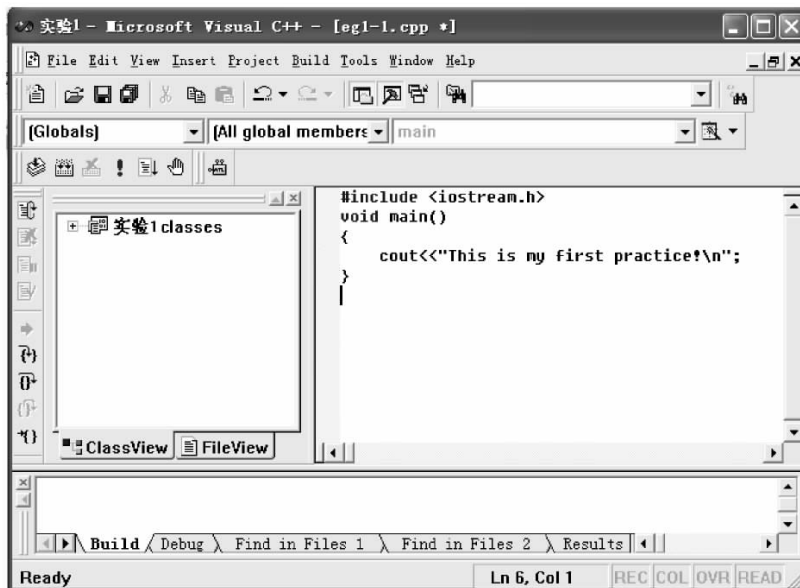


图 1.6 输入源程序代码

4. 编译、连接和运行源程序

(1) 选择【Build/Compile eg1-1.cpp】,这时系统开始对当前的源程序进行编译,在编译过程中,所发现的错误会显示在屏幕下方的“Build”窗口中。在显示的错误信息中指出该

错误所在行号和该错误的性质,用户可根据这些错误信息进行修改,如图 1.7 所示。

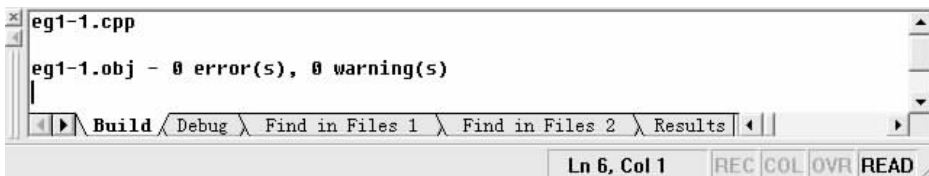


图 1.7 “编译”窗口

(2) 编译无错误后,可进行连接生成可执行文件(.exe)。选择【Build/Build 实验 1.exe】选项,“Build”窗口出现如图 1.8 所示的信息说明,编译连接成功后(无错误),即在当前工程文件夹下的 debug 文件夹生成可执行文件(实验 1.exe)。

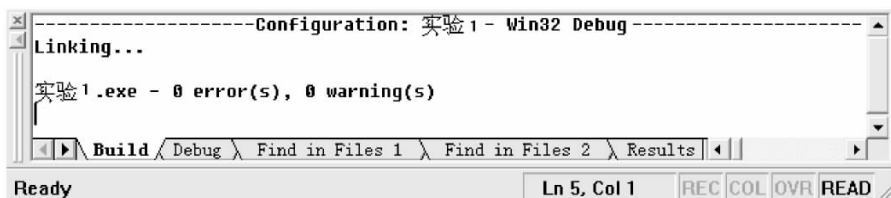


图 1.8 连接生成可执行文件

(3) 执行可执行文件:选择【Build/Execute 实验 1.exe】选项运行文件,结果显示在另外一个显示执行文件输出结果的窗口中(DOS 窗口),如图 1.9 所示。

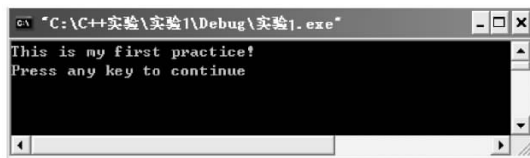


图 1.9 运行 C++ 程序结果

5. 关闭打开工作区、打开工作区

单击菜单【File/Close Workspace】,关闭工作区。

如果程序需要修改,可单击菜单【File/Open Workspace】,在弹出的对话框中选定“C:\C++实验\实验 1.dsw”,单击【打开】按钮,即可打开工作区,对已建立的工程文件进行修改。

6. 【实验内容】中第 3 个实验的步骤

可参考以上 1~5 步骤,建立、编辑、编译、连接、运行程序。

【思考与练习】

1. C++ 语言是在_____语言基础上发展起来的。
2. C++ 源程序文件的默认扩展名为_____。
3. C++ 程序从上机到得出结果的几个步骤依次是_____、_____、_____、_____。
4. 当执行 cin 语句时,从键盘上输入每个数据后必须接着输入一个_____符,然后才能继续输入下一个数据。
5. 编写一个 C++ 程序,显示您的姓名和地址。

实验 2 C++对 C 语言的扩充(1)

【实验目的】

1. 比较 C++与 C 语言中定义常量的不同方法。
2. 掌握重载函数的概念及函数参数在重载函数中的作用。
3. 理解函数模板和模板函数的概念,并会编写简单的函数模板。
4. 掌握内联函数的使用方法。

【实验准备】

1. 了解函数参数在重载函数中的作用。
2. 理解为什么要引入重载函数机制。
3. 熟悉函数模板的定义语法。
4. 明确内联函数的使用方法。

【实验内容】

1. 分别编写两个同名的函数 `max`,用于求两个整型数据的最大值和两个浮点型数据的最大值,在主函数中定义两个整型变量和两个浮点型变量,输入它们的数值,然后调用这两个函数。

分析: 重载函数的调用,编译器会根据传递参数的不同而决定使用哪一个函数。

程序设计如下:

```
#include <iostream. h>
inline int max(int x,int y)
{
    int z;
    z=x>y? x:y;
    return z;
}
inline float max(float x,float y)
{
    float z;
    z=x>y? x:y;
    return z;
}
int main()
{
```

```

int int _m,int _n;
float f _m,f _n;
cout<<"Please input int _m and int _n:";
cin>>int _m>>int _n;
cout<<"Please input f _m and f _n:";
cin>>f _m>>f _n;
cout<<"the max of two int is:"<<max(int _m,int _n)<<endl;
cout<<"the max of two float is:"<<max(f _m,f _n)<<endl;
return 0;
}

```

2. 根据上题内容,把 max 函数改为函数模板;与第 1 题进行比较,体会函数模板的用法。

分析: 功能相同的函数用函数模板设计,编译器从实参类型推导出模板函数的类型参数。

程序设计如下:

```

#include<iostream. h>
template<typename T>
T max(T x,T y)
{
    T z;
    z=x>y? x:y;
    return z;
}
int main()
{
    int int _m,int _n;
    float f _m,f _n;
    cout<<"Please input int _m and int _n:";
    cin>>int _m>>int _n;
    cout<<"Please input f _m and f _n:";
    cin>>f _m>>f _n;
    cout<<"the max of two int is:"<<max(int _m,int _n)<<endl;
    cout<<"the max of two float is:"<<max(f _m,f _n)<<endl;
    return 0;
}

```

【实验步骤】

1. 编辑源程序。
2. 对源程序进行编译并调试程序。

3. 连接并运行程序。
4. 检查输出结果是否正确。

【思考与练习】

1. 分析以下程序运行的结果,比较与 C 语言中符号常量的区别。

```
#include <iostream. h>
const int N=5;
void main()
{
    int i,p=1,s=0;
    for(i=1; i<N; i++)
    {
        p=p * i;
        s=s+p;
    }
    cout<<s<<endl;
}
```

2. 编写一个带有缺省参数的函数,该函数功能是求两个整数的和;在主函数中 3 次调用该函数,分别传递 2 个参数、传递 1 个参数和不传参数;观察结果并思考结果产生的原因。

3. 求最小值,主函数如下,请补充 min 函数,使程序完整。

```
#include<iostream. h>
void main( )
{
    cout<<"2 个数中的最小值:"<<min(2,8)<<endl;
    cout<<"3 个数中的最小值:"<<min(4,27,9)<<endl;
}
```

4. 编写一个分别能求整型数据和实型数据绝对值的函数模板。

实验3 C++对C语言的扩充(2)

【实验目的】

1. 掌握内联函数的作用及使用方法。
2. 掌握动态内存分配和释放的方法。
3. 了解引用和指针的区别,掌握引用的使用方法。

【实验准备】

1. 明确内联函数的使用方法。
2. 了解在什么情况下申请动态内存。
3. 理解引用的概念。

【实验内容】

设计一个描述学生的结构类型。结构中应当包括学生的学号、姓名、性别、年龄、家庭住址等信息。写一段代码,为该结构动态分配内存,然后给出结构中所有成员的值,并且显示输出。

分析: 不是声明结构变量,而是使用 new 操作符来为结构动态分配内存。注意 new 和 delete 的使用。

程序设计如下:

```
#include <iostream. h>
int main()
{
    struct Student
    {
        int num;
        char * name;
        char * sex;
        int age;
        char * add;
    };
    Student * p=new Student;           //动态内存分配
    p->num=1001;
    p->name="王小红";
    p->sex="女";
    p->age=19;
```

```

p->add="安徽省合肥市望江西路 555 号";
cout<<p->num<<"<<p->name<<"<<p->sex<<"<<p->add<<endl;
delete p; //内存释放,以免内存泄露
}

```

2. 编写程序,在主函数中输入两个整型数据,赋值给两个整型变量。通过函数调用,交换两个变量的值,参数传递采用引用传递方式,显示交换前和交换后的变量的值。被调函数请用内联函数实现。

分析: 本程序的目的是验证引用传递参数的作用。

程序设计如下:

```

#include <iostream. h>
inline swap(int & x,int & y)
{
    int t;
    t=x;
    x=y;
    y=t;
}
int main()
{
    int m,n;
    cout<<"input m and n:";
    cin>>m>>n;
    cout<<"m="<<m<<" n="<<n<<endl;
    swap(m,n);
    cout<<"after swap:\n";
    cout<<"m="<<m<<" n="<<n<<endl;
    return 0;
}

```

【实验步骤】

1. 编辑源程序。
2. 对源程序进行编译并调试程序。
3. 连接并运行程序。
4. 检查输出结果是否正确。

【思考与练习】

1. 思考内联函数的工程原理,明白内联函数的优点及使用注意事项。
2. 仔细阅读下列程序,写出结果并找出程序中存在的问题,请改正;仔细体会使用动态分配时应避免出现类似问题。

```
#include <iostream. h>
int fun();
int main()
{
    int a=fun();
    cout<<"the value of a is:"<<a;
    return 0;
}
int fun()
{
    int * p=new int(5);
    return * p;
}
```

3. 声明一个 int 型变量 a, 一个 int 型指针变量 p, 一个引用变量 r, 通过 p 把 a 的值改变为 5, 再通过 r 把 a 的值改为 3; 观察变量 a 值的变化, 思考指针和引用的区别。

实验 4 类和对象(1)

【实验目的】

1. 掌握类的定义方式。
2. 掌握类成员函数的定义方法。
3. 掌握类对象的定义及其访问方式。
4. 理解构造函数和析构函数的定义、作用和调用机制。
5. 掌握拷贝构造函数的作用和用法。

【实验准备】

1. 理解类和对象的概念。
2. 了解如何定义一个类,类中包括哪些内容。
3. 理解对象初始化的概念和方式。
4. 了解构造函数和析构函数的作用。
5. 了解什么时候需要拷贝构造函数。

【实验内容】

1. 定义一个学生类 Student,学生类中有 3 个私有数据成员:num(学号)、name(姓名)、age(年龄);3 个公有成员函数:Student(构造函数)、display、~Student(析构函数)。使用构造函数为 Student 类的对象赋值(name 使用动态内存分配空间),display 负责显示学生的基本信息,在析构函数中释放动态分配的内存。

分析: 在构造函数中为 name 动态分配内存,然后赋值,在析构函数中释放。

程序设计如下:

```
#include <iostream. h>
class Student
{
private:
    int num;
    char * name;
    int age;
public:
    Student(int n,char * na,int a)
    {
        cout<<"constructing..."<<endl;
        num=n;
```