



第1章 C++语言概述

练习题1

1.1 判断题

- × 1. C++语言和C语言都是面向对象的程序设计语言。
- ✓ 2. 面向对象方法具有封装性、继承性和多态性。
- ✓ 3. C语言是C++语言的一个子集。C++语言继承了C语言。
- × 4. C++语言程序与C语言程序一样都是函数串。
- × 5. C++语言支持封装性和继承性，不支持多态性。
- ✓ 6. C++语言比C语言对数据类型要求更加严格了。
- ✓ 7. C++语言对C语言进行了一次改进，使得编程更加方便了。
- × 8. C++源程序在编译时可能出现错误信息，而在连接时不会出现错误信息。
- ✓ 9. 编译C++源程序时，出现了警告错（Warning）也可以生成可执行文件。
- ✓ 10. C++语言程序的实现也要经过编辑、编译连接和运行3个步骤。

1.2 单选题

- 1. 下列关于面向对象概念的描述中，错误的是（ C ）。
 - A. 面向对象方法比面向过程方法更加先进
 - B. 面向对象方法中使用了一些面向过程方法中没有的概念
 - C. 面向对象方法替代了结构化程序设计方法
 - D. 面向对象程序设计方法要使用面向对象的程序设计语言
- 2. 下列各种高级语言中，不是面向对象的程序设计语言是（ D ）。
 - A. C++
 - B. Java
 - C. VB
 - D. C
- 3. 下列关于类的描述中，错误的是（ A ）。
 - A. 类就是C语言中的结构类型
 - B. 类是创建对象的模板
 - C. 类是抽象数据类型的实现
 - D. 类是具有共同行为的若干对象的统一描述体
- 4. 下列关于对象的描述中，错误的是（ C ）。
 - A. 对象是类的一个实例
 - B. 对象是属性和行为的封装体
 - C. 对象就是C语言中的结构变量
 - D. 对象是现实世界中客观存在的某种实体
- 5. 下列关于C++程序中使用提取符和插入符的输入/输出语句的描述中，错误的是（ C ）。
 - A. 提取符是对右移运算符（>>）重载得到的
 - B. 插入符是对左移运算符（<<）重载得到的
 - C. 提取符和插入符都是双目运算符，它们要求有两个操作数
 - D. 提取符和插入符在输入/输出语句中不可以连用

1.3 填空题

- 1. C++语言具有面向对象方法中要求的三大特性：封装性、继承性和多态性。
- 2. C++程序中，有且仅有一个主函数。
- 3. C++程序是由类和函数组成的。

4. C++源程序的扩展名是 cpp。
5. 使用插入符进行标准输出文件输出时，使用的输出流对象名是 cout。

1.4 上机调试下列程序，并分析输出结果

1.

```
#include <iostream.h>
void main()
{
    int i,j;
    cout<<"Enter i j: ";
    cin>>i>>j;
    cout<<"i="<<i<<','<<"j="<<j<<endl;
    cout<<"i+j="<<i+j<<','<<"i*j="<<i*j<<endl;
}
```

假定，输入数据如下：

Enter x y: 5 9

答： i=5,j=9

i+j=14,i *j=45

2.

```
#include <iostream.h>
int max(int,int);
void main()
{
    int a,b,c;
    a=5;
    b=8;
    c=max(a,b);
    cout<<"max("<<a<<','<<b<<")="<<c<<endl;
}
int max(int x,int y)
{
    return x>y?x:y;
}
```

答： max(5,8)=8

3. 文件 e1.4.3.cpp 内容如下：

```
#include <iostream.h>
void f1(),f2();
void main()
{
    cout<<"在北京.\n";
    f1();
    f2();
    cout<<"再见.\n";
}
```



```
}
```

文件 1.4.3-1.cpp 内容如下:

```
#include <iostream.h>
void f1()
{
    cout<<"在上海.\n";
}
```

文件 1.4.3-2.cpp 内容如下:

```
#include <iostream.h>
void f2()
{
    cout<<"在广州.\n";
}
```

答: 在北京。

在上海。

在广州。

再见。

1.5 编译下列程序, 修改所出现的错误, 获得正确结果

1.

```
main()
{
    cout<<"In main().\n"
}
```

答: 修改后如下:

```
#include <iostream.h>
void main()
{
    cout<<"In main().\n";
}
```

2.

```
#include <iostream.h>
void MAIN()
{
    cin>>a;
    int b=a+a;
    cout<<"b=<<b<<\n";
}
```

答: 修改后如下:

```
#include <iostream.h>
void main()
{
    int a;
```



```
cin>>a;
int b=a+a;
cout<<"b="<<b<<"\n";
}
```

3.

```
#include <iostream.h>
void main()
{
    int i,j;
    i=5;
    int k=i+j;
    COUT<<"i+j="<<k<<endl;
}
```

答:

```
#include <iostream.h>
void main()
{
    int i,j;
    i=10;
    i=5;
    int k=i+j;
    cout<<"i+j="<<k<<endl;
}
```

1.6 通过对 1.5 题中 3 个程序的修改，回答下列问题

1. 从 1.5 题中第 1 题程序的修改中，总结出编程应该注意哪些问题。

答: C++程序要包含 `iostraem.h`，`main()`函数前应加类型说明符 `void`。

2. C++程序中所出现的变量是否必须先说明后使用？在函数体内说明变量时是否都要放在函数体的开头？

答: C++程序中所出现的变量必须先说明后使用。在函数体内说明变量时不一定要放在函数体的开头。

3. 使用 `cout` 和插入符 (`<<`) 输出字符串常量时应注意什么？

答: 可输出一个字符串常量，也可输出多个字符串常量。每输出一个字符串常量要使用一次插入符 (`<<`)。

4. 程序中定义过的变量，但没有赋值，也没有默认值，这时能否使用？

答: 不能使用。因为它的值不确定。

5. 一个程序编译通过并已生成执行文件，运行后并获得输出结果，这一结果是否一定正确？

答: 不一定正确。可能还有算法错。

第 2 章 变量和表达式及数组

练习题 2

2.1 判断题

- ✓ 1. C++语言的合法字符集与 C 语言的完全相同。
- × 2. 标识符规定大小写字母没有区别。
- × 3. C++程序中，不得使用没有定义或说明的变量。
- ✓ 4. 变量的存储类指出了变量的作用域和寿命。
- ✓ 5. 变量的数据类型指出了变量在内存中存放的字节数。
- × 6. 定义变量时，变量的存储类说明符不得省略。
- × 7. 自动类变量与内部静态类变量的作用域和寿命都是相同的。
- × 8. 自动类变量可以定义在函数体外，这时应加说明符 `auto`。
- × 9. 外部类变量与外部静态类变量的作用域是相同的。
- × 10. 变量被定义后是否有默认值与存储类无关，与数据类型有关。
- ✓ 11. C++程序中，通常使用 `const` 来定义符号常量，定义时必须指出类型。
- × 12. 变量被定义或说明后，它一定具有有意义的值。
- × 13. 字符串常量与字符常量的区别仅表现在定义形式上的不同，一个用双撇号，另一个用单撇号。
- × 14. 所有变量的可见性和存在性都是一致的。
- ✓ 15. 变量在它的作用域内一定是可见的，又是存在的。
- ✓ 16. C++语言中除了包含 C 语言的所有运算符外，还规定自身的若干个运算符。
- ✓ 17. 增 1 和减 1 运算符以及赋值运算符都具有副作用。
- × 18. 增 1 和减 1 运算符不仅可以作用在变量上，也可以作用在表达式上。
- × 19. 关系运算符可以用来比较两个字符的大小，也可以比较两个字符串的大小。
- × 20. 移位运算符在移位操作中，无论左移还是右移，对移出的空位一律补 0。
- × 21. 变量的类型高低是指它被存放在内存的地址值大小。
- × 22. 使用 `sizeof` 运算符可以求得某种类型和某个变量在内存中占的字节数，不能求得某个表达式的类型在内存中所占的字节数。
- ✓ 23. 在 C++语言中，非保值转换应用强制类型转换。
- ✓ 24. 表达式中各操作数计算顺序取决于运算符的优先级和结合性。
- ✓ 25. 在 C++程序中，变量值是可以改变的，变量的地址值是不能改变的。
- ✓ 26. 数组中所有元素的类型都是相同的。
- × 27. 定义数组时必须对数组进行初始化。
- ✓ 28. 数组某维的大小可以用常量表达式，不可用变量名。
- × 29. 定义一个数组没有对它进行初始化，则该数组的元素值都是无意义的。
- ✓ 30. 用来给数组进行初始化的初始值表内的数据项的个数必须小于等于数组元素个数。
- × 31. 在定义一个数组时，对其部分元素进行了初始化，没有初始化的元素的值都是无意义的。
- ✓ 32. 数组被初始化时是判越界的。
- × 33. 字符数组就是字符串。
- ✓ 34. 使用 `printf()` 函数输出显示字符串时使用格式符 `%s`，输出显示字符时使用 `%c` 或



%d.

× 35. 给数组元素赋值时只可用常量表达式。

2.2 单选题

- 下列变量名中，非法的是（ C ）。
A. A25
B. My_car
C. My-str
D. abc
- 下列常量中，十六进制 int 型常量是（ A ）。
A. 0x5f
B. x2a
C. 046
D. 7a
- 下列常量中，不是字符常量的是（ B ）。
A. '\n'
B. "y"
C. 'x'
D. '\7'
- 在函数体内定义了下述变量 a，a 的存储类为（ D ）。
int a;
A. 寄存器类
B. 外部类
C. 静态类
D. 自动类
- 下列关于变量存储类的描述中，错误的是（ C ）。
A. 任何变量定义后都具有一个确定的存储类
B. 变量的存储类确定了变量的作用域和寿命
C. 定义变量时没有存储类说明符者一律为自动类
D. 内部静态类变量和外部静态类变量的存储类说明符都是 static
- 下列关于变量数据类型的描述中，错误的是（ A ）。
A. 定义变量时 int 型数据类型可以省略
B. 变量的数据类型可以决定该变量占内存的字节数
C. 变量的数据类型是可以被强制的
D. 变量的数据类型是有高低之分的
- 长双精度浮点型常量的后缀是（ C ）。
A. U
B. F
C. L
D. 无
- 下列运算符中，不能用于浮点数操作的是（ D ）。
A. ++
B. +
C. *=
D. &（双目）
- 下列运算符中，优先级最高的是（ A ）。
A. *（双目）
B. ||
C. >>
D. %=
- 下列运算符中，优先级最低的是（ B ）。
A. ==
B. ? :
C. |
D. &&
- 已知：int a(3); 下列表达式中，错误的是（ C ）。
A. a%2==0
B. a--+2
C. (a-2)++
D. a>>=2
- 已知：int b(5), 下列表达式中，正确的是（ D ）。
A. b="a"
B. ++(b-1)
C. b%2.5
D. b=3, b+1, b+2

13. 下列关于类型转换的描述中，错误的是（ C ）。
- A. 类型转换运算符是（<类型>）
 - B. 类型转换运算符是单目运算符
 - C. 类型转换运算符通常用于保值转换中
 - D. 类型转换运算符作用于表达式左边
14. 下列表达式中，其值为 0 的是（ A ）。
- A. 5/10
 - B. ! 0
 - C. 2>4? 0: 1
 - D. 2&&2||0
15. 下列表达式中，其值不为逻辑值的是（ A ）。
- A. 算术表达式
 - B. 关系表达式
 - C. 逗号表达式
 - D. 逻辑表达式
16. 下列关于数组概念的描述中，错误的是（ B ）。
- A. 数组中所有元素类型是相同的
 - B. 数组定义后，它的元素个数是可以改变的
 - C. 数组在定义时可以被初始化，也可以不被初始化
 - D. 数组元素的个数与定义时的每维大小有关
17. 下列关于数组维数的描述中，错误的是（ C ）。
- A. 定义数组时必须将每维的大小都明确指出
 - B. 二维数组是指该数组的维数为 2
 - C. 数组的维数可以使用常量表达式
 - D. 数组元素个数等于该数组的各维大小的乘积
18. 下列关于数组下标的描述中，错误的是（ B ）。
- A. C++语言中数组元素的下标是从 0 开始的
 - B. 数组元素下标是一个整常型表达式
 - C. 数组元素可以用下标来表示
 - D. 数组元素的某维下标值应小于该维的大小值
19. 下列关于初始值表的描述中，错误的是（ C ）。
- A. 数组可以使用初始值表进行初始化
 - B. 初始值表是用一对花括号括起的若干个数据项组成的
 - C. 初始值表中数据项的个数必须与该数组的元素个数相等
 - D. 使用初始值表给数组初始化时，没有被初始化的元素都具有默认值
20. 下列关于字符数组的描述中，错误的是（ D ）。
- A. 字符数组中的每一个元素都是字符
 - B. 字符数组可以使用初始值表进行初始化
 - C. 字符数组可以存放字符串
 - D. 字符数组就是字符串
21. 下列关于字符串的描述中，错误的是（ C ）。
- A. 一维字符数组可以存放一个字符串
 - B. 二维字符数组可以存放多个字符串
 - C. 可以使用一个字符串给二维字符数组赋值
 - D. 可以用一个字符串给二维字符数组初始化
22. 已知：int a[5] = {1, 2, 3, 4}; 下列数组元素值为 2 的数组元素是（ B ）。
- A. a[0]
 - B. a[1]
 - C. a[2]
 - D. a[3]

- ### 2.3 填空题

- ## 2.4 分析下列程序

- 1.

假定输入为 5 和 8。

- 答: $d=3$

- 2.

8


```

const double D=8.5;
cout<<"A="<<A<<endl;
cout<<"CH+2="<<char(CH+2)<<endl;
cout<<"D-5.8="<<D-5.8<<endl;
}

```

答: A=8
CH=2=m
D-5.8=2.7

3.

```

#include <iostream.h>
int a=9;
void main()
{
    int b=5;
    cout<<"a+b="<<a+b<<endl;
    static int c;
    cout<<"c+a="<<c+a<<endl;
    const long int d=8;
    cout<<"a+d-b="<<a+d-b<<endl;
}

```

答: a+b=14
c+a=9
a+d-b=12

4.

```

#include <iostream.h>
void main()
{
    int a=3;
    char b='m';
    cout<<"a="<<a<<','<<"b="<<b<<endl;
    {
        int a=5;
        b='n';
        cout<<"a="<<a<<','<<"b="<<b<<endl;
    }
    cout<<"a="<<a<<','<<"b="<<b<<endl;
}

```

答: a=5,b=m
a=3,b=n
a=3,b=n

5.

```

#include <iostream.h>

```



课后习题答案网

—— 思源岛下载

```

void main()
{
    cout<<6%4*5/3+3<<endl;
    cout<<(10&3|8)<<endl;
    cout<<(5<<2)+(5>>2)<<endl;
    cout<<~4<<endl;
}

```

答: 6
10
21
-5



课后习题答案网
—— 思源岛下载

6.

```

#include <iostream.h>
void main()
{
    int a=5;
    cout<<long(&a)<<','<<sizeof(a)<<','<<sizeof(int)<<endl;
    double b=1.5;
    cout<<sizeof(1.5f)<<','<<sizeof(b)<<','<<sizeof(1.5L)<<endl;
}

```

答: <地址值>,4,4
4,8,8

7.

```

#include <iostream.h>
void main()
{
    int a=3,b=5;
    cout<<(a>b+a==b-2)<<','<<(a!=b+a<=b)<<endl;
    char c='k';
    cout<<(c<='k')<<','<<(--c!='h'+2)<<endl;
    float f=2.3f;
    cout<<(--f<=f)<<endl;
}

```

答: 0,1
1,0
1

8.

```

#include <iostream.h>
void main()
{
    int i,j,k;
    i=j=k=5;
}

```

```

        !i&&++j&&--k;
        cout<<i<<','<<j<<','<<k<<endl;
        i||j--||++k;
        cout<<i<<','<<j<<','<<k<<endl;
        i-5||j-5&&++k;
        cout<<i<<','<<j<<','<<k<<endl;
    }

```

答: 5,5,5
5,5,5
5,5,5

9.

```

#include <iostream.h>
void main()
{
    int i(5),j(8);
    cout<<(i=i*j)<<endl;
    i=5,j=8;
    i=5*j/(j-- -5);
    cout<<i<<','<<j<<endl;
    i=5,j=8;
    cout<<(j+=i*j-5)<<endl;
}

```

答: 40
13,7
23



课后习题答案网
—— 思源岛下载

10.

```

#include <iostream.h>
int a=8;
void main()
{
    int b=6;
    double d=1.5;
    d+=a+b;
    cout<<d<<endl;
    cout<<(a=1,b=2,d=2.5,a+b+d)<<endl;
    cout<<(a<0?a:b<0?a++:b++)<<endl;
}

```

答: 15.5
5.5
2

11.

```

#include <iostream.h>
void main()

```

```

{
    int m[][3]={9,8,7,6,5,4,3,2,1},s=0;
    for(int i=0;i<3;i++)
        s+=m[i][i];
    cout<<s<<endl;
}

```

答： 15

12.

```

#include <iostream.h>
void main()
{
    char s[]="bhy543kpm345";
    for(int i=0;s[i]!='\0';i++)
    {
        if(s[i]>='a'&&s[i]<='z')
            continue;
        cout<<s[i];
    }
    cout<<endl;
}

```

答： 543345



课后习题答案网

——思路岛下载

13.

```

#include <iostream.h>
void main()
{
    int b[]={5,-3,4,1,-8,9,0,10};
    int i=0,j;
    for(j=i;i<8;i++)
        if(b[i]>b[j])
            j=i;
    cout<<j<<','<<b[j]<<endl;
}

```

答： 7,10

14.

```

#include <iostream.h>
void main()
{
    int a[][3]={ 1,2,3,4,5,6,7,8,9};
    int s1(0),s2(0);
    for(int i=0;i<3;i++)
        for(int j=0;j<3;j++)
        {
            if(i!=j)

```

```

        s1+=a[i][j];
        if(i+j==1)
            s2+=a[i][j];
    }
    cout<<"s1="<<s1<<','<<"s2="<<s2<<endl;
}

```

答: s1=30,s2=6

15.

```

#include <iostream.h>
void main()
{
    char s[]="#%#";
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<i;j++)
            cout<<' ';
        for(int k=0;k<3;k++)
            cout<<s[k];
        cout<<endl;
    }
    for(i=2;i>0;i--)
    {
        for(int j=0;j<i-1;j++)
            cout<<' ';
        for(int k=0;k<3;k++)
            cout<<s[k];
        cout<<endl;
    }
}

```

答: ##

##



课后习题答案网

www.sld.net.cn

思路岛下载

2.5 编程题

1. 已知: int a=3, b=5; 编程计算下列两个代数式的值, 并比较它们是否相等。

$$(a+b)^2$$

和

$$a^2+2ab+b^2$$

答: 编程如下:

```

#include <iostream.h>
void main()
{
    int a=3,b=5;
    int d1=(a+b)*(a+b);
}

```

```

int d2=a*a+2*a*b+b*b;
cout<<"d1="<<d1<<endl<<"d2="<<d2<<endl;
}

```

从结果中可以看出 d1 和 d2 值是相等的。

2. 已知：int x=5；编程求下列代数式的值。

$$f(x)=3x^3+2x^2+5x+2$$

答：编程如下：

```

#include <iostream.h>
void main()
{
    int x=5;
    int f=3*x*x*x+2*x*x+5*x+2;
    cout<<f<<endl;
}

```



课后习题答案网
—— 思源岛下载

3. 从键盘上输入两个 double 型数，编程输出其中最小者。

答：编程如下：

```

#include <iostream.h>
void main()
{
    double a,b;
    cout<<"请输入两个 double 型数： ";
    cin>>a>>b;
    cout<<"输入的两个数中较小的是";
    if(a<=b)
        cout<<a<<endl;
    else
        cout<<b<<endl;
}

```

4. 华氏温度转换成摄氏温度的计算公式如下：

$$C=(F-32)*5/9$$

其中，C 表示摄氏温度，F 表示华氏温度。从键盘上输入一摄氏温度，编程输出对应的华氏温度。

答：编程如下：

```

#include <iostream.h>
void main()
{
    double c,f;
    cout<<"请输入一个摄氏温度值： ";
    cin>>c;
    f=9.0/5.0*c+32.0;
    cout<<"摄氏温度为"<<c<<"华氏温度为"<<f<<endl;
}

```

```
}
```

5. 从键盘上输入 5 个浮点数，输出它们的和以及平均值。

答：编程如下：

```
#include <iostream.h>
void main()
{
    double a,b,c,d,e;
    cout<<"请输入 5 个 double 型数: ";
    cin>>a>>b>>c>>d>>e;
    double f=a+b+c+d+e;
    cout<<"输入的 5 个数的和是"<<f<<"平均值是"<<f/5.0<<endl;
}
```

6. 将字符串"12345", 逆向输出为"54321"。

答：编程如下：

```
#include <iostream.h>
void main()
{
    char a[]="12345";
    cout<<a[4]<<a[3]<<a[2]<<a[1]<<a[0]<<endl;
}
```

2.6 简单回答下述问题

1. C++语言中注释符的格式如何？注释信息的功能是什么？

答：注释符格式有 2 种：

格式一： /* <注释信息> */

格式二： // <注释信息>

注释信息是用来对程序进行提示和说明的。

2. 使用 const 定义符号常量比使用#define 定义符号常量有何优点？

答：使用 const 定义符号常量时要指出数据类型，而用#define 定义符号常量时不需提出类型，因此前者比后者对类型要求更严格，这样会更安全。

3. 内部静态存储类变量有何特点？

答：内部静态存储类变量的特点是作用域较小，但寿命很长，这种变量作用域与寿命不一致，因此会出现不可见但又存在的情况。

4. 在使用插入符 (<<) 输出若干个表达式值时，有的表达式中使用的运算符的优先级低于插入符优先级时应该如何处理？

答：这时需将表达式用括号括起来，已改变其优先级。

5. 由多种不同运算符组成的表达式，其类型如何确定？例如，

```
int a, b, c;
...
a=b>c? b:c;
```

这是一个条件表达式还是赋值表达式？

答：表达式类型由优先级低的运算符组成的表达式来确定。

这是一个赋值表达式，因为赋值运算符优先级低于三目运算符。

6. 数组元素个数是由什么决定的？

答：数组元素个数是由维数及其大小来决定的。一维数组元素个数是该维的大小，二维数组的元素个数是两维大小之积等。

7. 数组元素下标有何规定？

答：C++语言中，数组元素下标从0开始。

8. 初始值表中数据项的类型和个数有何规定？

答：初始值表中数据项的类型应与该变量类型相同，个数应小于或等于被初始化的数组元素的个数。

9. 字符数组和字符串有什么关系？

答：字符数组中可存放字符串，而字符数组不都是存放字符串的。如果字符数组中有字符串结束符，则说明该字符数组存放的是字符串。

10. 一个数组中的元素类型是否一定相同？何时定义的数组元素具有默认值？

答：数组中元素类型必须相同。存储类为外部和静态的数组其元素具有默认值。

第3章 语句和预处理

练习题3

3.1 判断题

- ✓ 1. 表达式和表达式语句是不同的。
- × 2. 空语句是一种没有用处的语句。
- × 3. 复合语句就是分程序。
- × 4. 条件语句中 if 子句和 else 子句都是必须有并且仅有一个。
- ✓ 5. 条件语句中 else if 子句可以没有，也可以有多个。
- ✓ 6. 开关语句可实现多路分支。
- × 7. 开关语句的<语句序列>中必须有一个 break 语句，否则该开关语句便无法退出。
- × 8. 任何循环语句都是至少执行一次循环体。
- × 9. 退出 for 循环语句必须是 for 后面括号内的中间一个表达式的值为0。
- ✓ 10. do-while 循环语句至少要执行一次循环体。
- × 11. 循环语句的循环体中可以出现 if 语句，if 语句的 if 体内不能出现循环语句。
- × 12. goto 语句中所使用的语句标号是一种标识符，它的作用域是文件级的。
- ✓ 13. break 语句和 continue 语句都可以出现在循环体中，但是它们的作用是不同的。
- × 14. 文件包含命令所能包含的文件类型是不受限制的。

3.2 单选题

1. 下列关于语句的描述中，错误的是（ A ）。
 - A. C++程序中的函数是由若干条语句组成的
 - B. 每条语句都要实现某种操作
 - C. 条件语句是用来实现分支操作的
 - D. 循环语句是用来在一定条件下重复执行某段程序的
2. 下列关于条件语句的描述中，错误的是（ C ）。
 - A. if 语句中最多只能有一个 else 子句

- B. if 语句的 if 体内可以出现开关语句
 C. if 语句中 else if 子句和 else 子句的顺序是没有限制的
 D. if 语句中 else 子句是与它最近的 if 子句配对的
3. 下列关于开关语句的描述中，错误的是（ B ）。
- A. 开关语句中，case 子句的个数是不受限制的
 B. 开关语句中，case 子句的语句序列中一定要有 break 语句
 C. 开关语句中，default 子句可以省略
 D. 开关语句中，右花括号具有退出开关语句的功能
4. 下列关于循环语句的描述中，错误的是（ A ）。
- A. while 循环语句中<条件>给定的表达式不能为非 0 的常量，否则便是死循环
 B. for 循环语句的循环体内可以出现 while 循环语句、do-while 循环语句和 for 循环语句
 C. 循环语句的循环体可以是空语句
 D. 循环语句的循环体内可以出现 break 语句，也可以出现 continue 语句
5. 已知：int i(3); 下列 do-while 循环语句的循环次数是（ D ）。
- ```
do{
 cout<<i--<<endl;
 i--;
}while(i!=0);
```
- A. 0  
 B. 3  
 C. 1  
 D. 无限
6. 下列 for 循环语句的循环次数是（ B ）。
- ```
for(int i(0),j(5); i=3; i++,j--);
```
- A. 3
 B. 无限
 C. 5
 D. 0
7. 下列 while 循环语句的循环次数是（ A ）。
- ```
while(int i(0)) i--;
```
- A. 0  
 B. 1  
 C. 2  
 D. 无限
8. 下列程序段执行后，j 值是（ B ）。
- ```
for(int i(0),j(0); i<10; i++)
    if(i) j++;
```
- A. 0
 B. 9
 C. 10
 D. 无限
9. 已知：typedef char CH; 下列描述中，正确的是（ A ）。
- A. 使用 CH 定义的变量是 char 型变量
 B. 使用 CH 定义的是一个字符常量
 C. 使用 CH 定义的变量其类型不确定
 D. 使用 CH 定义的是一个字符串
10. 下列关于预处理命令的描述中，错误的是（ D ）。
- A. 预处理命令最左边的标识符是#
 B. 预处理命令是在编译前处理的
 C. 宏定义命令可以定义符号常量
 D. 文件包含命令只能包含.h 文件

3.3 填空题

1. 表达式语句是一个表达式后边加上 分号(;) 组成的。空语句是 只有一个分号的语句。

2. 复合语句是由 两条 条或 两条 条以上的语句加上 花括号({}) 组成的。

3. 分程序是一种带有 说明 语句的复合语句。

4. 循环语句的共同特点是都应具有 循环体 和 退出循环的条件。

5. 下列程序是求 100 之内的能被 7 整除的自然数之和。

```
#include <iostream.h>
void main( )
{
    int sum;
    sum=0 ;
    for(int i(1); i<=100; i++)
        if( i%7==0 )
            sum+=i;
    cout<<sum<<endl;
}
```

3.4 上机调试下列程序，并分析其输出结果

1.

```
#include <iostream.h>
void main()
{
    int a(8),b(5);
    if(!a)
        b--;
    else if(b)
        if(a)
            a++;
        else
            a--;
    else
        b++;
    cout<<a<<','<<b<<endl;
}
```

答： 9,5

2.

```
#include <iostream.h>
void main()
{
    int a(10);
    while(--a)
    {
        if(a==5) break;
        if(a%2==0&& a%3==0) continue;
        cout<<a<<endl;
    }
}
```



课后习题答案网
——思路岛下载

}
答: 9
8
7



课后习题答案网
—— 思路岛下载

3.

```
#include <iostream.h>
void main()
{
    int b(10);
    do {
        ++b;
        cout<<++b<<endl;
        if(b==15) break;
    }while(b<15);
    cout<<"ok! "<<endl;
}
```

答: 12
14
16
ok!

4.

```
#include <iostream.h>
void main()
{
    int w(5);
    do {
        switch(w%2)
        {
            case 1: w--; break;
            case 0: w++; break;
        }
        w--;
        cout<<w<<endl;
    }while(w>0);
}
```

答: 3
1
-1

5.

```
#include <iostream.h>
void main()
{
    int a(4),b(5),i(0),j(0);
```

```

switch(a)
{
    case 4: switch(b)
        {
            case 4: i++; break;
            case 5: j++; break;
            default: i++;j++;
        }
    case 5: i++; j++;
        break;
    default: i++;j++;
}
cout<<i<<','<<j<<endl;
}

```

答: 1,2

6.

```

#include <iostream.h>
void main()
{
    int b(10);
    for(int i=9;i>=0;i--)
    {
        switch(i)
        {
            case 1: case 4: case 7: b++;break;
            case 2: case 5: case 8: break;
            case 3: case 6: case 9: b+=2;
        }
    }
    cout<<b<<endl;
}

```

答: 19

7.

```

#include <iostream.h>
void main()
{
    int a(6);
    for(int i(1);i<=a;i++)
    {
        for(int j=1;j<=a-i;j++)
            cout<<' ';
        for(j=1;j<=2*i-1;j++)
            cout<<'A';
        cout<<endl;
    }
}

```



课后习题答案网
—— 思源岛下载

www.sld.net.cn

```

    }
}
答:      A
          AAA
          AAAAA
          AAAAAAA
          AAAAAAAAA
          AAAAAAAAAA
          AAAAAAAAAAA

```



课后习题答案网
—— 思路岛下载

8.

```

#include <iostream.h>
#define MAX(x,y) (x)>(y)?(x):(y)
void main()
{
    typedef int IN;
    IN a(3),b(4),c;
    c=MAX(a,b)*2;
    cout<<c<<endl;
}

```

答: 8

3.5 编程题

1. 求 100 之内的自然数中奇数之和。

答: 编程如下:

```

#include <iostream.h>
void main()
{
    int s=0;
    for(int i=1;i<100;i+=2)
        s+=i;
    cout<<s<<endl;
}

```

2. 求两个整数的最大公约数和最小公倍数。

答: 编程如下:

假定两个整数分别为 15 和 25, 求它们的最大公约数和最小公倍数。

```

#include <iostream.h>
void main()
{
    int m,n,t;
    m=15;
    n=25;
    if(m<n)
    {
        t=m;

```

```

        m=n;
        n=t;
    }
    int m0=m,n0=n;
    while(n!=0)
    {
        t=m%n;
        m=n;
        n=t;
    }
    cout<<"最大公约数为"<<m<<endl;
    cout<<"最小公倍数为"<<m0*n0/m<<endl;
}

```



3. 求下列分数序列前 15 项之和。

$2/1, 3/2, 5/3, 8/5, 13/8, \dots$

答：编程如下：

```

#include <iostream.h>
void main()
{
    int a;
    double i=2,j=1,sum=0,t;
    for(a=1;a<15;a++)
    {
        sum+=i/j;
        t=i;
        i+=j;
        j=t;
    }
    cout<<"SUM="<<sum<<endl;
}

```

4. 按下列公式，求 e 的近似值。

$e=1+1/1! + 1/2! + 1/3! + \dots + 1/n!$

答：编程如下：

```

#include <iostream.h>
#include <math.h>
void main()
{
    double x(1),s(0);
    for(int i=1;fabs(x)>1e-8;i++)
    {
        x*=(-1.0)*(2*i-3)/(2*i-1);
        s+=x;
    }
}

```

```

    }
    s*=4;
    cout<<"pi="<<s<<endl;
}

```

5. 求下列式子之和，假定 $n=10$ 。

$$S=1+(1+2)+(1+2+3)+\cdots+(1+2+3+\cdots+n)$$

答：编程如下：

```

#include <iostream.h>
void main()
{
    int s(0),s1(0);
    for(int i=1;i<=10;i++)
    {
        s1+=i;
        s+=s1;
    }
    cout<<s<<endl;
}

```



课后习题答案网
—— 思皓岛下载

3.6 简单回答下列问题

1. 分程序是复合语句吗？复合语句与分程序的区别是什么？

答：分程序是复合语句。复合语句与分程序的区别在于含有说明语句的复合语句称为分程序。

2. if 语句中，else 子句的功能是什么？

答：该子句的功能是当 if 子句和所有 else if 子句的条件都为 0 时，则执行 else 子句。

3. 在开关语句中，是否每个 case 子句后面的<语句序列>中都应该有 break 语句？

答：通常 case 子句后面的<语句序列>中都有 break 语句，根据需要也可以没有 break 语句。

4. 循环语句的循环体中出现的 continue 语句的作用是什么？

答：结束本次循环，转到循环头判断是否继续循环。

5. 文件包含命令的功能吗？该命令一定要放在程序首部吗？

答：该预处理命令是将指定的包含文件内容放在该命令出现的地方。通常放在程序开头，便于后边语句的使用。

第 4 章 指针和引用

练习题 4

4.1 判断题

× 1. 指针是变量，它具有的值是某个变量或对象的地址值，它还具有一个地址值，这两个地址值是相等的。

- ✓ 2. 指针的类型是它所指向的变量或对象的类型。
- × 3. 定义指针时不可以赋初值。
- ✓ 4. 指针可以赋值，给指针赋值时一定要类型相同，级别一致。
- × 5. 指针可以加上或减去一个 `int` 型数，也可以加上一个指针。
- ✓ 6. 两个指针在任何情况下相减都是有意义的。
- ✓ 7. 数组元素可以用下标表示，也可以用指针表示。
- × 8. 指向数组元素的指针只可指向数组的首元素。
- ✓ 9. 指向一维数组的指针是一个二级指针。
- × 10. 指针数组的元素可以是不同类型的指针。
- ✓ 11. 字符指针是指向字符串的指针，可以用字符串常量给字符指针赋值。
- × 12. 引用是一种变量，它也有值和地址值。
- ✓ 13. 引用是某个变量的别名，引用是被绑定在被引用的变量上。
- ✓ 14. 创建引用时要用一个同类型的变量进行初始化。
- ✓ 15. 指针是变量，它可以有引用，而引用不能有引用。

4.2 单选题

1. 下列关于定义一个指向 `double` 型变量的指针，正确的是（ B ）。
 - A. `int a(5); double *pd=a;`
 - B. `double d(2.5), *pd=&d;`
 - C. `double d(2.5), *pd=d;`
 - D. `double a(2.5), pd=d;`
2. 下列关于创建一个 `int` 型变量的引用，正确的是（ A ）。
 - A. `int a(3), &ra=a;`
 - B. `int a(3),&ra=&a;`
 - C. `double d(3.1); int &rd=d;`
 - D. `int a(3), ra=a;`
3. 下列关于指针概念的描述中，错误的是（ B ）。
 - A. 指针中存放的是某变量或对象的地址值
 - B. 指针的类型是它所存放的数值的类型
 - C. 指针是变量，它也具有一个内存地址值
 - D. 指针的值（非常量指针）是可以改变的
4. 下列关于引用概念的描述中，错误的是（ A ）。
 - A. 引用是变量，它具有值和地址值
 - B. 引用不可以作数组元素
 - C. 引用是变量的别名
 - D. 创建引用时必须进行初始化
5. 已知：`int a[5], *p=a;` 则与 `++*p` 相同的是（ D ）。
 - A. `*++p`
 - B. `a[0]`
 - C. `*p++`
 - D. `++a[0]`
6. 已知：`int a[]={1,2,3,4,5}, *p=a;` 在下列数组元素地址的表示中，正确的是（ C ）。
 - A. `&(a+1)`
 - B. `&(p+1)`
 - C. `&p[2]`
 - D. `*p++`
7. 已知：`int a[3][4],(*p)[4];` 下列赋值表达式中，正确的是（ A ）。
 - A. `p=a+2`
 - B. `p=a[1]`
 - C. `p=*a`
 - D. `p=*a+2`
8. 已知：`int b[3][5]={0};` 下列数组元素值的表示中，错误的是（ D ）。
 - A. `**(&b+1)`
 - B. `(*(b+1))[2]`
 - C. `*(*(b+1)+1)`
 - D. `*(b+2)`
9. 已知：`int a=1, b=2, *p[2];` 下列表达式中正确的是（ C ）。
 - A. `*p[2]`
 - B. `*p[2]`
 - C. `*p[2]`
 - D. `*p[2]`

- A. p=&a
C. p[0]=&a, p[1]=&b
- B. p=&b
D. p[]={&a, &b};

10. 已知: int a(5), &ra=a; 下列描述中, 错误的是 (D)。

- A. ra 是变量 a 的引用, 即为变量的别名
C. ra 的地址值为 &a
- B. ra 的值为 5
D. 改变 ra 的值为 10, 变量 a 值仍为 5

4.3 填空题

1. 单目运算符&作用在变量名左边, 表示该变量的 地址值, 单目运算符*作用在指针名的左边, 表示取该变量的 值。

2. 指向一维数组元素的指针是 一 级指针, 指向二维数组元素的指针是 一 级指针, 指向一维数组的指针是 二 级指针, 指向一级指针的指针是 二 级指针, 一维一级指针数组名是 二 级指针的地址值, 二维数组的数组名是 二 级指针的地址值。

3. 在一个二维数组b[3][5]中, b[0]与 *b 是等价的, &b[1]与 b+1 是等价的。

4. 指针的运算有 4 种, 它们是 赋值 运算、一个指针加减整型数的运算、两个指针相减和 比较 运算。

5. 引用不是变量, 它是某个变量或对象的 别名。引用的值是 被引用的变量的值, 引用的地址值是 被引用的变量的地址值。

4.4 分析下列程序的输出结果

1.

```
#include <iostream.h>
void main()
{
    int a[]={5,4,3,2,1};
    int *p=&a[2];
    int m(5),n;
    for(int i(2);i>=0;i--)
    {
        n=(*(p+i)<*a)?*(p+i):*a;
        cout<<n<<endl;
    }
}
```

答: 1

2

3

2.

```
#include <iostream.h>
void main()
{
    char *p1,*p2;
    p1="abcqrv";
    p2="abcpqo";
    while(*p1&&*p2&&*p2++==*p1++)
        ;
    int n=*(p1-1)-*(p2-1);
    cout<<n<<endl;
}
```



课后习题答案网

—— 思路岛下载

答: 1

3.

```
#include <iostream.h>
int a[]={ 10,9,6,5,4,2,1};
void main()
{
    int n(7),i(7),x(7);
    while(x>*(a+i))
    {
        *(a+i+1)=*(a+i);
        i--;
    }
    *(a+i+1)=x;
    for(i=0;i<n;i++)
        cout<<*(a+i)<<',';
    cout<<a[i]<<endl;
}
```

答: 10,9,7,6,5,4,2,1



课后习题答案网
——思路岛下载

4.

```
#include <iostream.h>
int a[][3]={ 1,2,3,4,5,6,7,8,9};
int *p[]={a[0],a[1],a[2]};
int **pp=p;
void main()
{
    int (*s)[3]=a;
    for(int i(1);i<3;i++)
        for(int j(0);j<2;j++)
            cout<<*(a[i]+j)<<','<<*(*(p+i)+j)<<','<<
            <<(*(*pp+i))[j]<<','<<*(s+3*i+j)<<endl;
}
```

答: 4,4,4,4

5,5,5,5

7,7,7,7

8,8,8,8

5.

```
#include <iostream.h>
int a[]={ 1,2,3,4,5,6,7,8,9};
void main()
{
    int *pa=a;
    cout<<*pa<<',';
    cout<<*(pa++)<<',';
}
```

```

        cout<<*++pa<<',';
        cout<<*(pa--)<<',';
        pa+=4;
        cout<<*pa<<','<<*(pa+2)<<endl;
    }

```

答： 1,1,3,3,6,8

6.

```

#include <iostream.h>
void main()
{
    char str[][4]={"345","789"},*m[2];
    int s(0);
    for(int i=0;i<2;i++)
        m[i]=str[i];
    for(i=0;i<2;i++)
        for(int j(0);j<4;j+=2)
            s+=m[i][j]-'0';
    cout<<s<<endl;
}

```

答： 24



课后习题答案网
—— 思路岛下载

www.sld.net.cn

7.

```

#include <iostream.h>
void main()
{
    double d1=3.2,d2=5.2;
    double &rd1=d1,&rd2=d2;
    cout<<rd1+rd2<<','<<d1+rd2<<endl;
    rd1=9.3;
    cout<<rd1+rd2<<','<<d1+rd2<<endl;
    d2=0.8;
    cout<<2*rd2<<endl;
}

```

答： 8.4,8.4

14.5,14.5

1.6

8.

```

#include <iostream.h>
void main()
{
    int *p;
    int *&rp=p;
}

```

```
int a=90;
p=&a;
cout<<"a="<<a<<', '<<"*rp="<<*rp<<endl;
int b=50;
rp=&b;
cout<<"b="<<b<<', '<<"*rp="<<*rp<<endl;
}
```

答: a=90,*rp=90
b=50,*rp=50

4.5 编程题（使用指针）

1. 已知 4 个字符串，编程输出它们中最小的一个。

答：编程如下：

```
#include <iostream.h>
#include <string.h>
void swap(char p1[],char p2[]);
void main()
{
    char s1[10]="while",s2[10]="else",s3[10]="default",s4[10]="continue";
    if(strcmp(s1,s2))
        swap(s1,s2);
    if(strcmp(s3,s4))
        swap(s3,s4);
    if(strcmp(s1,s3))
        swap(s1,s3);
    cout<<s1<<endl;
}
void swap(char p1[],char p2[])
{
    char p[10];
    strcpy(p,p1);
    strcpy(p1,p2);
    strcpy(p2,p);
}
```

2. 将一个长度为 n 的字符串，编程实现其逆序输出。

答：编程如下：

```
#include <iostream.h>
#include <string.h>
void inverse(char []);
void main()
{
    char s[50];
    cout<<"输入一个字符串： ";
```

```

        cin>>s;
        inverse(s);
        cout<<"反序后的字符串是 "<<s<<endl;
    }
void inverse(char p[])
{
    int n=strlen(p);
    int limit=n/2;
    for(int i=0;i<limit;i++)
    {
        char t=p[i];
        p[i]=p[n-i-1];
        p[n-i-1]=t;
    }
}

```



课后习题答案网
—— 思路岛下载

3. 已知一个二维 int 型数组，编程求出它的最小的元素值。

答：编程如下：

```

#include <iostream.h>
void main()
{
    int n[3][2]={ {3,8},{-5,9},{2,4}};
    int lim=n[0][0];
    for(int i=0;i<3;i++)
        for(int j=0;j<2;j++)
            if(n[i][j]<lim)
                lim=n[i][j];
    cout<<"最小元素值为 "<<lim<<endl;
}

```

4. 已知字符型指针数组中存放若干个字符串，编程从键盘上修改其中某个字符串。

答：编程如下：

```

#include <iostream.h>
#include <string.h>
char *str[5]={ "abc","def","ghi","jkl","pqo"};
char *modify(int,char []);
void main()
{
    int n;
    char s[10];
    cout<<"修改第几个字符串: ";
    cin>>n;
    cout<<"该字符串修改为: ";
    cin>>s;
}

```

```

        cout<<"修改后的字符串是 "<<modify(n,s)<<endl;
        cout<<"修改后字符指针数组中的字符串如下: \n";
        for(int i=0;i<5;i++)
            cout<<str[i]<<endl;
    }
    char *modify(int n,char s[])
    {
        return *(str+n-1)=s;
    }
}

```

5. 有 n 个小孩排成一圈。从第 1 个小孩开始作 1 至 3 报数，凡报数为 3 的小孩从圈中出来，求最后出圈的小孩的顺序号是多少？

答：编程如下：

```

#include <iostream.h>
void main()
{
    int n,num[100];
    cout<<"输入排号人数: ";
    cin>>n;
    int *p=num;
    for(int i=0;i<n;i++)
        *(p+i)=i+1;
    i=0;
    int k=0,a=0;
    while(a<n-1)
    {
        if(*(p+i)!=0)
            k++;
        if(k==3)
        {
            *(p+i)=0;
            k=0;
            a++;
        }
        i++;
        if(i==n)
            i=0;
    }
    while(*p==0)
        p++;
    cout<<"最后留下的是第 "<<*p<<" 号"<<endl;
}

```



课后习题答案网
——思路岛下载

4.6 简单回答下列问题

- A. 一个函数中可以有多条 return 语句
B. return 语句具有返回程序控制权的作用
C. 函数通过 return 语句返回值时仅有一个
D. 一个函数中有且仅有一条 return 语句
4. 函数返回值的类型是由 (B) 决定的。
A. 调用该函数的调用函数的类型
B. 定义该函数时所指定的类型
C. return 语句中表达式的类型
D. 接收函数返回值的变量或对象的类型
5. 下列设置函数参数默认值的说明语句中, 错误的是 (C)。
A. int fun(int x, int y=10); B. int fun(int x=5, int =10);
C. int fun(int x=5, int y); D. int fun(int x, int y=a+b);
(其中, a 和 b 是已定义过具有有效值的变量)
6. 下列选择重载函数的不同实现的判断条件中, 错误的是 (D)。
A. 参数类型不同 B. 参数个数不同
C. 参数顺序不同 D. 函数返回值不同
7. 已知: int fun (int &a),m=10; 下列调用 fun()函数的语句中, 正确的是 (C)。
A. fun(&m); B. fun (m*2);
C. fun (m); D. fun (m++);
8. 在函数的引用调用中, 函数的实参和形参分别应是 (C)。
A. 变量值和变量 B. 地址值和指针
C. 变量名和引用 D. 地址值和引用
9. 在函数的传址调用中, 函数的实参和形参分别应是 (B)。
A. 变量值和变量 B. 地址值和指针
C. 变量名和引用 D. 地址值和引用
10. 说明一个内联函数时, 应加关键字是 (A)。
A. inline B. static
C. void D. extern

5.3 填空题

1. 如果将调用一个函数的过程分为 3 个步骤, 第 1 步是 转去执行被调用函数, 第 2 步是 执行被调用函数的语句, 第 3 步是返回操作。
2. 函数的存储类分两种, 它们分别是 外部 函数和内部函数, 其中 内部 函数的存储类说明不可省略, 该说明符是 static。
3. 使用关键字 inline 说明的函数称为 内联 函数, 具有相同函数名但具有不同实现的函数称为 重载 函数。
4. 在调用一个函数过程中可以直接或间接地调用该函数, 则该函数称为 被调用 函数。这种调用称为 嵌套 调用。
5. 在引用调用中, 函数实参用 变量名, 形参用 引用名。

5.4 分析下列程序的输出结果

1.

```
#include <iostream.h>
int fun(int n)
{
    int p(1),s(0);
```



```

        for(int i(1);i<n;i++)
        {
            p*=i;
            s+=p;
        }
        return s;
    }
void main()
{
    int s=fun(4);
    cout<<s<<endl;
}

```

答： 9

2.

```

#include <iostream.h>
void fun();
void main()
{
    for(int i(0);i<4;i++)
        fun();
}
void fun()
{
    int a(0);
    a++;
    static int b;
    b++;
    cout<<"a="<<a<<','<<"b="<<b<<endl;
}

```

答： a=1,b=1

a=1,b=2

a=1,b=3

a=1,b=4

3.

```

#include <iostream.h>
int fac(int);
void main()
{
    int s(0);
    for(int i(1);i<=5;i++)
        s+=fac(i);
    cout<<"5!+4!+3!+2!+1!="<<s<<endl;
}

```



课后习题答案网
—— 思路岛下载

```
int fac(int n)
{
    static int b=1;
    b*=n;
    return b;
}
```

答: $5!+4!+3!+2!+1!=153$

4.

```
#include <iostream.h>
int fun(char *,char *);
void main()
{
    char *p1,*p2;
    p1="abcfgy";
    p2="abcdhu";
    int n=fun(p1,p2);
    cout<<n<<endl;
}
int fun(char *s1,char *s2)
{
    while(*s1&&*s2&&*s1++=*s2++)
        ;
    s1--;
    s2--;
    return *s1-*s2;
}
```

答: 2

5.

```
#include <iostream.h>
void fun(int,int,int *);
void main()
{
    int a,b,c;
    fun(5,6,&a);
    fun(7,a,&b);
    fun(a,b,&c);
    cout<<a<<','<<b<<','<<c<<endl;
}
void fun(int i,int j,int *k)
{
    j+=i;
    *k=j-i;
}
```



课后习题答案网
—— 思路岛下载

答: 6,6,6



课后习题答案网
—— 思源岛下载

6.

```
#include <iostream.h>
int add(int,int =5);
void main()
{
    int a(8);
    cout<<"sum1="<<add(a)<<endl;
    cout<<"sum2="<<add(a,add(a))<<endl;
    cout<<"sum3="<<add(a,add(a,add(a)))<<endl;
}
int add(int i,int j)
{
    return i+j;
}
```

答: sum1=13
sum2=21
sum3=29

7.

```
#include <iostream.h>
void swap(int &,int &);
void main()
{
    int a(8),b(5);
    cout<<"a="<<a<<', '<<"b="<<b<<endl;
    swap(a,b);
    cout<<"a="<<a<<', '<<"b="<<b<<endl;
}
void swap(int &i,int &j)
{
    int t;
    t=i;
    i=j;
    j=t;
}
```

答: a=8,b=5
a=5,b=8

8.

```
#include <iostream.h>
void print(int),print(char),print(char *);
void main()
```

```

{
    int a(800);
    print(a);
    print('a');
    print("break");
}
void print(int i)
{
    cout<<i<<endl;
}
void print(char i)
{
    cout<<i<<endl;
}
void print(char *i)
{
    cout<<i<<endl;
}

```



课后习题答案网
—— 思路岛下载

答： 800

a

break

5.5 编程题

1. 从键盘上输入 8 个浮点数，编程求出其和以及平均值。要求写出求和以及平均值的函数。

答：编程如下：

```

#include <iostream.h>
double s,sum(double b[],int n),average(int n);
void main()
{
    double a[8];
    cout<<"输入 8 个 double 型数 ";
    for(int i=0;i<8;i++)
        cin>>a[i];
    s=sum(a,8);
    double ave=average(8);
    cout<<"SUM="<<s<<','<<"AVERAGE="<<ave<<endl;
}
double sum(double b[],int n)
{
    double sum=0;
    for(int i=0;i<n;i++)
        sum+=b[i];
    return sum;
}

```

```
double average(int n)
{
    return s/n;
}
```



2. 从键盘上输入 8 个整型数，编辑求出它们中间最大的数和最小的数。要求写出求最大数和最小数的函数。

答：编程如下：

```
#include <iostream.h>
int max(double b[],int n),min(double b[],int n);
void main()
{
    int a[8];
    cout<<"输入 8 个 int 型数 ";
    for(int i=0;i<8;i++)
        cin>>a[i];
    cout<<"最大数为 "<<max(a,8)<<','<<"最小数为 "<<min(a,8)<<endl;
}
int max(double b[],int n)
{
    int max=b[0];
    for(int i=0;i<n;i++)
        if(b[i]>max)
            max=b[i];
    return max;
}
int min(double b[],int n)
{
    int min=b[0];
    for(int i=0;i<n;i++)
        if(b[i]<min)
            min=b[i];
    return min;
}
```

3. 给定某个年、月、日，计算出这一天是属于该年的第几天。要求写出计算闰年的函数和计算日期的函数。

答：编程如下：

```
#include <iostream.h>
int sum_day(int,int),leap(int);
void main()
{
    int year,month,day;
    cout<<"请输入一个日期(yyyy mm dd): ";
```

```

        cin>>year>>month>>day;
        int days=sum_day(month,day);
        if(leap(year)&&month>2)
            days++;
        cout<<year<<"年"<<month<<"月"<<day<<"日是这一年的第"<<days<<"天。\\n";
    }
    int sum_day(int month,int day)
    {
        int day_month[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};
        for(int i=1;i<month;i++)
            day+=day_month[i];
        return day;
    }
    int leap(int year)
    {
        int leap=year%4==0&&year%100!=0||year%400==0;
        return leap;
    }

```

4. 使用递归方法将一个 n 位整数转换为一个字符串。编写转换函数。

答：编程如下：

```

#include <iostream.h>
void convert(int);
void main()
{
    int num;
    cout<<"请输入一个整型数:";
    cin>>num;
    cout<<"输出的字符串是";
    convert(num);
    cout<<endl;
}
void convert(int n)
{
    int a;
    if((a=n/10)!=0)
        convert(a);
    char c=n%10+'0';
    cout<<c;
}

```

5. 编写一个程序验证：任何一个充分大的偶数 (≥ 6) 总可以表示成两个素数之和。要求编写一个求素数的函数 `prine()`，它有一个 `int` 型参数，当参数值为素数时返回 1，否则返回 0。

答：编程如下：

```
#include <iostream.h>
int prime(int);
void main()
{
    int n;
    cout<<"请输入一个大于或等于 6 的偶数:";
    cin>>n;
    int i,j;
    cout<<"两个素数的所有可能有: \n";
    for(i=2;i<=n/2;i++)
    {
        j=n-i;
        if(prime(i)&&prime(j))
            cout<<"N="<<i<<'+ '<<j<<endl;
    }
}

int prime(int n)
{
    int j=2;
    while(n%j!=0)
        j++;
    if(n==j)
        return 1;
    else
        return 0;
}
```



课后习题答案网
—— 思路岛下载

5.6 简单回答下列问题

1. 没有返回值的函数是否有函数类型？函数参数类型与函数的类型是否有关系？

答：没有返回值的函数的类型为 **void**。没有关系。

2. 函数的缺省的存储类是什么？函数的存储类与函数什么关系？

答：函数的缺省的存储类是外部类。函数的存储类决定了函数的作用域。

3. 函数的调用方式有哪两种？传址和引用调用方式有何不同？

答：函数的调用方式有两种：传值调用和引用调用。传址调用时实参用地址值，形参用指针，引用调用时实参用变量名，形参用引用名。

4. 内联函数与一般函数有何不同？

答：内联函数的函数体是在编译时被替换，一般函数的函数体是在调用时被转去执行。前者可以提高速度，后者可以减少代码，各有利弊。

5. 对重载函数设置默认参数值，对重载函数的选择是否会有影响？

答：是会有影响的。

第6章 类和对象(一)

练习题 6

6.1 判断题

- × 1. 使用 `class` 定义的类，其默认访问权限是公有的，使用 `struct` 定义的类，其默认访问权限是私有的。
- × 2. 类中的成员函数都是公有的，数据成员都是私有的。
- × 3. 定义在类体内的成员函数是内联函数，定义在类体外的成员函数不能是内联函数。
- ✓ 4. 类定义后，它的成员个数及名称就不会再被改变了。
- × 5. 定义或说明对象时，系统会自动调用构造函数为创建的对象初始化。如果类中没有定义任何构造函数时，就无法给定义的对象初始化。
- ✓ 6. 定义一个对象时，系统只为该对象的数据成员开辟内存空间，而成员函数是同类对象共享的。
- ✓ 7. 对象成员的表示方法与结构变量成员的表示方法相同。
- ✓ 8. 创建对象时系统自动调用相应的构造函数为对象初始化，没有相应的构造函数时，系统会自动生成。
- ✓ 9. 构造函数是一个其名与类名相同的特殊的成员函数。
- × 10. 析构函数是一个函数体为空的成员函数。
- ✓ 11. 构造函数和析构函数都是系统自动调用的成员函数。
- × 12. 构造函数和析构函数都可以重载。
- ✓ 13. 成员函数与一般函数一样可以重载、内联和设置参数的默认值。
- × 14. 静态成员是指静态对象的成员。
- × 15. 静态数据成员必须在构造函数的成员初始化列表中进行初始化。
- ✓ 16. 静态成员都可以使用类名加作用域运算符的方法来引用。
- × 17. 静态成员函数中引用静态数据成员和非静态数据成员的方式是相同的。
- ✓ 18. 常成员指的是类体内使用 `const` 关键字说明的常数据成员和常成员函数。
- × 19. 常数据成员在常成员函数中的值是不允许改变的，而在非常成员函数中是允许改变的。
- ✓ 20. 常对象需要引用常成员函数，而不能引用非常成员函数。
- × 21. 常对象的数据成员都是常数据成员。
- ✓ 22. 友元函数是说明在类体内的非成员函数，它可以访问类中的所有成员。
- × 23. 可以把一个一般函数说明为某类的友元函数，也可以将某类的成员函数说明为另类的友元函数。
- ✓ 24. 友元类中的所有成员函数都是友元函数。
- ✓ 25. 类型转换函数是一种特殊的成员函数，定义时不加类型说明，无函数参数。
- ✓ 26. 单参数的构造函数具有类型转换的作用。

6.2 单选题

1. 下列关于类的定义格式的描述中，错误的是 (C)。
 - A. 类中成员有 3 种访问权限
 - B. 类的定义可分说明部分和实现部分
 - C. 类中成员函数都是公有的，数据成员都是私有的
 - D. 定义类的关键字通常用 `class`，也可用 `struct`
2. 下列关键字中，不属于定义类时使用的关键字是 (D)。
 - A. `class`
 - B. `struct`
 - C. `public`
 - D. `default`
3. 下列关于成员函数的描述中，错误的是 (A)。



- A. 成员函数的定义必须在类体外
 - B. 成员函数可以是公有的，也可以是私有的
 - C. 成员函数在类体外定义时，前加 `inline` 可为内联函数
 - D. 成员函数可以设置参数的默认值
4. 下列关于对象的描述中，错误的是（ D ）。
- A. 定义对象时系统会自动进行初始化
 - B. 对象成员的表示与 C 语言中结构变量的成员表示相同
 - C. 属于同一个类的对象占有内存字节数相同
 - D. 一个类所能创建对象的个数是有限制的
5. 下列关于构造函数的描述中，错误的是（ C ）。
- A. 构造函数可以重载
 - B. 构造函数名同类名
 - C. 带参数的构造函数具有类型转换作用
 - D. 构造函数是系统自动调用的
6. 下列关于析构函数的描述中，错误的是（ A ）。
- A. 析构函数的函数体都为空
 - B. 析构函数是用来释放对象的
 - C. 析构函数是系统自动调用的
 - D. 析构函数是不能重载的
7. 下列关于静态成员的描述中，错误的是（ C ）。
- A. 静态成员都是使用 `static` 来说明的
 - B. 静态成员是属于类的，不是属于某个对象的
 - C. 静态成员只可以用类名加作用域运算符来引用，不可用对象引用
 - D. 静态数据成员的初始化是在类体外进行的
8. 下列关于常成员的描述中，错误的是（ C ）。
- A. 常成员是用关键字 `const` 说明的
 - B. 常成员有常数据成员和常成员函数两种
 - C. 常数据成员的初始化是在类体内定义它时进行的
 - D. 常数据成员的值是不可以改变的
9. 下列关于友元函数的描述中，错误的是（ B ）。
- A. 友元函数不是成员函数
 - B. 友元函数只可访问类的私有成员
 - C. 友元函数的调用方法同一般函数
 - D. 友元函数可以是另一类中的成员函数
10. 下列关于类型转换函数的描述中，错误的是（ C ）。
- A. 类型转换函数是一种成员函数
 - B. 类型转换函数定义时不指出类型，也没有参数
 - C. 类型转换函数的功能是将函数名所指定的类型转换为该类类型
 - D. 类型转换函数在一个类中可定义多个

6.3 填空题

1. 类体内成员有 3 个访问权限，说明它们的关键字分别是 公有、私有 和 保护。
2. 使用 `class` 定义的表中，成员的默认访问权限是 私有 的；由 `struct` 定义的类中，成员的默认的访问权限是 公有 的。
3. 如果一个类中没有定义任何构造函数时，系统会自动提供一个 默认 构造函数；同样，类中没有定义析构函数时，系统会自动提供一个 默认 析构函数。
4. 静态成员是属于 类 的，它除了可以通过对象名来引用外，还可以使用 类名限定 来引用。

5. 友元函数是被说明在 类体 内的 非 成员函数。友元函数可访问该类中的成员。

6. 完成下列类的定义。

```
class A
{
public:
    A() { a=0; }
    int    int Geta(A &m);
private:
    int a;
};
int Geta(A &m)
{ return m.a; }
```

6.4 分析下列程序的输出结果

1.

```
#include <iostream.h>
class A
{
public:
    A()
    {
        a1=a2=0;
        cout<<"Default constructor called.\n";
    }
    A(int i,int j);
    ~A()
    { cout<<"Destructor called.\n"; }
    void Print()
    { cout<<"a1="<<a1<<', '<<"a2="<<a2<<endl; }
private:
    int a1,a2;
};
A::A(int i,int j)
{
    a1=i;
    a2=j;
    cout<<"Constructor called.\n";
}
void main()
{
    A a,b(5,8);
    a.Print();
    b.Print();
}
```

答: Default constructor called.

Constructor called.

a1=0,a2=0

a1=5,a2=8



课后习题答案网

——思路岛下载

Destructor called.

Destructor called.

2.

```
#include <iostream.h>
class B
{
public:
    B()
    { cout<<"+b<<endl; }
    ~B()
    { cout<<"b-<<endl; }
    static int Getb()
    { return b; }
private:
    static int b;
};
int B::b=10;
void main()
{
    B b1,b2,b3;
    cout<<"B::Getb()<<endl;
}
```

答: 11

12

13

13

13

12

11



课后习题答案网

—— 思源岛下载

3.

```
#include <iostream.h>
class Date
{
public:
    Date(int y,int m,int d)
    {
        year=y;
        month=m;
        day=d;
    }
    friend void Print(Date &);
private:
    int year,month,day;
};
void Print(Date &d)
{
    cout<<d.year<<"/"<<d.month<<"/"<<d.day<<endl;
```

```

    }
    void main()
    {
        Date d1(2005,10,1),d2(2005,12,9);
        Print(d1);
        Print(d2);
    }

```

答: 2005/10/1
2005/12/9

4.

```

#include <iostream.h>
class C
{
public:
    C(int i,int j)
    {   c1=i;c2=j;   }
    void Sum(C a,C b)
    {
        c1=a.c1+b.c1;
        c2=a.c2+b.c2;
    }
    void Print()
    {   cout<<"c1="<<c1<<','<<"c2="<<c2<<endl;   }
private:
    int c1,c2;
};
void main()
{
    C a(6,9);
    C b(a);
    C c(b);
    c.Sum(a,b);
    c.Print();
}

```

答: c1=12,c2=18

5.

```

#include <iostream.h>
class S
{
public:
    S()
    {   PC=0;   }
    S(S &s)
    {
        PC=s.PC;
        for(int i=0;i<PC;i++)
            elems[i]=s.elems[i];
    }
    void Empty()
    {   PC=0;   }
    int IsEmpty()
    {   return PC==0;   }
    int IsMemberOf(int n);
    int Add(int n);
    void Print();
private:
    int elems[100],PC;
};
int S::IsMemberOf(int n)

```

```

    {
        for(int i=0;i<PC;i++)
            if(elems[i]==n)
                return 1;
        return 0;
    }
int S::Add(int n)
{
    if(IsMemberOf(n))
        return 1;
    else if(PC==100)
        return 0;
    else
    {
        elems[PC++]=n;
        return 1;
    }
}
void S::Print()
{
    cout<<'{'<<endl;
    for(int i=0;i<PC-1;i++)
        cout<<elems[i]<<','<<endl;
    if(PC>0)
        cout<<elems[PC-1]<<endl;
    cout<<'}'<<endl;
}

void main()
{
    S a;
    cout<<a.IsEmpty()<<endl;
    a.Print();
    S b;
    for(int i=1;i<=5;i++)
        b.Add(i);
    b.Print();
    cout<<b.IsMemberOf(3)<<endl;
    cout<<b.IsEmpty()<<endl;
    for(i=6;i<=10;i++)
        b.Add(i);
    S c(b);
    c.Print();
}

```

答: 1
 {}
 {1,2,3,4,5}
 1
 0
 {1,2,3,4,5,6,7,8,9,10}

6.5 编程题

1. 按下列要求编程:

- (1) 定义一个描述矩形的类 `Rectangle`，包括的数据成员有宽 (`width`) 和长 (`length`);
- (2) 计算矩形周长;
- (3) 计算矩形面积;
- (4) 改变矩形大小。

通过实例验证其正确性。

答：编程如下：

```
#include <iostream.h>
class Rectangle
{
public:
    Rectangle(int a,int b)
    { width=a; length=b; }
    int Area()
    { return width*length; }
    int Periment()
    { return 2*(width+length); }
    void Changesize(int a,int b)
    { width=a; length=b; }
    void Print();
private:
    int width,length;
};
void Rectangle::Print()
{
    cout<<"AREA="<<Area()<<endl;
    cout<<"PERIMENT="<<Periment()<<endl;
}

void main()
{
    Rectangle r(5,8);
    r.Print();
    r.Changesize(3,9);
    r.Print();
}
```

2. 编程实现一个简单的计算器。要求从键盘上输入两个浮点数，计算出它们的加、减、乘、除运算的结果。

答：编程如下：

```
#include <iostream.h>
class ASMD
{
public:
    ASMD(double a,double b)
    { x=a; y=b; }
    void Addition()
    { cout<<x+y<<endl; }
    void Subtration()
    { cout<<x-y<<endl; }
    void Multiply()
    { cout<<x*y<<endl; }
    void Divison()
    { cout<<x/y<<endl; }
```

```

        { cout<<x/y<<endl; }
        void Print();
    private:
        double x,y;
};
void ASMD::Print()
{
    Addition();
    Subtration();
    Multiply();
    Divison();
}

void main()
{
    ASMD a(40,8);
    a.Print();
}

```

3. 编一个关于求多个某门功课总分和平均分的程序。具体要求如下:

- (1) 每个学生信息包括姓名和某门功课成绩。
- (2) 假设 5 个学生。
- (3) 使用静态成员计算 5 个学生的总成绩和平均分。

答: 编程如下:

```

#include <iostream.h>
#include <string.h>
class Student
{
public:
    Student(char s[],int gr)
    { strcpy(name,s); grade=gr; totalize+=gr; }
    static void Print();
private:
    char name[10];
    int grade;
    static int totalize;
};
int Student::totalize=0;
void Student::Print()
{
    cout<<"总成绩为 "<<totalize<<","平均成绩为 "<<totalize/5<<endl;
}

void main()
{
    Student s1("ma",85),s2("Wang",96),s3("Li",82),s4("lu",78),s5("zhang",80);
    Student::Print();
}

```

6.6 简单回答下列问题

1. 一个类中是否必须有用户定义的构造函数? 如果用户没有定义构造函数, 又如何对创建的对象初始化?

答：一个类用户可以不定义构造函数，这时系统自动提供一个默认的构造函数，并可用该构造函数对创建的对象初始化。

2. 拷贝构造函数具有几个参数？它有类型转换的作用吗？

答：拷贝构造函数具有一个参数，即为该类对象的引用。拷贝构造函数没有类型转换作用。

3. 静态成员属于类的，是否每个对象都可以引用该静态成员？

答：可以。

4. 常对象可以引用非常成员函数吗？非常对象可以引用常成员函数吗？

答：不可以。可以。

5. 友元函数能否访问类中的保护成员？友元函数访问类中私有成员与成员函数访问私有成员的形式相同吗？

答：能。相同。

第 7 章 类和对象(二)

练习题 7

7.1 判断题

- × 1. 定义对象指针时也要调用构造函数。
- ✓ 2. 对象指针可用同类对象的地址值给它赋值。
- × 3. 对象指针成员表示与对象引用相同。
- ✓ 4. 常类型指针有两种，一种是指针的地址值是常量，另一种是指针所指向的变量或对象是常量。
- × 5. 指向对象的常指针的地址值是可以改变的。
- × 6. 指向常对象的指针所指向的对象是可以改变的。
- ✓ 7. `this` 指针是系统生成的指向当前被某个成员函数操作对象的指针。
- × 8. 对象引用可以用一个同类对象的地址值对它赋值。
- × 9. 定义对象引用时，可以对其引用进行初始化，也可以不进行初始化。
- × 10. 对象数组的元素可以是不同类的对象。
- × 11. 对象指针数组可以使用不同类的对象的地址值进行初始化。
- × 12. 给对象数组元素赋值时都要创建临时对象。
- ✓ 13. 指向一维对象数组的指针是一个二级指针。
- × 14. 自身类对象可作该类的子对象。
- ✓ 15. 子对象的初始化要在构造函数的成员初始化列表中进行。
- ✓ 16. 使用 `new` 运算符创建的对象称为堆对象。
- ✓ 17. 任何对象都可以使用 `delete` 运算符来释放。
- × 18. 使用 `new` 运算符创建的对象数组其元素都具有默认值。
- ✓ 19. 类的作用域范围指的是类体内。
- × 20. 对象的存储类只有外部的和静态的。

7.2 单选题

1. `fl()` 函数是类 A 的公有成员函数，`p` 是指向类的成员函数 `fl()` 的指针，下列表示中正确的是 (C)。
- A. `p=fl()`
 - B. `p=fl`
 - C. `p=A::fl`
 - D. `p=A::fl()`
2. `p` 是指向类 A 数据成员 `a` 的指针，`a` 是类 A 的一个对象。在给 `a` 成员赋值为 5 的下列表

6. 使用new创建一个对象数组，该数组用指针pa指向，并使数组的 3 个元素都是类A的对象a1 的值。应该是 A a1,*pa; pa=new A[3]; pa[0]=pa[1]=pa[2]=a1;。
7. 在某个函数体内定义一个静态类的对象sa，应该是 static A sa;。
8. 定义一个类A的对象a1，并给它赋值，其实参值为 8，应该是 A a1(8);。

7.4 分析下列程序的输出结果

1.

```
#include <iostream.h>
class A
{
public:
    A(int i)
    { a=i; }
    A()
    {
        a=0;
        cout<<"Default constructor called."<<a<<endl;
    }
    ~A()
    { cout<<"Destructor called."<<a<<endl; }
    void Print()
    { cout<<a<<endl; }
private:
    int a;
};

void main()
{
    A a[4],*p;
    int n=1;
    p=a;
    for(int i=0;i<4;i++)
        a[i]=A(++n);
    for(i=0;i<4;i++)
        (p+i)->Print();
}
```

答: Default constructor called. 0
 Default constructor called. 0
 Default constructor called. 0
 Default constructor called. 0
 Destructor called. 2
 Destructor called. 3
 Destructor called. 4
 Destructor called. 5

2

3
4
5
Destructor called. 5
Destructor called. 4
Destructor called. 3
Destructor called. 2

2.

```
#include <iostream.h>
class B
{
public:
    B(int i)
    {   b=i;   }
    B()
    {
        b=0;
        cout<<"Default constructor called."<<b<<endl;
    }
    ~B()
    {   cout<<"Destructor called."<<b<<endl;   }
    void Print()
    {   cout<<b<<endl;   }

private:
    int b;
};

void main()
{
    B *pb[4];
    int n=1;
    for(int i=0;i<4;i++)
        pb[i]=new B(n++);
    for(i=0;i<4;i++)
        pb[i]->Print();
    for(i=0;i<4;i++)
        delete *(pb+i);
}
```

答: 1

2

3

4

Destructor called. 1
Destructor called. 2
Destructor called. 3
Destructor called. 4

3.

```
#include <iostream.h>
class C
{
public:
    C(int i)
    {   c=i;   }
    C()
    {
        c=0;
        cout<<"Default constructor called."<<c<<endl;
    }
    ~C()
    {   cout<<"Destructor called."<<c<<endl;   }
    void Print()
    {   cout<<c<<endl;   }
private:
    int c;
};
void main()
{
    C *p;
    p=new C[4];
    int n=1;
    for(int i=0;i<4;i++)
        p[i]=C(n++);
    for(i=0;i<4;i++)
        p[i].Print();
    delete []p;
}
```

答: Default constructor called. 0

Default constructor called. 0

Default constructor called. 0

Default constructor called. 0

Destructor called. 2

Destructor called. 3

Destructor called. 4

Destructor called. 5

2

3

4

5

Destructor called. 5

Destructor called. 4

Destructor called. 3

Destructor called. 2

4.

```
#include <iostream.h>
class D
{
public:
    D()
    {
        d1=d2=0;
        cout<<"Default constructor callrd.\n";
    }
    D(int i,int j)
    {
        d1=i; d2=j;
        cout<<"Constructor called."<<"d1="<<d1<<','<<"d2="<<d2<<endl;
    }
    ~D()
    {   cout<<"Destructor called."<<"d1="<<d1<<','<<"d2="<<d2<<endl;   }
    void Set(int i,int j)
    {   d1=i;d2=j;   }
private:
    int d1,d2;
};
void main()
{
    int n(10),m(20);
    D d[4]={D(5,7),D(3,6),D(7,9),D(1,4)};
    for(int i=0;i<4;i++)
        d[i].Set(n++,m++);
}
```

答: Constructor called. d1=5,d2=7
 Constructor called. d1=3,d2=6
 Constructor called. d1=7,d2=9
 Constructor called. d1=1,d2=4
 Destructor called. d1=13,d2=23
 Destructor called. d1=12,d2=22
 Destructor called. d1=11,d2=21
 Destructor called. d1=10,d2=20

5.

```
#include <iostream.h>
class E
{
public:
    E(int i,int j)
    {
        e1=i; e2=j;
        cout<<"Constructor called."<<"e1="<<e1<<','<<"e2="<<e2<<endl;
    }
    void FunE(E *e)
```

```

    {
        e1=e->e1;
        e2=e->e2;
        cout<<" Constructor called."<<"e1="<<e1<<','<<"e2="<<e2<<endl;
    }
    void FunE(E &e)
    {
        e1=e.e1;
        e2=e.e2;
        cout<<"In FunE(E &e)."<<"e1="<<e1<<','<<"e2="<<e2<<endl;
    }
private:
    int e1,e2;
};
void main()
{
    E a(5,6),b(3,4);
    a.FunE(&b);
    b.FunE(a);
}

```



课后习题答案网
—— 思路岛下载

答: Constructor called. e1=5,d2=6
 Constructor called. e1=3,e2=4
 In FunE(E &e). e1=3,e2=4
 In FunE(E &e). e1=3,e2=4

6.

```

#include <iostream.h>
class F
{
public:
    class G
    {
    public:
        G()
        {}
        G(int i)
        { g=i; }
        int GetValue()
        { return g; }
        void Print(F *p);
    private:
        int g;
    }myg;
    friend class G;
    F(int i,int j):myg(i)
    { f=j; }
private:
    int f;
};
void F::G::Print(F *p)
{

```

```

        cout<<p->f<<endl;
    }
    void main()
    {
        F::G g;
        F f(5,10);
        f.myg.Print(&f);
        g=f.myg;
        cout<<g.GetValue()<<endl;
    }

```

答: 10
5

7.5 编程题

1. 按下列要求实现一个栈类的操作。

该类名为 **Stack**，包括如下操作：

- (1) 压栈操作: **Push()**;
- (2) 弹栈操作: **Pop()**;
- (3) 获取栈顶元素: **Peer()**;
- (4) 判栈空操作: **IsEmpty()**;
- (5) 判栈满操作: **IsPull()**。

设栈最多可存放 50 个整数。

栈中成员用数组表示。

编写一个程序，定义一个栈类的对象数组来验证该类操作。

答：栈的正常操作程序如下：

```

#include <iostream.h>
#include <stdlib.h>
class Stack
{
public:
    Stack(int i);
    ~Stack()
    { delete [] sta; }
    void Push(int i);
    int Pop();
    void IsUpll()
    {
        if(tos==length)
        {
            cout<<"Stack is fill.\n";
            return;
        }
    }
    int IsEmpty()
    {
        if(tos==0)
        {
            cout<<"Stack underflow.\n";
            return 0;
        }
    }
private:
    int *sta;;

```

```

        int tos,length;
    };
    Stack::Stack(int i)
    {
        sta=new int[i];
        if(!sta)
        {
            cout<<"Can't allocate stack.\n";
            abort();
        }
        tos=0;
        length=i;
    }
    void Stack::Push(int i)
    {
        sta[tos]=i;
        tos++;
    }
    int Stack::Pop()
    {
        tos--;
        return sta[tos];
    }
    void main()
    {
        Stack a(50);
        a.Push(1);
        a.Push(2);
        a.Push(3);
        a.Push(4);
        a.Push(5);
        a.Push(6);
        a.Push(7);
        a.Push(8);
        a.Push(9);
        a.Push(10);
        cout<<a.Pop()<<' ';
        cout<<a.Pop()<<' ';
        cout<<a.Pop()<<' ';
        cout<<a.Pop()<<' ';
        cout<<a.Pop()<<' ';
        cout<<a.Pop()<<' ';
        cout<<a.Pop()<<' ';
        cout<<a.Pop()<<' ';
        cout<<a.Pop()<<' ';
        cout<<a.Pop()<<' '<<endl;
    }

```

判断栈的空、满操作，请读者修改上述程序。

2. 按下列要求实现一个有关学生成绩的操作。

该类名为 **Student**。

(1) 每个学生的信息包含有姓名（字符数组）和成绩（int 型）。

(2) 共有 5 个学生，用对象数组表示。

(3) 计算出 5 个学生中的最高分, 并输出姓名及分数。

答: 编程如下:

```
#include <iostream.h>
#include <string.h>
class Student
{
public:
    Student(char s[],int gr)
    {   strcpy(name,s); grade=gr; }
    friend void fun();
private:
    char name[10];
    int grade;
};
Student ss[5]={Student("马力",85),Student("王欣",96),Student("李明",82),
               Student("赵亮",78),Student("张京",80)};
void fun()
{
    int k=0;
    for(int i=0;i<5;i++)
        if(ss[i].grade>ss[0].grade)
            k=i;
    cout<<"最高分的学生姓名和成绩如下: \n"<<ss[k].name<<','<<ss[k].grade<<endl;
}
void main()
{
    fun();
}
```

3. 按如下要求编程验证子对象的有关操作。

(1) 定义两个类 A 和类 B。

(2) 在类 B 中有两个类 A 的对象 one, two。

验证如下事实:

(1) 在类 B 的构造函数中应该包含对两个类 A 的子对象的初始化项, 被放在成员初始化列表中。

(2) 在类 B 的默认构造函数中隐含着子对象的初始化项。

(3) 在类 B 的析构函数中也隐含着子对象的析构函数。

(4) 调用子对象构造函数的顺序。

答: 运行下列程序, 从输出结果中分析要验证的 4 个事实。

程序内容如下:

```
#include <iostream.h>
class A
{
public:
    A()
    {   cout<<"In A0.\n"; }
    A(int i)
    {   a=i; cout<<"In A1."<<a<<endl; }
    ~A()
    {   cout<<"In A2."<<a<<endl; }
}
```

```

        int a;
    };
    class B
    {
    public:
        B()
        { cout<<"In B0.\n"; }
        B(int i,int j,int k):two(j),one(k)
        { b=i; cout<<"In B1.\n"; }
        void Print()
        {
            cout<<b<<','<<one.a<<','<<two.a<<endl;
        }
        ~B()
        { cout<<"In B2.\n"; }
    private:
        int b;
        A one,two;
    };

    void main()
    {
        cout<<"构造函数的调用情况: \n";
        static B bb0;
        B bb(1,2,3);
        cout<<"输出对象 bb 的数据成员值: \n";
        bb.Print();
        cout<<"析构函数的调用情况: \n";
    }

```



7.6 简单回答下列问题

1. 对象指针可以指向一个有名对象，它可以指向一个无名对象吗？如何实现？

答：对象指针可以指向一个有名对象，也可以指向一个无名对象。例如，对象指针指向堆对象就是一个例子。假定已知类 A，定义该类对象指针 pa：

```

A *pa;
pa=new A(8);

```

pa 就是一个指向类 A 的无名对象的指针。

2. 对象数组和对象指针数组的区别在哪里？

答：对象数组的元素是同一个类的若干个对象，对象指针数组的元素是指向同一个类的若干个对象指针。

3. 在一个类中定义了多个子对象，其构造函数调用子对象的构造函数的顺序取决于什么？

答：构造函数调用子对象的顺序取决于定义子对象的顺序，而与构造函数的成员初始化列表中给出的子对象速设无关。

4. 使用 new 运算符创建的对象，如果不使用 delete 运算符释放，它们会在生存期到了时被系统释放吗？

答：不会。

5. 对象指针与对象引用作函数参数时具有相同的特点，为什么人们更喜欢使用对象引用

答：因为使用对象引用比使用对象指针更简单。

第 8 章 继承性和派生类

练习题 8

8.1 判断题

- × 1. 派生类只继承基类中的公有成员和保护成员，而不继承私有成员。
- × 2. 多重继承是指一个基类派生出多个派生类的情况。
- ✓ 3. 单重继承是指派生类只有一个基类的情况。
- ✓ 4. 派生类还可以作基类派生出新的派生类。
- ✓ 5. 派生类中成员的访问权限与基类的继承方式有关。
- × 6. 派生类中只包含直接基类的成员，不包含间接基类的成员。
- ✓ 7. 继承反映了类之间“是”的关系，组合反映了类之间“有”的关系。
- × 8. 基类中成员在派生类中都是可以访问的。
- × 9. 私有继承中基类的私有成员在派生类中还是私有的。
- ✓ 10. 保护继承方式下基类的保护成员在派生类仍是保护成员。
- × 11. 派生类的对象和派生类的派生类对派生类成员的访问权限是一样的。
- ✓ 12. 派生类的构造函数包含着直接基类的构造函数。
- × 13. 派生类的默认构造函数不包含有直接基类的构造函数。
- × 14. 派生类的析构函数中不包含直接基类的析构函数。
- × 15. 派生类是基类的子类型。
- × 16. 如果一个类是另一个类的子类型，则这个类的对象可以给另一个类的对象赋值，反之亦然。
- ✓ 17. 多重继承派生类的构造函数中应包含所有直接基类的构造函数。
- × 18. 多重继承的派生类构造函数中执行基类构造函数的顺序取决于该派生类构造函数的成员初始化列表中出现基类初始化项的顺序。

8.2 单选题

1. 下列关于继承的描述中，错误的是（ B ）。
 - A. 继承是重用性的重要机制
 - B. C++语言支持单重继承和双重继承
 - C. 继承关系不是可逆的
 - D. 继承是面向对象程序设计语言的重要特性
2. 下列关于基类和派生类的描述中，错误的是（ C ）。
 - A. 一个基类可以生成多个派生类
 - B. 基类中所有成员都是它的派生类的成员
 - C. 基类中成员访问权限继承到派生类中不变
 - D. 派生类中除了继承的基类成员还有自己的成员
3. 下列关于派生类的描述中，错误的是（ D ）。
 - A. 派生类至少有一个基类
 - B. 一个派生类可以作另一个派生类的基类
 - C. 派生类的构造函数中应包含直接基类的构造函数
 - D. 派生类默认的继承方式是 public
4. 派生类的对象可以直接访问的基类成员是（ A ）。

- A. 公有继承的公有成员
 - B. 保护继承的公有成员
 - C. 私有继承的公有成员
 - D. 公有继承的保护成员
5. 下列描述中，错误的是（ D ）。
- A. 基类的 protected 成员在 public 派生类中仍然是 protected 成员
 - B. 基类的 private 成员在 public 派生类中是不可访问的
 - C. 基类 public 成员在 private 派生类中是 private 成员
 - D. 基类 public 成员在 protected 派生类中仍是 public 成员
6. 派生类构造函数的成员初始化列表中，不能包含的初始化项是（ B ）。
- A. 基类的构造函数
 - B. 基类的子对象
 - C. 派生类的子对象
 - D. 派生类自身的数据成员
7. 下列关于子类型的描述中，错误的是（ B ）。
- A. 在公有继承下，派生类是基类的子类型
 - B. 如果类 A 是类 B 的子类型，则类 B 也是类 A 的子类型
 - C. 如果类 A 是类 B 的子类型，则类 A 的对象就是类 B 的对象
 - D. 在公有继承下，派生类对象可以初始化基类的对象引用
8. 下列关于多继承二义性的描述中，错误的是（ D ）。
- A. 一个派生类的多个基类中出现了同名成员时，派生类对同名成员的访问可能出现二义性
 - B. 一个派生类有多个基类，而这些基类又有一个共同的基类，派生类访问公共基类成员时，可能出现二义性
 - C. 解决二义性的方法是采用类名限定
 - D. 基类和派生类中同时出现同名成员时，会产生二义性

8.3 填空题

1. 继承的 3 种方式是 公有、私有 和 保护。
2. 如果类 A 继承了类 B，则类 A 被称为 派生 类，类 B 被称为 基 类。
3. 在保护继承方式下，基类的 public 成员成为派生类的 保护 成员，基类的 protected 成员成为派生类的 保护 成员。
4. 当一个派生类中含有子对象时，该派生类的析构函数中应包含 直接基类 的析构函数、子对象类的析构函数 和 自身类 的析构函数。
5. 派生类的构造函数的成员初始化列表中可以包含的初始化项有 基类的构造函数、子对象类的构造函数、常数据成员 和 派生类其他非静态数据成员。

8.4 分析下列程序的输出结果

1.

```
#include <iostream.h>
class A
{
public:
    A(int i,int j)
    { a1=i;a2=j; }
    void Move(int x,int y)
    { a1+=x;a2+=y; }
    void Print()
    { cout<<'('<<a1<<','<<a2<<')<<endl; }
private:
    int a1,a2;
};
class B:private A
{
public:
    B(int i,int j,int k,int l):A(i,j)
    { b1=k;b2=l; }
```

```

        void Print()
        {   cout<<b1<<','<<b2<<endl;   }
        void f()
        {   A::Print();   }
        void fun()
        {   Move(5,8);   }
    private:
        int b1,b2;
};
void main()
{
    A a(11,12);
    a.Print();
    B b(31,32,33,34);
    b.fun();
    b.Print();
    b.f();
}

```

答: (11,12)
33,34
(36,40)

2.

```

#include <iostream.h>
class A
{
    public:
        void InitA(int i,int j)
        {   a1=i;a2=j;   }
        void Move(int x,int y)
        {   a1+=x;a2+=y;   }
        int Geta1()
        {   return a1;   }
        int Geta2()
        {   return a2;   }
    private:
        int a1,a2;
};
class B:public A
{
    public:
        void InitB(int i,int j,int k,int l)
        {
            InitA(i,j);
            b1=k;
            b2=l;
        }
        void Move(int x,int y)
        {   b1+=x;b2+=y;   }
        int Getb1()
        {   return b1;   }
        int Getb2()
        {   return b2;   }
    private:
        int b1,b2;
};
class C:public B
{
    public:
        void fun()
        {   Move(10,15);   }
}

```

```

};

void main()
{
    C c;
    c.InitB(11,12,13,14);
    c.fun();
    cout<<c.Geta1()<<','<<c.Geta2()<<','<<c.Getb1()<<','<<c.Getb2()<<endl;
}

```

答: 11,12,23,29

3.

```

#include <iostream.h>
class A
{
public:
    A(int i):a(i)
    {   cout<<"A:constructor called.\n";   }
    ~A()
    {   cout<<"A:Destructor called.\n";   }
    void Print()
    {   cout<<a<<endl;   }
    int Geta()
    {   return a;   }
private:
    int a;
};

class B:public A
{
public:
    B(int i=0,int j=0):A(i),a(j),b(i+j)
    {   cout<<"B:Constructor called.\n";   }
    ~B()
    {   cout<<"B:Destructor called.\n";   }
    void Print()
    {
        A::Print();
        cout<<b<<','<<a.Geta()<<endl;
    }
private:
    int b;
    A a;
};

```

```

void main()
{
    B b1(8),b2(12,15);
    b1.Print();
    b2.Print();
}

```

答: A:constructor called.
 A:constructor called.
 B:Constructor called.
 A:constructor called.
 A:constructor called.
 B:Constructor called.
 8
 8,0
 12
 27,15
 B:Destructor called.
 A:Destructor called.
 A:Destructor called.
 B:Destructor called.
 A:Destructor called.
 A:Destructor called.

4.

```

#include <iostream.h>
class A
{
public:
    A(int i)
    { cout<<"Constructor in A."<<i<<endl; }
    ~A()
    { cout<<"Destructor in A.\n"; }
};
class B
{
public:
    B(int i)
    { cout<<"Constructor in B."<<i<<endl; }
    ~B()
    { cout<<"Destructor in B.\n"; }
};
class C

```

```

{
public:
    C(int i)
    { cout<<"Constructor in C."<<i<<endl; }
    ~C()
    { cout<<"Destructor in C.\n"; }
};
class D:public A,public B,public C
{
public:
    D(int i,int j,int k,int l):A(i),B(j),C(k),a(l)
    { cout<<"Constructor in D."<<l<<endl; }
    ~D()
    { cout<<"Destructor in D.\n"; }
private:
    A a;
};
void main()
{
    D d(3,4,5,6);
}

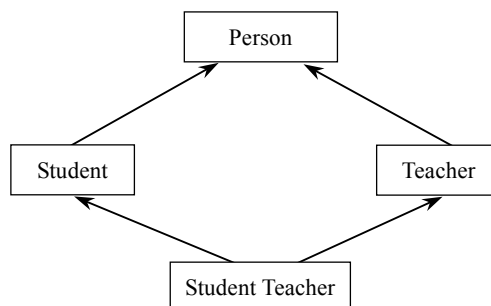
```

答: Constructor in A. 3
 Constructor in B. 4
 Constructor in C. 5
 Constructor in A. 6
 Constructor in D. 6
 Destructor in D.
 Destructor in A.
 Destructor in C.
 Destructor in B.
 Destructor in A.

8.5 编程题

1. 按下列要求编程: 按照右边图中所示的各类的关系, 编程输出它们的信息。各类中的数据成员如下:

Person: char *name(姓名), *dept(系别)
 Student: char *grade(年级)
 Teacher: char *lesson(授课名)
 Student Teacher(在职读研): char *major(专业方向)



答: 编程如下:



```
#include <iostream.h>
#include <string.h>
class Person
{
public:
    Person(char *s)
    {
        name=new char[strlen(s)+1];
        strcpy(name,s);
    }
    void Print()
    { cout<<"姓名: "<<name<<endl; }
protected:
    char *name;
};
class Student:virtual public Person
{
public:
    Student(char *s,char *t):Person(s)
    {
        grad=new char[strlen(t)+1];
        strcpy(grad,t);
    }
    void Print()
    {
        Person::Print();
        cout<<"年级: "<<grad<<endl;
    }
protected:
    char *grad;
};
class Teacher:virtual public Person
{
public:
    Teacher(char *s,char *t):Person(s)
    {
        lesson=new char[strlen(t)+1];
        strcpy(lesson,t);
    }
    void Print()
    {
        Person::Print();
        cout<<"授课名: "<<lesson<<endl;
    }
protected:
    char *lesson;
};
class StudentTeacher:public Student,public Teacher
{
public:
    StudentTeacher(char *p,char *q,char *r,char *s):Student(p,q),Teacher(p,r),Person(p)
    {
        Major=new char[strlen(s)+1];
```

```

        strcpy(Major,s);
    }
    void Print()
    {
        Student::Print();
        cout<<"专业方向: "<<Major<<endl;
    }
protected:
    char *Major;
};

void main()
{
    Student stu("马宁","06 级");
    Teacher tea("王新","高等代数");
    StudentTeacher stutea("刘力","07 级","数据结构","计算机软件");
    stu.Print();
    tea.Print();
    stutea.Print();
}

```

2. 设计一个程序，一行是信息，下一行画线，所画的线与信息行同长。例如，
C++

Programming

答： 编程如下：

```

#include <iostream.h>
#include <string.h>
class Line
{
public:
    Line(int i)
    { length=i; }
    void Show()
    {
        for(int i=0;i<length;i++)
            cout<<'_'<<endl;
    }
private:
    int length;
};
class Message
{
public:
    Message(char *ptr)
    {
        msg=new char[strlen(ptr)+1];
        strcpy(msg,ptr);
    }
    ~Message()
    { delete []msg; }
    void Show()
    {
        int i=0;

```

```

        cout<<msg<<endl;
    }
private:
    char *msg;
};
class MsgLine:public Line,public Message
{
public:
    MsgLine(char *ptr):Message(ptr),Line(strlen(ptr))
    { }
    void Show()
    {
        Message::Show();
        Line::Show();
    }
};

void main()
{
    MsgLine string1("C++");
    string1.Show();
    MsgLine string2("Progrmming");
    string2.Show();
}

```

8.6 简单回答下列问题

1. 在继承关系中，派生类中包含基类所有成员，基类是否也包含派生类的部分成员？
答：不包含。
2. 构造函数不能继承，派生类的构造函数中是否应包含直接基类的构造函数和所有间接基类的构造函数？
答：派生类的构造函数中只包含直接基类的构造函数。
3. 派生类的析构函数中不包含直接基类的析构函数，对吗？
答：不对。
4. 派生类的对象可以给基类对象赋值吗？
答：一般情况下不可以，只有在公有继承的情况下可以。
5. 多重继承的二义性可以避免吗？
答：可以避免，通常使用类名限定。

第9章 多态性和虚函数

练习题 9

9.1 判断题

- ✓ 1. 运算符重载是通过已有的运算符重新定义操作功能来实现的，它是 C++ 语言多态性的一种表现。
- ✓ 2. 所有的运算符都可以重载。
- ✓ 3. 运算符重载只可以是类的成员函数和友元函数，不可以是普通函数。
- ✓ 4. 运算符重载是通过函数来实现的，定义运算符重载函数时，函数名中要使用关键字

operator。

- ✓ 5. 用于类对象的运算符中只有取地址运算符 (&) 有默认的可以使用。
- ✓ 6. 运算符重载后, 优先级、结合性和操作数都是不变的。
- ✓ 7. 重载运算符的函数也可以设置默认参数。
- ✓ 8. 使用成员函数方法和使用友元函数的方法所定义的重载运算符函数的参数个数是不相同的。
- ✓ 9. 静态联编和动态联编都是在编译时进行的, 二者的区别仅是前者对非虚函数, 后者对虚函数。
- ✓ 10. 只要是成员函数就可以说明为虚函数, 因为虚函数是一种成员函数。
- ✓ 11. 虚函数有继承性, 基类中说明的虚函数只要在它的派生类中与它名字相同的, 一定是虚函数。
- ✓ 12. 虚函数可以被类的对象调用, 也可以被类的对象指针和对象引用调用。
- ✓ 13. 动态联编指的是在运行期间来选择不同类的虚函数。
- ✓ 14. 虚函数是实现动态联编的充分必要条件。
- ✓ 15. 含有纯虚函数的类称为抽象类, 与抽象类相对应的是具体类。
- ✓ 16. 抽象类可以定义对象, 不可以定义对象指针和对象引用。
- ✓ 17. 成员函数和构造函数调用虚函数都可以实现动态联编。
- ✓ 18. 析构函数可以说明为虚函数, 而构造函数说明为虚函数没有意义。
- ✓ 19. 抽象类的派生类一定是具体类。
- ✓ 20. 一个抽象类中可以包含有多个纯虚函数, 一个派生类中也可以包含多个虚函数。

9.2 单选题

1. 下列运算符中, 不可以重载的是 (D)。
 - A. &&
 - B. &
 - C. []
 - D. ?:
2. 下列关于运算符重载的描述中, 错误的是 (B)。
 - A. 运算符重载不改变优先级
 - B. 运算符重载后, 原来运算符操作不可再用
 - C. 运算符重载不改变结合性
 - D. 运算符重载函数的参数个数与重载方式有关
3. 下列关键字中, 用来说明虚函数的关键字是 (C)。
 - A. inline
 - B. operator
 - C. virtual
 - D. public
4. 下列的成员函数中, 纯虚函数是 (A)。
 - A. virtual void f1() = 0
 - B. void f1() = 0;
 - C. virtual void f1() {}
 - D. virtual void f1() == 0;
5. 含有一个或多个纯虚函数的类称为 (A)。
 - A. 抽象类
 - B. 具体类
 - C. 虚基类
 - D. 派生类
6. 下列关于虚函数的描述中, 错误的是 (C)。
 - A. 虚函数是一个成员函数
 - B. 虚函数具有继承性
 - C. 静态成员函数可以说明为虚函数
 - D. 在类的继承的层次结构中, 虚函数是说明相同的函数
7. 下列各种类中, 不能定义对象的类是 (B)。
 - A. 派生类
 - B. 抽象类
 - C. 嵌套类
 - D. 虚基类
8. 下列关于抽象类的描述中, 错误的是 (D)。

- A. 抽象类中至少应该有一个纯虚函数 B. 抽象类可以定义对象指针和对象引用
 C. 抽象类通常用作类族中最顶层的类 D. 抽象类的派生类必定是具体类
9. 一个类的层次结构中，定义有虚函数，并且都是公有继承，在下列情况下，实现动态联编的是（ D ）。
- A. 使用类的对象调用虚函数
 B. 使用类名限定调用虚函数，其格式如下：<类名>::<虚函数名>
 C. 使用构造函数调用虚函数
 D. 使用成员函数调用虚函数
10. 下列关于动态联编的描述中，错误的是（ C ）。
- A. 动态联编是函数联编的一种方式，它是在运行时来选择联编函数的
 B. 动态联编又可称为动态多态性，它是 C++ 语言中多态性的一种重要形式
 C. 函数重载和运算符重载都属于动态联编
 D. 动态联编只是用来选择虚函数的

9.3 填空题

1. C++ 语言多态性主要表现在动态联编、函数重载和 运算符重载。
 2. 运算符重载函数的两种主要方式是 成员函数和 友元函数。
 3. 静态联编支持的多态性称为 静态多态性，它是在 编译时进行的；动态联编支持的多态性称为 动态多态性，它是在 运行时进行的。
 4. 虚函数是一种 非静态成员函数。说明方法是在函数名前加关键字 virtual。虚函数具有 继承性，在基类中被说明的虚函数，具有相同说明的函数在派生类中自然是虚函数。
 6. 含有 纯虚函数 的类称为抽象类。它不能定义对象，但可以定义 对象指针和 对象引用。

9.4 分析下列程序的输出结果

1.

```
#include <iostream.h>
class Matrix
{
public:
    Matrix(int r,int c)
    {
        row=r;
        col=c;
        elem=new double[row*col];
    }
    double & operator()(int x,int y)
    { return elem[x*col+y]; }
    ~Matrix()
    { delete []elem; }
    void print(int i)
    { cout<<elem[i]; }
private:
    double *elem;
    int row,col;
};

void main()
{
    Matrix m(3,4);
    for(int i=0;i<3;i++)
        for(int j=0;j<4;j++)
            m(i,j)=4*i+j;
    for(i=0;i<3;i++)
        for(int j=0;j<4;j++)
```

```

        {
            m.print(4*i+j);
            cout<<" ";
        }
        cout<<endl;
    }
}

```

答: 0 1 2 3 4 5 6 7 8 9 10 11

2.

```

#include <iostream.h>
class A
{
public:
    virtual void fun()
    {   cout<<"A::fun() called.\n";   }
};
class B:public A
{
    void fun()
    {   cout<<"B::fun() called.\n";   }
};
void ffun(A *pa)
{
    pa->fun();
}
void main()
{
    A *pa=new A;
    ffun(pa);
    B *pb=new B;
    ffun(pb);
}

```

答: A::fun() called.
B::fun() called.

3.

```

#include <iostream.h>
class A
{
public:
    A()
    {   ver='A';   }
    virtual void print()
    {   cout<<"The A version "<<ver<<endl;   }
protected:
    char ver;
};
class B1:public A
{
public:
    B1(int i)
    {   info=i;ver='B';   }
    void print()
    {   cout<<"The B1 info: "<<info<<" version "<<ver<<endl;   }
private:
    int info;
};
class B2:public A
{
public:
    B2(int i)

```

```

        { info=i; }
        void print()
        { cout<<"The B2 info: "<<info<<" version "<<ver<<endl; }
    private:
        int info;
};
class B3:public B1
{
    public:
        B3(int i):B1(i)
        { info=i;ver='C'; }
        void print()
        { cout<<"The B3 info: "<<info<<" version "<<ver<<endl; }
    private:
        int info;
};
void print_info(A *pa)
{
    pa->print();
}
void main()
{
    A a;
    B1 b1(14);
    B2 b2(88);
    B3 b3(65);
    print_info(&a);
    print_info(&b1);
    print_info(&b2);
    print_info(&b3);
}

```

答: The A version A
 The B1 info: 14 version B
 The B2 info: 88 version A
 The B3 info: 65 version C

4.

```

#include <iostream.h>
class B
{
    public:
        virtual void fun1()
        { cout<<"B::fun1().\n"; }
        virtual void fun2()
        { cout<<"B::fun2().\n"; }
        void fun3()
        { cout<<"B::fun3().\n"; }
        void fun4()
        { cout<<"B::fun4().\n"; }
};
class D:public B
{
    public:
        void fun1()
        { cout<<"D::fun1().\n"; }
        void fun2()
        { cout<<"D::fun2().\n"; }
        void fun3()
        { cout<<"D::fun3().\n"; }
        void fun4()
        { cout<<"D::fun4().\n"; }
};

```



课后习题答案网
 —— 思路岛下载

```

void main()
{
    B *pb;
    D d;
    pb=&d;
    pb->fun1();
    pb->fun2();
    pb->fun3();
    pb->fun4();
}

```

答: D::fun1().
D::fun2().
B::fun3().
B::fun4().

5.

```

#include <iostream.h>
class A
{
public:
    A()
    { cout<<"In A cons.\n"; }
    virtual ~A()
    { cout<<"In A des.\n"; }
    virtual void f1()
    { cout<<"In A f1().\n"; }
    void f2()
    { f1(); }
};
class B:public A
{
public:
    B()
    { f1();cout<<"In B cons.\n"; }
    ~B()
    { cout<<"In B des.\n"; }
};
class C:public B
{
public:
    C()
    { cout<<"In C cons.\n"; }
    ~C()
    { cout<<"In C des.\n"; }
    void f1()
    { cout<<"In C f1().\n"; }
};

```

```

void main()
{
    A *pa=new C;
    pa->f2();
    delete pa;
}

```

答: In A cons.
In A f1().
In B cons.
In C cons.
In C f1().
In C des.
In B des.
In A des.

6.

```
#include <iostream.h>
class A
{
public:
    virtual void print()=0;
};
class B:public A
{
public:
    void print()
    { cout<<"In B print().\n"; }
};
class C:public B
{
public:
    void print()
    { cout<<"In C print().\n"; }
};
void fun(A *pa)
{
    pa->print();
}
void main()
{
    A *pa;
    B b;
    C c;
    pa=&b;
    fun(pa);
    pa=&c;
    fun(pa);
}
```

答: In B print().
In C print().

9.5 编程题

1. 编程求圆、圆内接正方形和圆外切正方形的面积和周长。要求使用抽象类。

答: 编程如下:

```
#include <iostream.h>
const double PI=3.1415;
class Shape
{
public:
    Shape(double i)
    { r=i; }
    virtual void Area()=0;
    virtual void Perimeter()=0;
protected:
    double r;
};
class Circle:public Shape
{
public:
    Circle(double i):Shape(i)
    { }
    void Area()
```

```

        { cout<<"圆的面积是 "<<PI*r*r<<endl; }
    void Perimeter()
    { cout<<"圆的周长是 "<<2*PI*r<<endl; }
};
class In_Square:public Shape
{
public:
    In_Square(double i):Shape(i)
    { }
    void Area()
    { cout<<"圆内接正方形的面积是 "<<2*r*r<<endl; }
    void Perimeter()
    { cout<<"圆内接正方形的周长是 "<<4*1.414*r<<endl; }
};
class Ex_Square:public Shape
{
public:
    Ex_Square(double i):Shape(i)
    { }
    void Area()
    { cout<<"圆外切正方形的面积是 "<<4*r*r<<endl; }
    void Perimeter()
    { cout<<"圆外切正方形的周长是 "<<8*r<<endl; }
};

void main()
{
    Shape *ps;
    ps=new Circle(8);
    ps->Area();
    ps->Perimeter();
    ps=new In_Square(8);
    ps->Area();
    ps->Perimeter();
    ps=new Ex_Square(8);
    ps->Area();
    ps->Perimeter();
    delete ps;
}

```

2. 在类的多层次继承结构中，类之间哪些函数是按作用域规则处理的？哪些函数是按多态性规则处理的？试编程说明之。

答：一般成员函数是按作用域规则处理的，虚函数是按多态性规则中的动态联编处理的。

下列程序中有一般成员函数和虚函数。

```

#include <iostream.h>
class A
{
public:
    void Print()
    { cout<<"In A.\n"; }
    virtual void fun()
    { cout<<"virtual A.\n"; }
}

```

```

};
class B:public A
{
public:
    void Print()
    {   cout<<"In B.\n";   }
    virtual void fun()
    {   cout<<"virtual B.\n";   }
};
void text(A &a)
{   a.fun();   }

void main()
{
    A a;
    B b;
    a.Print();
    b.Print();
    text(b);
}

```

9.6 简单回答下列问题

1. 运算符重载使用成员函数方法和友元函数方法是否都可以？并且是没有区别的吗？

答：两种方法都可以。有区别，参数个数不同。

2. 运算符重载实际上通过函数来重新定义运算符的功能，运算符重载的功能直接通过函数调用是否可以？

答：可以。

3. 多态性中对函数的选择从时间上来区分有哪两种方式？

答：一种是在编译时选定函数，称静态联编方式，另一种是在运行时选定函数，称动态联编方式。

4. 有虚函数是否就一定是动态联编？非虚函数是否就一定是静态联编？

答：不一定。一定是。

5. 在多层次的继承结构中，基类与派生类中存在着虚函数，这时调用虚函数就一定实现动态联编吗？

答：不一定。

第 10 章 C++语言的输入/输出操作

练习题 10

10.1 判断题

- ✓ 1. C++语言提供的文件操作包含文本文件和二进制文件。
- × 2. 预定义的提取符和插入符不可以再重载。
- ✓ 3. C++语言中的读写函数，不仅可用于标准文件，也可用于磁盘文件。

- × 4. 使用插入符输出一个字符串与使用 `write()` 成员函数输出一个字符串是没有差别的。
- × 5. 使用成员函数 `put()` 一次可写入一个字符串。
- × 6. 读取一个字符的成员函数 `get()` 是不可以带参数的。
- ✓ 7. 读取一行字符的成员函数 `getline()` 具有一个默认的参数值 `'\n'`。
- × 8. 使用 `read()` 成员函数一次只能读取一行字符。
- ✓ 9. 输出数据的默认对齐方式是右对齐。
- ✓ 10. 使用流对象的成员函数进行格式输出要比使用控制符进行格式输出复杂些, 因为成员函数需要对象调用, 而控制符可直接使用。
- ✓ 11. 在非标准文件操作中, 应该先定义流对象, 再打开文件, 必须使用 `open()` 函数。
- ✓ 12. 关闭打开的文件时使用成员函数 `close()`, 该函数无参数。
- ✓ 13. 进行二进制文件操作时, 在打开文件方式中增加 `ios::binary` 选项。
- × 14. 在定位文件的读/写指针的操作中, 只能向增加字节数的方向移动, 不能向减少字节数的方向移动。
- ✓ 15. C++ 语言的文件操作中, 既可以顺序读写, 又可以随机读写。
- × 16. 读/写指针相关函数 `tellp()` 的返回值是一个 `long int` 型数, 这表示当前读指针距文件尾的字节数。
- ✓ 17. 在文件操作中, 通常使用打开文件的流对象的值来判断打开文件是否成功。
- × 18. 文件结束函数 `eof()` 返回值为 0 时, 表示文件结束。

10.2 单选题

1. C++ 语言程序中进行文件操作时应包含的头文件是 (A)。
 - A. `fstream.h`
 - B. `math.h`
 - C. `stdlib.h`
 - D. `strstream.h`
2. C++ 语言程序中进行字符串流操作时应包含的头文件是 (D)。
 - A. `fstream.h`
 - B. `math.h`
 - C. `stdlib.h`
 - D. `strstream.h`
3. C++ 语言程序中使用控制符进行格式输出时应包含的头文件是 (B)。
 - A. `fstream.h`
 - B. `iomanip.h`
 - C. `math.h`
 - D. `strstream.h`
4. 下列各语句是输出字符 'A' 的, 其中错误语句是 (D)。
 - A. `cout<<'A';`
 - B. `cout.put('A');`
 - C. `char ch='A';cout<<ch;`
 - D. `cout<<put('A');`
5. 在 `ios` 类中提供的控制格式的标志位中, 八进制形式的标志位是 (C)。
 - A. `hex`
 - B. `dec`
 - C. `oct`
 - D. `basefield`
6. 在打开磁盘文件的访问方式常量中, 用来以追加方式打开文件的是 (C)。
 - A. `in`
 - B. `out`
 - C. `ate`
 - D. `app()`
7. 在下列读写函数中, 进行写操作的函数是 (C)。
 - A. `get()`
 - B. `read()`
 - C. `put()`
 - D. `getline()`
8. 已知文本文件 `abc.txt`, 以读方式打开, 下列的操作中错误的是 (D)。
 - A. `fstream infile ("abc.txt", ios::in);`
 - B. `ifstream infile ("abc.txt");`
 - C. `ofstream infile ("abc.txt");`
 - D. `fstream infile; infile.open("abc.txt", ios::in);`
9. 已知: `ifstream input`; 下列写出的语句中, 将 `input` 流对象的读指针移到距当前位置后 (文件尾方向) 100 个字节处的语句是 (C)。
 - A. `input.seekg(100, ios::beg);`
 - B. `input.seekg(100, ios::cur);`
 - C. `input.seekg(-100, ios::cur);`
 - D. `input.seekg(100, ios::end);`

10.3 填空题

1. 在格式输出的标志字中, 设置格式对齐位的有 left、right 和 internal。
2. 在格式输出中, 设置和清除格式标志位的成员函数分别是 setf() 和 unsetf()。
3. 系统规定与标准设备对应的 4 个流对象是 cin、cout、cerr 和 clog。
4. 在控制符中, 用来指定八进制、十进制和十六进制的控制符分别是 oct、dec 和 hex。
5. 在定位读/写指针的带有两个参数的函数中, 表示相对位置方式的 3 个常量是 cur、beg 和 end。

10.4 分析下列程序的输出结果

1.

```
#include <iostream.h>
#include <iomanip.h>
void main()
{
    int a=234;
    cout<<oct<<a<<endl;
    cout<<hex<<a<<endl;
    cout<<dec<<a<<endl;
    cout<<setfill('*')<<setw(8)<<a<<"ok"<<endl;
    double b=1.234567;
    cout<<b<<endl;
    cout<<setw(8)<<setprecision(4)<<b<<endl;
}
```

答: 352

```
ea
234
*****234ok
1.23457
***1.235
```

2.

```
#include <iostream.h>
#include <iomanip.h>
ostream &out1(ostream &outs)
{
    outs.setf(ios::left);
    outs<<setw(8)<<oct<<setfill('#');
    return outs;
}
void main()
{
    int a=123;
    cout<<a<<endl;
    cout<<out1<<a<<endl;
}
```

答: 123

```
173#####
```

3.

```
#include <iostream.h>
#include <iomanip.h>
void main()
{
    for(int i=1;i<6;i++)
        cout<<setfill(' ')<<setw(i)<<' '<<setfill('W')
            <<setw(11-2*i)<<'W'<<endl;
```

答:

```

}
WWWWWWWWW
WWWWWWWW
WWWWW
WWW
W

```

4.

```

#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
void main()
{
    fstream inf,outf;
    outf.open("my.dat",ios::out);
    if(!outf)
    {
        cout<<"Can't open file!\n";
        abort();
    }
    outf<<"abcdef"<<endl;
    outf<<"123456"<<endl;
    outf<<"ijklmn"<<endl;
    outf.close();
    inf.open("my.dat",ios::in);
    if(!inf)
    {
        cout<<"Can't open file!\n";
        abort();
    }
    char ch[80];
    int a(1);
    while(inf.getline(ch,sizeof(ch)))
        cout<<a++<<":"<<ch<<endl;
    inf.close();
}

```

答: 1:abcdef
2:123456
3:ijklmn

5.

```

#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
void main()
{
    fstream f;
    f.open("my1.dat",ios::out|ios::in);
    if(!f)
    {
        cout<<"Can't open file!\n";
        abort();
    }
    char ch[]="abcdefg1234567.\n";
    for(int i=0;i<sizeof(ch);i++)
        f.put(ch[i]);
    f.seekg(0);
    char c;
    while(f.get(c))
        cout<<c;
    f.close();
}

```

```
}  
答: abcdefg1234567.
```

6.

```
#include <iostream.h>  
#include <fstream.h>  
#include <stdlib.h>  
struct student  
{  
    char name[20];  
    long int number;  
    int totalscore;  
}stu[5]={"Li",502001,287,"Gao",502004,290,"Yan",5002011,278,"Lu",502014,285,"Hu",502023,279};  
void main()  
{  
    student s1;  
    fstream file("my3.dat",ios::out|ios::in|ios::binary);  
    if(!file)  
    {  
        cout<<"Can't open file!\n";  
        abort();  
    }  
    for(int i=0;i<5;i++)  
        file.write((char *)&stu[i],sizeof(student));  
    file.seekp(sizeof(student)*2);  
    file.read((char *)&s1,sizeof(stu[i]));  
    cout<<s1.name<<"\t"<<s1.number<<"\t"<<s1.totalscore<<endl;  
    file.close();  
}
```

答: Yan 5002011 278

10.5 编程题

1. 计算从键盘输入的字符串中子串"xy"出现的次数。

提示: 可使用函数 `peek()` 返回输入流中的下一个字符, 并不提取该字符。

答: 编程如下:

```
#include <iostream.h>  
void main()  
{  
    int ch,n=0;  
    cout<<"输入含有若干个 xy 子串的字符序列, 以<Ctrl+z>结束: \n";  
    while((ch=cin.get())!=EOF)  
        if(ch=='x'&&cin.peek()=='y')  
            n++;  
    cout<<"出现 xy 子串的次数为 " <<n<<endl;  
}
```

2. 编程统计一个文本文件中字符的个数。

答: 编程统计已存文件 `abc.txt` 中字符个数, 程序如下:

```
#include <iostream.h>  
#include <fstream.h>  
#include <stdlib.h>  
void main()  
{  
    fstream f;  
    f.open("abc.txt",ios::in);  
    if(!f)
```

```

    {
        cout<<"abc.txt can't open.\n";
        abort();
    }
    char ch;
    int n=0;
    while(!f.eof())
    {
        f.get(ch);
        n++;
    }
    cout<<"该文件字符数为 " <<n<<endl;
    f.close();
}

```

3. 编程给一个文件的所有行上加行号，并保存到另一个文件中。

答：将 del.cpp 文件编写行号后存入 del22.cpp 文件中，程序如下：

```

#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
void main()
{
    fstream inf,outf;
    inf.open("del.cpp",ios::in);
    if(!inf)
    {
        cout<<"Can't open.\n";
        abort();
    }
    outf.open("del22.cpp",ios::out);
    if(!outf)
    {
        cout<<"Can't open.\n";
        abort();
    }
    char s[80];
    int n=1;
    while(!inf.eof())
    {
        inf.getline(s,sizeof(s));
        outf<<n++<<": "<<s<<endl;
    }
    inf.close();
    outf.close();
}

```

10.6 使用 C++语句实现下列各种要求

1. 设置标志使得十六进制数中字母按大写格式输出。

答： cout.setf(ios::hex,ios::basefield);
cout.setf(ios::uppercase);

2. 设置标志以科学记数法显示浮点数。

答: `cout.setf(ios::scientific,ios::floatfield);`

3. 按右对齐方式,域为 5 位,输出常整型数 123,并使用'#'填充空位。

答: `cout.setf(ios::right,ios::adjustfield);`
`cout.width(5);`
`cout.fill('#');`
`const int a=123;`
`cout<<a<<endl;`

4. 按域宽为 i,精度为 j (i 和 j 为 int 型数),输出显示浮点数 d。

答: `int j=10,j=5;`
`cout.width(i);`
`cout.precision(j);`
`cout<<d<<endl;`

5. 使用前导 0 的格式显示输出域宽为 10 的浮点数 1.2345。

答: `cout.width(10);`
`double d=1.2345;`
`cout.fill('0');`
`cout<<d<<endl;`



www.sld.net.cn

课后习题答案网

—— 思源岛下载