

## 第一章 程序设计与 C++ 语言初步

### 填空题、选择题

1 程序设计方法大致可分为两类 A 和 B , 其中A将复杂的问题分解成许多小的、简单的问题来求解, 而B则将已经求解的小问题组合起来以得到复杂问题的解, 然而许多程序设计方法是这两者的结合。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 自顶向下方法; B 自底向上方法

2 很早就出现了许多程序设计语言, 其中 A 主要用于科学计算领域, B 主要应用商业事务处理领域, C 是一种通用的算法语言, 而 D 则主要面向初学者。

可选项

PASCAL

BASIC

C++

LISP

COBOL

FORTRAN

ADA

ALGOL

PROLOG

FOXBASE

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A ; B ; C ; D

3 结构化程序是一种单入口 / 单出口程序, 有人证明了只用三种基本控制结构就可实现这种程序, 这三种基本控制结构是 A、B、C。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 顺序; B 选择; C 循环

4 面向对象程序设计建立在结构化程序设计的基础上, 它围绕 A 来设计程序, 而不是操作本身, 且以 B 为构造程序的基本单位, 具有 C、D、E、F 等特点。

难易程度 B 中等

认知程度 识记

预计答题时间 1 分钟

参考答案 A 被操作的数据; B 类; C 封装; D 数据抽象; E 继承; F 多态性

5 定义一种程序设计语言必须描述它的 A 和 B。其中A刻划什么样的符号串是有效的程序, 而B描述的是用这种语言编写的程序的含义。其中A较常用的定义方法包括 C 和 D 两种。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 语法; B 语义; C BNF; D 语法图

6 我们在微机上使用的C++语言是建立在 A 字符集之上的, 而当前流行的还有IBM

提出的\_\_B\_\_字符集，APL语言就使用这种字符集。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A ASCII 编码；B EBCDIC 编码；

7 一个用C++语言编写的程序是由若干\_\_A\_\_再加上一个主函数组成，必须经过\_\_B\_\_、\_\_C\_\_、\_\_D\_\_等过程才能运行。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 类或函数；B 编辑；C 编译；D 链接

8 C++程序在编译时可能出现两种类型的错误 \_\_A\_\_和\_\_B\_\_，如果出现前一类错误则无法生成目标代码。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 严重错误；B 警告错误

问答题

9 试说明算法、实体与程序之间的关系。

难易程度 C 较难

认知程度 理解

预计答题时间 5 分钟

参考答案

10 类与对象之间有什么关系，两者又有何区别？

难易程度 B 中等

认知程度 理解

预计答题时间 2 分钟

参考答案 对象是现实世界中的实体在计算机中的表示，对象具有属性和行为。类是具有共同属性和行为的对象的抽象描述，将对象划分成类依赖于问题领域的知识。类刻划一组对象的共同属性和行为，对象是类的具体实例。

11 给出十进制数 496 的二进制、八进制和十六进制的表示。

难易程度 B 中等

认知程度 简单应用

预计答题时间 3 分钟

参考答案  $496_{10}=(111110000)_2=(760)_8=(1F0)_{16}$

12 我们可将仓库中存储的商品看成实体，试描述该实体。

难易程度 B 中等

认知程度 理解

预计答题时间 10 分钟

参考答案 库存商品实体可如下描述

实体	库存商品
属性	商品品号、名称、规格型号、库存数量 qty、库存金额 cost、入库时间
行为	1) 入库 输入 入库数量 in_qty、入库单价 in_price 输出 无 步骤 1 令 $qty = qty + in\_qty$ , $cost = cost + in\_qty * in\_price$ 并终止 2) 出库 输入 出库数量 out_qty 输出 实际出库数量 out_qty、实际出库金额 out_cost 步骤 1 如果 $out\_qty > qty$ 则令 $out\_qty = 0$ 并转向步骤 2; 否则令 $out\_cost = out\_qty * (cost / qty)$ , $out\_qty = out\_qty$ , $qty = qty - out\_qty$ $cost = cost - out\_cost$ 并终止; 步骤 2 提示库存数量不足并终止

13 给定字母表  $\Sigma = \{0, 1\}$ 。串 S 由 1 开头，中间有零个或多个 0 或 1，最后由 1 结束，试用 BNF 描述串 S 的语法。

难易程度 A 较易

认知程度 简单应用

预计答题时间 0.5 分钟

参考答案  $S ::= 1 \{ 0 | 1 \}^* 1$

14 给定下述 BNF

句子	::=	主语 谓语
主语	::=	冠词 形容词 名词
冠词	::=	the   a   an
形容词	::=	black   gray
谓语	::=	动词 直接宾语
动词	::=	助动词 动词原形
助动词	::=	can   will
动词原形	::=	eat
直接宾语	::=	冠词 名词
名词	::=	wolf   goat

试判断下列句子是否符合以上描述的语法

- (1) the gray wolf will eat the wolf
- (2) an black goat will eat the wolf
- (3) an gray wolf can eat a black goat

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 2 分钟

参考答案 、 符合题中 BNF 所描述的语法，而 不符合。

15 给定下述 BNF 所描述的语法

$bexpr ::= bexpr \text{ or } bterm \mid bterm$

$bterm ::= bterm \text{ and } bfactor \mid bfactor$

$bfactor ::= \text{not } bfactor \mid ( bexpr ) \mid \text{true} \mid \text{false}$

试给出符合上述语法几个句子实例。

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 2 分钟

参考答案 下面给出几个实例

$\text{true and false or (not true)}$

$\text{false or true and not false}$

$\text{true and false and (not false and (true or false))}$

16 质数是指该数除了 1 之外没有其他因子。试给出一个算法，判断一个整数是否为质数。

难易程度 C 较难

认知程度 综合运用

预计答题时间 10 分钟

参考答案

17 完全数是指该数的所有因子之和等于它自身的数。例如 6 是完全数( $1+2+3$ )，28 也是完全数( $1+2+4+7+14$ )。试给出一个算法，判断一个整数是否为完全数。

难易程度 D 难度较大

认知程度 综合运用

预计答题时间 15 分钟

参考答案 判断一个数  $m$  是否是完全数的算法如下

步骤 1 令  $sum = 1$ ， $a=2$ （初始化）。

步骤 2 如果  $a$  能整除  $m$ ，则令  $sum$  等于  $sum$  加上  $a$ ，否则直接转步骤 3。

步骤 3 令  $a$  等于  $a$  加 1，如果  $a$  等于  $m$  转步骤 4，否则转步骤 2。

步骤 4 如果  $sum$  等于  $m$ ，输出  $m$  是完全数，否则输出  $m$  不是完全数。

## 第二章 基本数据类型

### 填空题、选择题

1 如果编译程序在语法分析阶段严格检查数据上的操作是否满足数据所属类型的要求,这种程序设计语言称为\_\_A\_\_, 否则称为\_\_B\_\_。Ada和Pascal等语言都是\_\_C\_\_, 而Smalltalk、BASIC等语言则是\_\_D\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 强类型的; B 弱类型的; C 强类型语言; D 弱类型语言

2 组成C++程序的最小单位是\_\_A\_\_,C++程序中有五种A \_\_B\_\_、\_\_C\_\_、\_\_D\_\_、\_\_E\_\_、\_\_F\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 单词; B 保留字; C 标识符; D 常量; E 运算符; F 分界符

3 每一种类型的数据都可分为\_\_A\_\_和\_\_B\_\_两类。类型决定了数据的\_\_C\_\_、\_\_D\_\_与\_\_E\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 变量; B 常量; C 表示方式; D 取值范围; E 可进行的操作

4 对于常量,其类型由\_\_A\_\_决定;对于变量,其类型由\_\_B\_\_决定。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 书写常量的语法形式; B 程序员显示声明

5 在C++语言中,有符号字符的取值范围可以从\_\_A\_\_到\_\_B\_\_,而有符号整数的取值范围可以从\_\_C\_\_到\_\_D\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A -128; B 127; C -32768; D 32767

6 对数据进行操作是通过运算完成的,\_\_A\_\_与\_\_B\_\_组成表达式。一个表达式有两个重要内容 \_\_C\_\_和\_\_D\_\_。其中C不仅取决于运算符的语义,而且还与运算符执行次序密切相关,而运算符的执行次序由运算符的\_\_E\_\_和\_\_F\_\_决定。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 运算符 ;B 操作数 ;C 表达式的求值结果 ;D 表达式结果的类型 ;E 优先级 ;F 结合性

7 变量的值主要通过赋值运算改变。赋值运算由赋值符号“=”将赋值表达式分为左右两边,左边必须是 A, 不能是 B、C 或其他表达式。能放在赋值号左边的语法单位又称为 D。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 变量 ;B 常量 ;C 符号常量 ;D 左值

8 当二元运算的两个操作数类型不一致时,C++编译程序会自动作 A, 注意在赋值运算的类型转换过程中可能 B。程序员也可显示地 C 类型转换。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 隐式类型转换 ;B 丢失数据 ;C 强制

9 下列单词是C++语言中的合法标识符, 请选择 \_\_\_\_\_

begin	array[i]	new	_1first	3rd
file_name	struct	OK?	_while	number3.5

难易程度 A 较易

认知程度 简单应用

预计答题时间 0.5 分钟

参考答案 、 、 、

10 下列哪些是 C++ 语言中的合法常量, 请选择 \_\_\_\_\_

10*2	100L	0238	'abc'	"A"
"ABC"	0237	0xABCD	0356L	0xAL

难易程度 B 中等难度

认知程度 理解

预计答题时间 2 分钟

参考答案 、 、 、 、 、 、

11 下列哪些是 C++ 语言中的合法常量, 请选择 \_\_\_\_\_

+.234E5	50.0U	3.14E0.2	-4.2E-10f	0.1206
E+10	35E	0.E-5	.E+5	0.0

难易程度 B 中等难度

认知程度 理解

预计答题时间 2 分钟

参考答案 、 、 、 、

12 请分别指出下列常量的类型

2            -2                            2u            2L            0xFFFF  
0x3FFF            0xFFFFFFFF            01234567

难易程度    B    中等难度

认知程度            简单应用

预计答题时间    5    分钟

参考答案

常量 2、-2 所属的类型是整数类型(int)。

常量 2u 所属的类型是无符号整数类型(unsigned int)。

常量 2L 所属的类型是长整数类型(long)。

常量 0xFFFF 0xFFFFFFFF 01234567 所属的类型是长整数类型(long)，因为它们都超过了整数类型(int)的取值范围，整数类型中能表示的最大整数是 0x7FFF，或八进制表示的 077777。

13 请分别指出下列常量的类型

2.F            .2L                            0.2E100            0.0

难易程度    B    中等难度

认知程度            简单应用

预计答题时间    3    分钟

参考答案

常量 2.F 所属的类型是浮点类型(float)。

常量 .2L、2E100、0.0 所属的类型是双精度类型(double)。

名词解释

14 标识符

难易程度    A    较易

认知程度            识记

预计答题时间    0.5    分钟

参考答案    在 C++ 程序中，标识符是由字母或下划线开头的字母、数字与下划线的序列。

15 常量

难易程度    A    较易

认知程度            识记

预计答题时间    0.5    分钟

参考答案    在程序执行过程中，其值不能改变的量称为常量。

16 变量

难易程度    A    较易

认知程度            识记

预计答题时间    0.5    分钟

参考答案    在程序执行过程中，其值可以改变的量称为变量。

17 表达式

难易程度    A    较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 表达式是由操作数和运算符按一定语法形式组成的符号序列。

问答题

18 试说明程序设计语言中类型的作用。

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案

19 试说明为什么在程序中要使用好的标识符？

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案

20 C++中有哪几种基本的数据类型？

难易程度 A 较易

认知程度 识记

预计答题时间 2 分钟

参考答案

21 请尽可能多地列出 C++中的表达式种类。

难易程度 A 较易

认知程度 识记

预计答题时间 2 分钟

参考答案

22 试说明 C++中隐式类型转换的规则。

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案

23 设 is\_ready、is\_busy 和 no\_paper 的值分别为 1、0 和 0，试计算下列表达式的值

!is\_ready && !is\_busy

is\_ready || is\_busy && no\_paper

(!is\_ready || is\_busy) && (is\_busy || no\_paper)

5 && !is\_ready && is\_busy || !no\_paper

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 8 分钟

参考答案



```

!is_ready && !is_busy = 0 && 1
= 0
is_ready || is_busy && no_paper = 1
= (0 && 0) = 1 || 0 = 1
(!is_ready || is_busy) && (is_busy || no_paper)
= (0 || 0) && (0 || 0) = 0 && 0 = 0
5 && !is_ready && is_busy || !no_paper
= 5 && 0 && 0 || 1 = 0 || 1 = 1

```

#### 24 求下列表达式的值

$x = 2.5$ 、 $a = 7$ 、 $y = 4.7$ ，求  $x + a \% 3 * (\text{int})(x + y) / 24$  的值。

$a = 2$ 、 $b = 3$ 、 $x = 3.5$ 、 $y = 2.5$ ，求  $(\text{float})(a + b) / 2 + (\text{int})x / (\text{int})y$  的值。

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 5 分钟

参考答案

(1) 设  $x = 2.5$ 、 $a = 7$ 、 $y = 4.7$ ，表达式

```

x + a % 3 * (int)(x + y) / 24
= 2.5 + (7 % 3) * (int)(2.5 + 4.7) / 24
= 2.5 + 1 * (int)(7.2) / 24 = 2.5 + 1 * 7 / 24 = 2.5 + 7 / 24
= 2.5 + 0 = 2.5

```

因此表达式  $x + a \% 3 * (\text{int})(x + y) / 24$  的值为 2.5。

(2)  $a = 2$ 、 $b = 3$ 、 $x = 3.5$ 、 $y = 2.5$ ，表达式

```

(float)(a + b) / 2 + (int)x / (int)y
= (float)(2 + 3) / 2 + (int)(3.5) / (int)(2.5)
= 5.0 / 2 + 3 / 2 = 2.5 + 1 = 3.5

```

因此表达式  $(\text{float})(a + b) / 2 + (\text{int})x / (\text{int})y$  的值为 3.5。

#### 25 试根据 C++ 语言中运算符的优先级和结合性质 给下列表达式加上括号而不改变其求值结果

$a = b + c * d < 2 \& 8$

$a \& 077 != 3$

$a == b \parallel a == c \&\& c < 5$

$c = x != 0$

$a < b == c == d$

$\text{file\_eof} \&\& \text{!input\_error} \&\& \text{MIN\_NUM} < \text{count} \&\& \text{count} < \text{MAX\_NUM} \parallel \text{!error}$

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 10 分钟

参考答案

```

a = ((b + (c * d)) < 2) & 8)
a & (077 != 3)
(a == b) || (a == c) && (c < 5)
c = (x != 0)

```

```
((a < b) == c) == d
(((file_eof && (!input_error)) && (MIN_NUM < count) && (count < MAX_NUM))
|| (!error))
```

26 试使用 C++ 中的表达式求解下列问题

- 用 C++ 中表达式表示半径为  $r$  的圆面积  $S$  ;
- 用 C++ 中表达式表示数  $a$  和  $b$  两者的大者 ;
- 用 C++ 中表达式表示数  $a$ 、 $b$  和  $c$  三者的最大者 ;
- 用 C++ 中的表达式表示自然数  $d$  是自然数  $a$  和  $b$  的公约数 ;
- 用 C++ 中的表达式表示  $ch$  是英文字母。
- 用 C++ 中的表达式表示将小写英文字母转换成大写字母。
- 用 C++ 中的表达式表示将大写英文字母转换成小写字母。

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 10 分钟

参考答案

27 试给出下列程序的输出结果

```
#include <iostream.h>
main()
{
    int a = 2000;
    float b = 15.5;
    cout << "a * b is: " << a * b << "\n";
    cout << "a * (int)b is: " << a * (int)b << "\n";
    cout << "(float)a * b is: " << (float)a * b << "\n";
}
```

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 3 分钟

参考答案

题目中程序的输出结果如下

```
a * b is: 31000
a * (int)b is: 30000
(float)a * b is: 31000
```

28 试给出下列程序的输出结果

```
#include <iostream.h>
main()
{
    const float pi = 3.14;
    char ch = 0x5A;
    int i;
```

```

cout << "(1) " << sizeof(pi) << "\n";
cout << "(2) " << sizeof(ch) << "\n";
cout << "(3) " << sizeof(ch && (pi || 0)) << "\n";
cout << "(4) " << sizeof(i = 4 + pi * 8) << "\n";
}

```

难易程度      B    中等难度

认知程度              简单应用

预计答题时间      5    分钟

参考答案

题目中程序的输出结果如下

- (1) 4
- (2) 1
- (3) 2
- (4) 2

29 试给出下列程序的输出结果

```

#include <iostream.h>
main()
{
    const float pi = 3.14;
    char ch = 0x5A;
    int i;

    cout << "(1) " << (ch && (pi || 0)) << "\n";
    cout << "(2) " << (i = 4 + pi * 8) << "\n";
    cout << "(3) " << (i = 4 + int(pi) * 8) << "\n";
    cout << "(4) " << (3 && !0 || 0 && (5 - 4)) << "\n";
}

```

难易程度      B    中等难度

认知程度              简单应用

预计答题时间      5    分钟

参考答案

题目中程序的输出结果如下

- (1) 1
- (2) 29
- (3) 28
- (4) 1

30 试给出下列程序的输出结果

```

#include <iostream.h>
main()
{
    const float pi = 3.14;
    char ch = 0x5A;

```

```

int i;

cout << hex;
cout << "(1) " << (ch + 2) << "\n";
cout << "(2) " << (ch & 0x45) << "\n";
cout << "(3) " << (ch | 0x47) << "\n";
cout << "(4) " << (ch ^ 0x45) << "\n";
cout << dec;
}

```

难易程度      B    中等难度

认知程度            简单应用

预计答题时间      5    分钟

参考答案

题目中程序的输出结果如下

(1) 5C

(2) 40

(3) 5F

(4) 1F

31 试将下列 C++ 程序片段划分为单词序列,并说明每一个单词的种类,对于其中的常量和变量还要说明其数据类型。

```

const float pi = 3.14f;
float d;
d = 20u * pi;

```

难易程度      C    较难

认知程度            简单应用

预计答题时间      10   分钟

参考答案

编程题

32 读入摄氏温度 C, 将它转换为华氏温度 F 并输出, 已知  $F = (9 * C) / 5 + 32$ 。

33 输入长方体的长、宽、高, 计算并输出长方体的体积和表面积。

34 为检查我们的观察能力, 以便找出以 7 为分母的分数在小数位后数字出现的规律, 请编写一个小程序计算并打印分数 1/7、2/7、3/7、4/7、5/7 和 6/7。

35 输入三个整数, 计算并输出它们的平均值以及三个数的乘积, 写出程序。

36 已知地球半径为 6371km, 计算并输出地球的表面积和体积。

37 输出一个三位整数, 将它反向输出, 例如输入 123, 输出应为 321。

38 输入两个复数的实部和虚部, 计算并输出两个复数的和、差、积和商。

---

### 第三章 基本控制结构

#### 填空、选择题

1 在设计程序的时候,我们常使用图形工具来表达程序的控制结构,早期这样的图形工具以\_\_A\_\_为主,后来为了适应结构化程序设计的要求,人们又开发出象\_\_B\_\_、\_\_C\_\_、\_\_D\_\_、\_\_E\_\_等新的图形工具。

难易程度 B 中等难度

认知程度 识记

预计答题时间 2 分钟

参考答案 A 程序流程图; B N-S 盒图; C PAD 图; D Warnier 图; E Jackson 图

2 C++语言中解决嵌套 if 语句的垂悬 else 问题使用 A 原则,即 B 。

难易程度 B 中等难度

认知程度 识记

预计答题时间 3 分钟

参考答案 A 最近匹配原则; B 规定 else 必须匹配最近那个没有匹配的 if

3 以下语句是合法的 C++语言语句吗?不合法的又错在哪里?

if (x > 0) then x = x + 1; else x = x - 1;

难易程度 A 较易

认知程度 理解

预计答题时间 2 分钟

参考答案 该语句中有 then,显然不符合 C++语言中的 if 语句的定义。

4 以下语句是合法的 C++语言语句吗?不合法的又错在哪里?

if (x > 0) x = x + 1 else x = x - 1;

难易程度 A 较易

认知程度 理解

预计答题时间 2 分钟

参考答案 x = x + 1 不是一个语句,只是一个表达式,需要在后面加上分号才构成语句。

5 以下语句是合法的 C++语言语句吗?不合法的又错在哪里?

if x > 0 x = x + 1; else x = x - 1;

难易程度 A 较易

认知程度 理解

预计答题时间 2 分钟

参考答案 x = x + 1 不是一个语句,只是一个表达式,需要在后面加上分号才构成语句。

6 以下语句是合法的 C++语言语句吗?不合法的又错在哪里?

if x > 0 x = x + 1; else x = x - 1;

难易程度 A 较易

认知程度 理解

预计答题时间 2 分钟

---

参考答案 应该在  $x > 0$  两边加上圆括号。

7 以下语句是合法的 C++ 语言语句吗？不合法的又错在哪里？

```
if (x > 0) if (x <= 10) x = x + 1; else x = x - 1;
```

难易程度 A 较易

认知程度 理解

预计答题时间 2 分钟

参考答案 整条语句是对的

8 以下语句是合法的 C++ 语言语句吗？不合法的又错在哪里？

```
if (x > 0) { if (x <= 10) x = x + 1 }; else x = x - 1;
```

难易程度 A 较易

认知程度 理解

预计答题时间 2 分钟

参考答案 要将花括号后面的分号移到花括号里面就得到了合法的 if 语句。

9 以下语句是合法的 C++ 语言语句吗？不合法的又错在哪里？

```
if (x > 0) { if (x <= 10) x = x + 1; else x = x - 1};
```

难易程度 A 较易

认知程度 理解

预计答题时间 2 分钟

参考答案 需要将花括号后的分号移到花括号里面。

10-12 假设下列语句中变量均已作适当声明与初始化，指出下列程序片段是否有语法错误？如有错误请改正它们。

```
10 float items;
    cin >> items;
    switch (items) {
        case 0.0: cout << "Radio"
        case 1.0: cout << "Television"
        case 1.5: cout << "Video Camera"
    }
```

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 该语句中 switch 后面的表达式是浮点型，显然不合语法，其次，每个 case 后面也是浮点常量，同样不合语法，再次每个输出语句应该以分号结束，否则就只是一个表达式，而不是语句，最后每个 case 后面的语句序列应该以 break 结束，虽然没有 break 并不违犯语法，但通常应该以 break 结束，因为往往有 break 才是程序编写者真正要表达的意思。

```
11 int x, y;
    int choice;
```

---

```

x = 3; y = 4;
cin >> choice;
switch (choice) {
    case 0, 1, 2: cout << "Small";
    case x: cout << "Middle";
    case y: cout << "Large";
};

```

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 显然由于 case 后面使用变量 x, y, 因此本语句不合法。值得注意的是第一个 case 后面接 0, 1, 2 是合法的, 因为它被理解为一个逗号表达式, 其最后的结果是 3, 因此它与 case 2 等价。同样本题的 case 语句存在缺少 break 的问题。

```

12    const int x = 3;
        const int y = 4;
        int choice;
        switch (choice) {
            case 1, 2, 3: cout << "Work hurry"; break;
            case x: cout << "busy"; break;
            case y: cout << "idle"; break;
            default: cout << "Unknow";
        }

```

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 本题与上题不同, 由于 x, y 是由 const 说明的常量, 因此它们也是常量表达式, 这样 case 后面接 x, y 都是合法的。但是正如上题所指出的, 由于 case 1, 2, 3 相当于 case 3, 而 x 的值又是 3, 因此这次发生了 case 后常量表达式的值相同的情况, 从而本题的 switch 语句也是不合法的。

13-16 假设以下程序片段中的变量已经正确声明, 试指出下列循环语句片段中的语法或语义错误, 所谓语义错误指循环体是否执行, 是否是死循环等。

```

13    for (k = 1, k <= n, k = k + 1);
        {
            f = f + (f * 1.5) % k;
            cout << f << "\n";
        }

```

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 for 后面括号中的逗号应该改成分号。另外在 for 后面括号后的分号表明该 for 语句的循环体是一条空语句, 而编程者心目中的循环体 (由花括号括起来的块语句) 实

---

际上不会循环执行。

```
14      p = 100;  f = 5;
        do {
            p = p - 3;
            f = f + p * p
        } while (p != 0)
        cout << f;
```

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 while(p!=0)后面少了一个分号，另外循环体中右花括号前的语句 f=f+p\*p 也少了一个分号。由于 p 初始化为 100，而每次循环都减 3，所以它是不可能等于 0 的，因此循环终止条件永远也不会得到满足。

```
15      sum = 1;
        count = 10;
        for (int i = 1; i <= count; i++) {
            sum = sum + sum * i;
            count = sum + count * i;
        }
        count << "Sum: " << sum << ", Count: " << count;
```

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 在循环体中对 count 的改变使得循环终止条件 i <= count 不可能得到满足，从而造成了死循环。

```
16      total_score = 0;
        count = 0;
        cin >> score;
        while (score > 0) {
            total_score = total_socre + score;
            count = count + 1;
        };
        cout << "Total score: " << total_score << "Average score: ";
        cout << (double(total_socre) / count);
```

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 应该在 count = count + 1;后面加上语句 cin >> score; 以录入新的 score 值。

17-22 整数类型变量 x 和 y 分别被初始化为 3 和 100，下列语句的循环共执行几次？执行完



---

下列语句后的 x 和 y 的值分别是多少？

17 while (x <= y) x = 2 \* x;

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 循环的总次数为 6 ,而循环结束后x为  $3 * 2^6=192$  ,y的值在循环体中没有改变 , 仍然为 100。

18 do x = 2 \* x; while (x < y);

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 循环执行次数为 6 次，而循环执行完毕后 x=192，y=100。

19 while (y / x > 5) if (y - x > 25) x = x + 1; else y = y / x;

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 循环总次数应该为 14，x = 14，y = 100。

20 do { x = y / x; y = y - x; } while (x >= y);

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 共执行 1 次循环，循环执行完毕后 x = 33，y = 67。

21 do { y = y / x - 1; if (y >= x) y = x; } while (y >= 1);

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 循环执行 2 次，执行完毕后 x=3, y = 0。

22 为下列程序片段填空，使程序片段(1)与(2)的结果相同。

(1) if \_\_\_\_\_ min = a;  
    else if \_\_\_\_\_ min = b;  
    else min = c;

(2) min = a;  
    if \_\_\_\_\_ min = b;  
    if \_\_\_\_\_ min = c;

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

### 参考答案

```
(1) if (a < b && a < c) min = a;
    else if (b < a && b < c) min = b;
    else min = c;
(2) min = a;
    if (min > b) min = b;
    if (min > c) min = c;
```

### 程序输出结果题

23 下面程序根据录入的年、月、日求该天是星期几，其中的公式来自蔡勒(Zeller)。假如程序输入的三个数分别为 1971、2、26，即其中的 year, month, day 分别是 1971、2 和 26，试给出程序的输出结果。

```
#include <iostream.h>
int main()
{
    int year, month, day, centry;
    int temp_year, temp_month;
    int week_day;

    cout << "\nPlease input the year, month and day: ";
    cin >> year >> month >> day;

    temp_year = year; temp_month = month;
    if (month == 1 || month == 2) {
        temp_year = temp_year - 1;
        temp_month = temp_month + 12;
    }

    centry = temp_year / 100; temp_year = temp_year % 100;
    if (centry == 0) centry = 19; // The default centry is 19

    week_day = centry / 4 - 2 * centry;
    week_day = week_day + temp_year + temp_year / 4;
    week_day = week_day + 26 * (temp_month + 1) / 10 + day - 1;
    week_day = week_day % 7;

    cout << "The week day is " << week_day << " of ";
    cout << year << "/" << month << "/" << day << "\n";

    getch();
    return 1;
}
```

难易程度 A 较易

---

认知程度          简单应用

预计答题时间    4    分钟

参考答案

Please input the year, month and day: 1971 2 26

The week day is 5 of 1971/2/26

#### 24 给出下面程序的输出结果

```
#include <iostream.h>
int main()
{
    int last_fib, curr_fib, next_fib;
    int index;
    const int max_number = 5;

    last_fib = 0; curr_fib = 1;
    cout << "\nFibonacci(0): " << last_fib;
    for (index = 0; index < max_number; index = index + 1) {
        cout << "\nFibonacci(" << (index + 1) << "): " << curr_fib;
        next_fib = curr_fib + last_fib;
        last_fib = curr_fib;
        curr_fib = next_fib;
    }
    return 0;
}
```

难易程度    A    较易

认知程度          简单应用

预计答题时间    4    分钟

参考答案

Fibonacci(0): 0

Fibonacci(1): 1

Fibonacci(2): 1

Fibonacci(3): 2

Fibonacci(4): 3

Fibonacci(5): 5

#### 25 指出下面程序的输出结果

```
#include <iostream.h>
int main()
{
    unsigned long factor;
    unsigned int number;
    const unsigned int max_number = 6;

    factor = 1;
```

```

    for (number = 2; number <= max_number; number = number + 1) {
        factor = factor * number;
        if (number % 2 == 0) cout << "\n" << number << "!" << " = " << factor;
    }
    return 1;
}

```

难易程度     B   中等难度

认知程度         简单应用

预计答题时间     5    分钟

参考答案

2! = 2

4! = 24

6! = 720

26 给出下面程序的输出结果

```

#include <iomanip.h>
int main()
{
    const int start_num = 1, end_num = 5;
    const int width = 2;

    int start_pos, row_num, col_num;
    int index;

    cout << "\n";
    start_pos = (end_num - start_num) * width;
    for (col_num = start_num; col_num <= end_num; col_num = col_num + 1) {
        for (index = 0; index < start_pos; index = index + 1) cout << ' ';
        for (row_num = start_num; row_num < col_num; row_num = row_num + 1) {
            cout << setw(width) << row_num;
        }
        for (row_num = col_num; row_num >= start_num; row_num = row_num - 1) {
            cout << setw(width) << row_num;
        }
        start_pos = start_pos - width;
        cout << "\n";
    }
    return 1;
}

```

难易程度     C   难度较大

认知程度         简单应用

预计答题时间     10   分钟

参考答案

1

```

    1 2 1
  1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1

```

## 27 给出下面程序的输出结果

```

#include <iostream.h>
#include <iomanip.h>
int main()
{
    const int start_num = 0, end_num = 6;
    const int width = 4;

    int start_pos, row_num, col_num;
    int index, combine;

    cout << "\n";
    start_pos = (end_num / 2 - start_num) * width;
    for (col_num = start_num; col_num <= end_num; col_num = col_num + 1) {
        for (index = 0; index < start_pos; index = index + 1) cout << ' ';
        combine = 1; row_num = start_num;
        while (row_num <= col_num) {
            cout << setw(width) << combine;
            row_num = row_num + 1;
            combine = ((col_num - row_num + 1) * combine) / row_num;
        }
        start_pos = start_pos - width / 2;
        cout << "\n";
    }
    return 1;
}

```

难易程度    C    难度较大  
 认知程度    综合应用  
 预计答题时间    12 分钟  
 参考答案

```

      1
    1 1
  1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

```

## 28 给出下面程序的输出结果

```
#include <iostream.h>
int main()
{
    int number;
    const int max_number = 9;

    int factor;
    int left_number;
    int is_first_factor;

    for (number = 2; number <= max_number; number = number + 1) {
        cout << "Number: " << number << " = ";

        is_first_factor = 1;
        left_number = number;
        while (left_number > 1) {
            factor = 2;
            while (left_number % factor != 0) factor++;
            if (is_first_factor == 1) cout << factor;
            else cout << " * " << factor;

            left_number = left_number / factor;
            is_first_factor = 0;
        }
        cout << '\n';
    }
    return 0;
}
```

难易程度 C 难度较大

认知程度 综合应用

预计答题时间 12 分钟

参考答案

Number: 2 = 2

Number: 3 = 3

Number: 4 = 2 \* 2

Number: 5 = 5

Number: 6 = 2 \* 3

Number: 7 = 7

Number: 8 = 2 \* 2 \* 2

Number: 9 = 3 \* 3

程序填空题

29 下面程序，判断用户输入的年份是否闰年。其中闰年年份是要么能被 4 整除且不被 100 整除，要么能被 400 整除，此外都不是闰年。

```
// File: isleap.cpp
// Function: The user input a year number, check whether it is a leap year.

#include <iostream.h>

int main()
{
    int year;

    cout << "\nPlease input a year: ";
    cin >> year;

    if ( _____ ) {
        cout << "The year " << year << " is a leap year!";
    } else {
        if ( _____ ) cout << "The year " << year << " is a leap year!";
        else cout << "The year " << year << " is NOT a leap year!";
    }

    return 0;
}
```

难易程度     B   中等难度

认知程度         理解

预计答题时间     5   分钟

参考答案

```
(year % 4 == 0) && (year % 100 != 0)
year % 400 == 0
```

30 下面程序求一个自然数中含有多少个 2 的因子。

```
// File: fact2num.cpp
// Function: Calculate the factor 2 number of a nature number.

#include <iostream.h>

int main()
{
    int number, temp_number;
    int factor_count;
    const int base = 2;
```

```

cout << "\nPlease input a nature number(such as 140): ";
cin >> number;
if (number <= 0) {
    cout << "Your input " << number << " is error!!";
    return 1;
}

factor_count = 0;
temp_number = _____;
while (temp_number % base == 0) {
    factor_count = factor_count + 1;
    _____
}

cout << "The factor(" << base << ") number of " << number << " is: ";
cout << factor_count;
return 0;
}

```

难易程度     A    较易

认知程度            理解

预计答题时间     5    分钟

参考答案

```

number
temp_number = temp_number / base;

```

31 下面程序解决爱因斯坦台阶问题：有人走一台阶，若以每步走两级则最后剩一级；若每步走三级则剩两级；若每步走四级则剩三级；若每步走五级则剩四级；若每步走六级则剩五级；若每步走七级则刚好不剩。问台阶共有几级？

```

// File: enststep.cpp
// Function: Solve the Einsten step question.

#include <iostream.h>

int main()
{
    int step;

    _____

    while (step % 3 != 2 || step % 4 != 3 || step % 5 != 4 || step % 6 != 5) {
        _____
    }
}

```



```
cout << "\nThe step is: " << step << "\n";
return 1;
}
```

难易程度    B    中等难度  
认知程度        理解  
预计答题时间    10   分钟  
参考答案

```
step = 7;
step += 14;
```

32    下面程序求两个正整数的最小公倍数。

```
// File: commmult.cpp
// Function: Calculate the least common multiple of two nature number.

#include <iostream.h>

int main()
{
    int num1, num2;
    unsigned long mult;

    cout << "\nPlease input two number(such as 56 78): ";
    cin >> num1 >> num2;

    _____

    while (_____) mult = mult + num1;

    cout << "The least common multiple of [" << num1 << ", ";
    cout << num2 << "] is: " << mult;
    return 1;
}
```

难易程度    B    中等难度  
认知程度        理解  
预计答题时间    10   分钟  
参考答案

```
mult = num1;
mult % num2 != 0
```

33    下面程序求不超过正整数 n 的 2 的最大幂值。

```
// File: max2pow.cpp
// Function: Calculate the max power of 2 which just less than the input number.

#include <iostream.h>

int main()
{
    int max_power, max_number;
    const int base = 2;
    int times;

    cout << "\n\nPlease input a number(such as 140): ";
    cin >> max_number;
    if (max_number <= 0) {
        cout << "\nYour input is error!!";
        return 1;
    }

    times = 0;

    while ((max_power * base) <= max_number) {
        times = times + 1;
    }

    cout << "The max power of 2 is " << max_power << "(= 2 ** " << times;
    cout << ") Which just less than " << max_number;
    return 0;
}
```

难易程度     B   中等难度

认知程度         理解

预计答题时间     10   分钟

参考答案

```
max_power = 1;
max_power = base * max_power;
```

34 下面程序统计《面向对象程序设计》考试成绩：输入全班学生的考试成绩，计算平均成绩并找出其中最高分与最低分。（提示：批量数据输入通常不事先规定输入次数，而是以一个特殊标志作为输入结束。程序根据结束标志统计人数。）

```
// File: score.cpp

#include <iostream.h>
```

```

int main()
{
    float score, max_score, min_score, total_score;
    int count;

    cout << "\nPlease input the score of student, ";
    cout << "\nTo Input a negative number will end the input: ";

    count = 0;
    cout << "\nNo. " << count+1 << ": ";
    cin >> score;

    max_score =          ;
    min_score =          ;
    total_score = 0;
    while (score >= 0 && score <= 100) {
        if (score < min_score) _____;
        if (score > max_score) _____;
        total_score = total_score + score;

        _____

        cout << "No. " << count+1 << ": ";
        cin >> score;
    }

    if (count > 0) {
        cout << "\nThe number of student is: " << count;
        cout << "\nThe max score is: " << max_score;
        cout << "\nThe min score is: " << min_score;
        cout << "\nThe average score is: " << (total_score / count);
    } else cout << "\nYou have not input valid score.";

    return 0;
}

```

难易程度    B    中等难度  
 认知程度        理解  
 预计答题时间    10    分钟  
 参考答案

```

score
score
min_score = score
max_score = score
count = count + 1;

```

35 下面程序水仙花数 (Armstrong number) 是一个 $n$  ( $n \geq 3$ ) 位数字的数, 它等于每个数字的 $n$ 次幂的和。例如 153 是水仙花数 (具有 3 位数字且等于  $1^3 + 5^3 + 3^3$ )。下面程序求小于 999 的另外三个水仙花数。

```
// File: armstron.cpp
// Function: Calculate the armstrong number.

#include <iostream.h>

int main()
{
    const int n = 3;
    const int base = 10;

    unsigned int start_num, end_num;
    unsigned int num, sum, power;
    int i, j, bit;
    unsigned int bit_power;

    power = 1;
    for (i = 1; i <= n; i++) power = power * base;
    start_num = _____;
    end_num = _____;

    for (num = start_num; num <= end_num; num++) {
        sum = 0; power = 10;
        for (i = 1; i <= n; i++) {
            bit = _____;

            bit_power = 1;
            for (j = 1; j <= n; j++) bit_power = bit_power * bit;
            sum = _____;

            power = _____ ;
        }

        if (sum == num) {
            cout << "\nThe number " << num << " is an armstrong number!";
        }
    }

    return 1;
}
```

难易程度 D 难度较大  
认知程度 理解  
预计答题时间 20 分钟  
参考答案

```
power
power / base - 1
(num % power) / (power / base)
sum + bit_power
power * base
```

36 下面程序使用下面公式计算  $\pi$

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + \frac{1}{4n-3} - \frac{1}{4n-1}, \text{ 下面程序计算到 } n=10000$$

试根据以上公式填写下面程序中的空，以完成正确的计算。

```
#include <iostream.h>
int main()
{
    double pi;
    unsigned int number;
    int sign;          // 用来记录当前正计算项的符号
    const unsigned int max_number = 10000;

    pi = 0; sign = 1;
    for (number = 1; number <= max_number; _____) {
        pi = pi + _____;
        sign = _____;
    }
    pi = _____;

    cout.precision(12);
    cout << "\nPi = " << pi << ", when max number = " << max_number;
    return 0;
}
```

难易程度 A 较易  
认知程度 理解  
预计答题时间 5 分钟  
参考答案

```
number = number + 2
sign * (1.0 / number)
-sign
pi * 4
```

37 下面程序求解《孙子算经》中的第 26 题 今有物不知数，三三数之，剩二；五五数之，

剩三；七七数之，剩二；问物几何？程序中找到从 1 到 500 满足该条件的自然数，试填写程序中的空，使其完成正确的功能。

```
#include <iostream.h>
int main()
{
    unsigned int number;
    const unsigned int max_number = 500;

    // 下面循环中保证 number 除 7 余 2，
    number = 9;
    while (_____) {
        while ((number % 3 != 2 || number % 5 != 3) && (number < max_number)) {
            _____
        }
        if (number < max_number) {
            cout << "\nFind " << number << " % 3 == 2 && ";
            cout << number << " % 5 == 3 && ";
            cout << number << " % 7 == 2";
            number = number + 7;
        }
    }
    return 0;
}
```

难易程度     A   较易

认知程度         理解

预计答题时间     5   分钟

参考答案

```
number < max_number
number = number + 7;
```

38 下面程序输出 1-999 中能被 3 整除，且至少有一位数字是 5 的所有整数，试填写程序中空以完成正确的功能。

```
#include <iostream.h>
int main()
{
    unsigned int number, temp_number;
    unsigned int bit_number;
    // 引进各种常量，以便程序具有良好的扩充性。
    const int base = 10, factor = 3, need_bit = 5;
    const unsigned int max_number = 1000;

    // 下面循环中保证 number 能够被 factor(即 3)整除
    for (number = factor; number < max_number; _____) {
```

```

temp_number = number;
// 下面循环求 number 的每一位
while (temp_number > 0) {
    bit_number = _____ ;
    if ( _____ ) {
        cout << "\nFind " << number << " % " << factor << " == 0";
        cout << ", and " << need_bit << " in it!";
        break;
    }
    _____
}
}
return 0;
}

```

难易程度     B   中等难度

认知程度         综合应用

预计答题时间     8   分钟

参考答案

```

number = number + factor
temp_number % base
bit_number == need_bit
temp_number = temp_number / 10;

```

39 下面程序求 2-1000 中的守形数（若某数的平方，其低位与该数本身相同，则称该数为守形数。例如 25， $25^2=625$ ，625 的低位与 25 相同，则称 25 为守形数），试填写程序中的空。

```

#include <iostream.h>
int main()
{
    unsigned int number, temp_number;
    unsigned int bit_number;
    unsigned long num_square, temp_square;
    int index;
    // 引进 times 是为了将程序扩充到计算立方守形数、四次方守形数等。
    const int base = 10, times = 2;
    const unsigned int max_number = 1000;

    for (number = 2; number < max_number; number = number + 1) {
        num_square = 1;
        // 计算 number 的 times 次方到 num_square。
        for (index = 0; index < times; index = index + 1) num_square = _____;
        // 下面检查 num_square 的低位是否与 number 相同。
        temp_square = num_square;
        temp_number = number;
    }
}

```

```

while (temp_number > 0) {
    bit_number = _____;
    if (temp_number % base != bit_number) break;    // 发生了不同，不再检查。
    temp_number = _____;
    temp_square = _____;
}
if (temp_number == 0) {
    // temp_number == 0 说明 number_square 和 number 的低位都相同，即 number 满足要求。
    cout << "\nFind " << number << ", and " << number;
    for (index = 0; index < times - 1; index = index + 1) {
        cout << " * " << number;
    }
    cout << " = " << num_square;
}
}
return 0;
}

```

难易程度     B   中等难度

认知程度         综合应用

预计答题时间     8   分钟

参考答案

```

num_square * number
temp_square % base
temp_number / base
temp_square / base

```

40 下面程序利用以下公式计算  $\cos(x)$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

其中程序计算到最后一项的绝对值小于  $10^{-6}$  时为止，试填写其中的空。

```

#include <iostream.h>
int main()
{
    double cos_value, x, x_square, last_value;
    unsigned int number;
    unsigned long factor;
    int sign;
    const double epsilon = 10E-6;

    // 让用户录入 x
    cout << "\nPlease input x for calculate cos(x): ";
    cin >> x;

    // 下面的 cos_value 用来保存每次计算 cos(x) 的结果，x_square 用来保存 x 的各次方

```



```

// sign 用来记录当前项的符号，factor 用来保存 number 的阶乘
cos_value = 0; x_square = 1.0;
sign = 1; number = 1; factor = 1;
do {
    // last_value 实际上记录当前一项的值
    last_value = _____;
    cos_value = cos_value + sign * last_value;

    x_square = _____;
    factor = _____;
    number = number + 2;
    sign = _____;
} while (last_value > epsilon);

cout << "Cos(" << x << ") = " << cos_value << "\n";
return 1;
}

```

难易程度     B   中等难度

认知程度         综合应用

预计答题时间     8     分钟

参考答案

```

x_square / factor
x_square * x * x
factor * number * (number + 1)
temp_square / base
-sign

```

41 下面程序利用以下公式迭代地求 x 的立方根

$$y_{n+1} = \frac{2}{3}y_n + \frac{x}{3y_n^2}$$

初始值 $y_0=x$ （用户录入x），误差要求 $\epsilon=10^{-4}$ 。

```

#include <iostream.h>
#include <math.h>
int main()
{
    double curr_value, x, next_value;
    const double epsilon = 10E-4;

    cout << "\nPlease input x for calculate subtriplicate(x): ";
    cin >> x;

    last_value = x;
    do {
        curr_value = _____;
    }
}

```

```

        next_value = _____;
    } while (fabs(next_value - curr_value) > epsilon);

    cout << "Subtriplicate(" << x << ") = " << curr_value << "\n";
    return 1;
}

```

难易程度     A    较易

认知程度         简单应用

预计答题时间     3    分钟

参考答案

```

next_value
2.0 * curr_value / 3 + x / (3 * curr_value * curr_value)

```

42 下面程序利用以下公式计算  $\ln(1/(1-x))$

$$\ln(1/(1-x)) = x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \dots$$

程序计算到最后一项的绝对值小于  $10^{-5}$  时为止。

```

#include <iostream.h>
int main()
{
    double result, x, x_square, last_value;
    unsigned int number;
    const double epsilon = 10E-5;

    cout << "\nPlease input x for calculate ln(1/(1-x)): ";
    cin >> x;

    result = 0; x_square = x; number = 1;
    do {
        last_value = _____;
        result = result + last_value;
        x_square = _____;
        number = number + 1;
    } while (last_value > epsilon);

    cout << "ln(1/(1-" << x << ")) = ln(" << (1/(1-x)) << ") = " << result << "\n";
    getch();
    return 1;
}

```

难易程度     A    较易

认知程度         简单应用

预计答题时间     3    分钟

参考答案

```

x_square / number

```

x \* x\_square

43 以下程序用来求 1-100 之间的所有素数，试填写其中的空。

```
#include <iostream.h>
#include <iomanip.h>
int main()
{
    unsigned int number, factor;
    const unsigned int max_number = 10000;
    // 下面的常量 width, screen_width 用来控制输出格式，即每一行输出多少个素数。
    const int width = 6, screen_width = 80;
    // 下面的变量 count 用来记录已经得到的素数个数，line_max_num 用来记录屏幕一行最多
    // 输出的素数个数。
    int is_prime, count, line_max_num;

    line_max_num = screen_width / width;
    cout << "\n" << setw(width) << 2;    // 起初令 2 是素数，下面循环从 3 开始检查。
    count = 1;
    for (number = 3; number < max_number; number = number + 2) {
        factor = 2; is_prime = 1;
        // 只检查到 factor 小于 number 的平方根即可，因为数 number 的大于 number 的平方根
        // 的因子必对应与一个小于 number 平方根的因子。
        while (factor * factor <= number) {
            if (_____) { is_prime = 0; break; }
            factor = _____;
        }
        if (is_prime) {
            count = count + 1;
            if (count % line_max_num == 0) cout << "\n";
            cout << setw(width) << number;
        }
    }

    cout << "\nTotal prime number = " << count;
    return 0;
}
```

难易程度     A   较易

认知程度         简单应用

预计答题时间     3   分钟

参考答案

```
number % factor == 0
factor + 1
```

44 下面程序用来求 1-10000 之间的完全数(所谓完全数，就是其因子之后等于该数本身，

例如 6 就是完全数，因为其因子包括 1、2、3 三个数)。

```
#include <iostream.h>
#include <iomanip.h>
int main()
{
    unsigned int number, factor, factor_sum, factor_square;
    const unsigned int max_number = 10000;

    cout << "\n";
    for (number = 2; number < max_number; number = number + 1) {
        factor_sum = 1; factor = 2;
        // 因为如果 factor 是 number 的因子的话，那么 number / factor 也是 number 的因子
        // 因此下面的程序中一次加两个因子到和 factor_sum 上。但注意在 number 恰好是
        // factor 的平方时，factor 只算 number 的一个因子。
        factor_square = factor * factor;
        while (_____) {
            if (number % factor == 0) {
                if (factor_square != number) {
                    factor_sum = _____;
                } else factor_sum = _____;
            }
            factor = factor + 1; factor_square = factor * factor;
        }

        if (_____) {
            cout << "\nFind " << number << " is a complete number!";
        }
    }
    return 0;
}
```

难易程度 B 中等难度

认知程度 综合应用

预计答题时间 8 分钟

参考答案

```
factor_square <= number
factor_sum + factor + number / factor
factor_sum + factor
factor_sum == number
```

45 要将一张 100 元的大钞票，换成等值的 10 元、5 元、2 元和 1 元一张的小钞票。要求每次换成 40 张小钞票，每种至少一张。下面程序输出所有可能的换法，试填写其中的空。

```
#include <iostream.h>
#include <iomanip.h>
```

```

int main()
{
    // 下面变量 number_10, number_5, number_2, number_1 分别用来记一种换法中
    // 10 元、5 元、2 元和 1 元钞票的数目。
    unsigned int number_10, number_5, number_2, number_1;
    unsigned int max_10, max_5, max_2, max_1;
    // 引进常量 total_cost 来记待换的总钱数, total_number 记要求换的小钞票总数,
    // 以便程序容易扩充。
    const unsigned int total_cost = 100, total_number = 40;
    int count, curr_cost;

    // 先计算每种钞票可能的最多张数, 以减少下面循环的循环次数, 计算的依据都是因为题目
    // 要求每种钞票要求至少一张
    max_10 = (total_cost - (5 + 2 + 1)) / 10;
    if (max_10 > (total_number - 3)) max_10 = total_number - 3;
    max_5 = _____;
    if (max_5 > (total_number - 3)) max_5 = total_number - 3;
    max_2 = (total_cost - (10 + 5 + 1)) / 2;
    if (max_2 > (total_number - 3)) max_2 = total_number - 3;

    count = 0;
    for (number_10 = 1; number_10 <= max_10; number_10++) {
        for (number_5 = 1; number_5 <= max_5; number_5++) {
            for (number_2 = 1; number_2 <= max_2; number_2++) {
                number_1 = _____;
                curr_cost = _____;
                if (curr_cost == total_cost) {
                    cout << "\nFind a change method: <10: " << number_10;
                    cout << ", 5: " << number_5 << ", 2: " << number_2;
                    cout << ", 1: " << number_1 << ">";
                    _____
                }
            }
        }
    }

    cout << "\nTotal change method number is: " << count;
    return 0;
}

```

难易程度     B   中等难度

认知程度     综合应用

预计答题时间     8   分钟

参考答案

(total\_cost - (10 + 2 + 1)) / 5

```

total_number = (number_10 + number_5 + number_2)
number_10 * 10 + number_5 * 5 + number_2 * 2 + number_1
count = count + 1;

```

46 下面计算以下公式的和 S

$$S = \frac{1}{1 \times 2} + \frac{1}{3 \times 4} + \frac{1}{5 \times 6} + \dots + \frac{1}{99 \times 100}$$

```

#include <iostream.h>
int main()
{
    double result;
    unsigned int number;
    const unsigned int max_number = 100;

    result = 0;
    for (number = 2; number <= max_number; _____) {
        result = result + _____;
    }

    cout << "\nThe result is: " << result;
    return 1;
}

```

难易程度 A 较易

认知程度 简单应用

预计答题时间 3 分钟

参考答案

```

number = number + 2
1.0 / (number * (number-1))

```

47 下面程序根据以下公式计算 $\pi$ 的值

$$\frac{\pi}{6} = \arctg \frac{1}{\sqrt{3}}, \text{ 而 } \arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$$

下面程序计算到最后一项的绝对值小于  $10^{-5}$  时为止。

```

#include <iostream.h>
#include <math.h>
int main()
{
    double pi, x, x_square, last_value;
    unsigned int number;
    int sign;
    const double epsilon = 10E-5;

    pi = 0.0; sign = 1; number = 1;

```

```
x = sqrt(3.0) / 3.0; x_square = x;
do {
    last_value = _____;
    pi = pi + sign * last_value;

    x_square = _____;
    number = number + 2;
    sign = _____;
} while (last_value > epsilon);
pi = pi * 6;

cout.precision(12);
cout << "\nPi = " << pi;
return 0;
}
```

难易程度    A    较易  
认知程度        简单应用  
预计答题时间    5    分钟  
参考答案

x\_square / number  
x\_square \* x \* x  
-sign

48 下面程序利用以下公式计算 e 的值

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots$$

其中程序计算到最后一项的绝对值小于  $10^{-6}$  时为止，试填写其中的空。

```
#include <iostream.h>
#include <math.h>
int main()
{
    double e_value, last_value;
    unsigned int number;
    unsigned long factor;
    const double epsilon = 10E-6;

    e_value = 1; number = 1; factor = 1;
    do {
        last_value = _____;
        e_value = e_value + last_value;
        number = number + 1;
        factor = _____;
    } while (last_value > epsilon);
}
```

```

    cout.precision(12);
    cout << "\ne = " << e_value;
    return 0;
}

```

难易程度     A    较易

认知程度            简单应用

预计答题时间     5    分钟

参考答案

```

1.0 / factor
factor * number

```

49 下面程序用来求解十分有名的百鸡问题(选自《张邱建算经》) 今有鸡翁一,值钱五;鸡母一,值钱三;鸡雏三,值钱一。凡百鸡买百只。问鸡翁、母、雏各几何?下面程序给出所有解答,试填写其中的空。

```

#include <iostream.h>
#include <iomanip.h>
int main()
{
    unsigned int number_cock, number_hen, number_chicken;
    unsigned int max_cock, max_hen;
    const unsigned int total_number = 100;
    const double total_cost = 100;
    double curr_cost;
    int count;

    // 计算可能最多的公鸡、母鸡数。
    max_cock = total_cost / 5;
    max_hen = total_cost / 3;
    count = 0;
    for (number_cock = 1; _____; number_cock++) {
        for (number_hen = 1; _____; number_hen++) {
            number_chicken = _____ ;
            curr_cost = number_cock * 5.0 + number_hen * 3.0 + number_chicken / 3.0;
            if ( _____ ) {
                cout << "\nFind a buy method: <cock: " << number_cock;
                cout << ", hen: " << number_hen << ", chicken: " << number_chicken << ">";
                count = count + 1;
            }
        }
    }

    cout << "\nTotal buy method number is: " << count;
    getch();
    return 0;
}

```



```
}
```

难易程度     A    较易

认知程度         简单应用

预计答题时间     5    分钟

参考答案

```
number_cock <= max_cock
number_hen <= max_hen
total_number = (number_cock + number_hen)
curr_cost == total_cost
```

50 下面程序用来求四位的喀普利卡数,所谓喀普利卡数是指该数的前两位与后两位相加后再求平方,其结果正好是该数本身。试填写其中的空。

```
#include <iostream.h>
int main()
{
    const int bit_num = 4, base = 10;
    unsigned long start_num, end_num;
    unsigned long num, first_part, second_part;
    unsigned long square, middle_power;

    // 下面求 bit_num 位数的最小数和最大数,并求 10 的 bit_num/2 次方,以对 bit_num 位数
    // 进行剖分为两部分,每部分的位数为 bit_num 位。
    end_num = 1; middle_power = 1;
    for (int i = 0; i < bit_num; i++) {
        end_num = end_num * base;
        if (_____) middle_power = middle_power * base;
    }
    start_num = _____;
    end_num = end_num - 1;

    for (num = start_num; num <= end_num; num = num + 1) {
        first_part = _____;
        second_part = num % middle_power;
        square = (first_part + second_part) * (first_part + second_part);
        if (_____) {
            cout << "\nThe num <" << num << "> is ";
            cout << bit_num << " bit Kaprekar number!";
        }
    }
    return 1;
}
```

难易程度     B    中等难度

认知程度         简单应用

预计答题时间     8    分钟

---

### 参考答案

```
i < bit_num / 2
end_num / base
num / middle_power
num == square
```

### 编程题目

51 一个人离家  $1/3$  公里，以每小时 10 公里的速度骑车回家；一只苍蝇以每小时 30 公里的速度在骑车人与家门之间不停地来回飞。如果苍蝇正好离开自行车向家门飞去，到达门后再飞回自行车，这样来回地飞，直到被自行车撞在门上不幸压死，问苍蝇临死前飞行总距离是多少？（提示：寻找一个精美算法。）

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 8 分钟

### 参考答案

```
// File: commult.cpp
#include <iostream.h>

int main()
{
    int fly_distance;

    cout << "\nThe total distance is : " << 30.0 * (1.0/3.0) / 10.0;
    return 1;
}
```

52 广州大桥按不同型号征收车辆过桥费：自行车免费，摩托车 2 元，小汽车 5 元，大客车与货车 8 元，货柜车 12 元。编写一个程序，按不同型号车辆计算通过广州大桥应征的过桥费。（提示：可以用整数对不同型号的车辆进行编码。）

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 5 分钟

### 参考答案

```
// File: brdgefee.cpp
// Function: Output the passing bridge fee of the vechicle according to the choice of user.

#include <iostream.h>

int main()
{
    int choice;

    cout << "\nPlease select the kind of your vechicle:";
```

```
cout << "\n1. Bicycle.";
cout << "\n2. Motorcycle.";
cout << "\n3. Car.";
cout << "\n4. Bus.";
cout << "\n5. Truck.";
cout << "\nEnter your choice(1-5): ";
cin >> choice;

switch (choice) {
    case 1: cout << "The bicycle is free of charge."; break;
    case 2: cout << "The fee of motorcycle is 2 yuan."; break;
    case 3: cout << "The fee of car is 5 yuan."; break;
    case 4: cout << "The fee of bus is 8 yuan."; break;
    case 5: cout << "The fee of truck is 12 yuan."; break;
    default: cout << "Your choice is error!!"; break;
}

return 0;
}
```

53 有关专家十分关注珠江渔业资源的问题。目前珠江中大约有八千万条鱼，平均每年以3.5%的速度减少。请编写一个程序，计算在多少年之后鱼的数目下降到目前的一半？多少年后下降到目前的十分之一？

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 8 分钟

参考答案

```
// File:fish.cpp

#include <iostream.h>

int main()
{
    const float current_fish = 8E7;
    const float decrease_ratio = 3.5E-2;
    float feature_fish, half_fish;
    int half_years, one_tenth_years;

    feature_fish = current_fish;
    one_tenth_years = 0;
    while (feature_fish > current_fish / 2) {
        feature_fish = feature_fish * (1 - decrease_ratio);
        one_tenth_years = one_tenth_years + 1;
    }
}
```

---

```
half_years = one_tenth_years;
half_fish = feature_fish;

while (feature_fish > current_fish / 10) {
    feature_fish = feature_fish * (1 - decrease_ratio);
    one_tenth_years = one_tenth_years + 1;
}

cout << "\nAfter " << half_years << " years, the fish will be ";
cout << half_fish << ", \nWhich is less than half of current " << current_fish << " fishes!";
cout << "\nAfter " << one_tenth_years << " years, the fish will be ";
cout << feature_fish << ", \nWhich is less than one tenth of current ";
cout << current_fish << " fishes!";
return 1;
}
```

## 第四章 函数

### 填空、选择题

1 在程序设计过程使用函数符合\_\_A\_\_的思维方式，是自顶向下、逐步求精技术的体现。另外函数的使用还提高了程序的\_\_B\_\_，因为我们可在不同环境下使用这些函数。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 分而治之；B 可重用性

2 函数的声明有两种形式 引用性声明和定义性声明，引用性声明又称为\_\_A\_\_，只说明函数的\_\_B\_\_、\_\_C\_\_和\_\_D\_\_。而定义性声明又称为\_\_E\_\_，它还要给出\_\_F\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 函数原型；B 返回类型；C 函数名称；D 形式参数；E 函数定义；F 函数体

3 函数的类型是指\_\_A\_\_，可以是基本数据类型，也可以是结构类型等复合类型，但不能是\_\_B\_\_或\_\_C\_\_。函数没有返回值的准确说法是\_\_D\_\_。函数类型被省略时，其类型是\_\_E\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 函数的返回类型；B 数组类型；C 函数类型；D 函数的返回类型为空值类型；E 整数类型 int

4 在\_\_A\_\_的时候，对应函数的形式参数必须给出实际参数，实际参数的\_\_B\_\_必须和对应形式参数的类型相匹配，在函数不具有\_\_C\_\_和\_\_D\_\_时，实际参数的个数也必须与形式参数相同。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 调用函数；B 类型；C 缺省参数；D 可变参数

5 当程序越来越大的时候，通常将程序划分为模块，在C++语言中使用\_\_A\_\_作为划分模块的主要机制。当一个程序由多个模块时，函数名字的\_\_B\_\_原则会产生影响，这时需要使用编译指令\_\_C\_\_将其他模块函数的\_\_D\_\_包含进来。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 文件；B 先声明、后使用；C #include；D 原型说明

6 函数调用者和函数之间交换信息的方式通过参数传递和\_\_A\_\_。C++语言提供了两种参数传递方式 \_\_B\_\_ 和\_\_C\_\_。其中方式B也称为\_\_D\_\_，它在函数对参数修改时并不影响到函数调用者，

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 返回值；B 按值调用；C 按引用调用；D 按拷贝调用

7 具有\_\_A\_\_的形式参数称为缺省参数，当函数有多个形式参数时，缺省参数必须从右至左定义，一个缺省参数的右边不能有\_\_B\_\_的参数。定义缺省参数可以在\_\_C\_\_中给出，也可在\_\_D\_\_时给出，但不允许两者重复给出。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 缺省值；B 未指定缺省值；C 函数原型；D 函数定义

8 C++程序中，在一个块内说明的标识具有\_\_A\_\_，在函数定义时的形式参数具有\_\_B\_\_，在函数原型的形式参数具有\_\_C\_\_，在函数及类以外声明的标识符具有\_\_D\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 块作用域；B 块作用域；C 函数原型作用域；D 文件作用域

9 在\_\_A\_\_中声明的变量称为全局变量，在块作用域中声明的变量称为\_\_B\_\_，其中\_\_C\_\_也可作用函数之间通信的一种方式，但它破坏了模块之间的独立性。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 文件作用域；B 局部变量；C 全局变量

10 变量的生存期是指在程序运行过程中变量\_\_A\_\_的时间。变量的生存期是由声明变量时选择的\_\_B\_\_决定的，B也同时影响着编译程序为变量\_\_C\_\_的方式及为变量设置的\_\_D\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 被实际分配内存；B 存储类别；C 分配内存的方式；D 初始值

11 C++语言的变量有四种存储类别 \_\_A\_\_、\_\_B\_\_、\_\_C\_\_和\_\_D\_\_，分别用保留字auto、register、static和extern表示。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 自动的；B 寄存器的；C 静态的；D 外部的

12 如果声明\_\_A\_\_与\_\_B\_\_时没有进行初始化,则它们的初始值是随机的,因此我们在使用这两种变量之前应该进行\_\_C\_\_。如果在声明静态变量时未指定初始化表达时,则其初始值缺省为\_\_D\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 自动变量; B 寄存器变量; C 初始化; D 0

13 静态变量既可以是局部变量,也可能是全局变量,但无论是哪一种情况,它都具有\_\_A\_\_,被声明为静态的全局变量其作用域\_\_B\_\_。自动变量与寄存器变量的生存期则是从他们\_\_C\_\_的地方开始,到其\_\_D\_\_时生存期结束,这种生存期称为局部寿命。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 全局寿命; B 仅限于该文件之内; C 被声明; D 作用域结束

14 声明外部全局变量时,带有初始化表达式的声明语句称为\_\_A\_\_,否则成为\_\_B\_\_,如果一个全局变量在声明时没有指定存储类别,则缺省存储类别是\_\_C\_\_的,并且该声明是\_\_D\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 定义性声明; B 引用性声明; C 外部; D 定义性声明

15 作为\_\_A\_\_声明出现的外部变量必须是全局变量,而\_\_B\_\_声明中出现的变量既可是全局变量,也可能是局部变量。在一个程序中,一个变量有且仅有一次\_\_C\_\_,而可以有\_\_D\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 定义性; B 引用性; C 定义性声明; D 引用性声明

16 有如下函数原型说明

```
static int function(int *para_one, float &para_two, int para_three);
```

则参数para\_one的传递方式是\_\_\_\_\_, para\_two的传递方式是\_\_\_\_\_, para\_three的传递方式是\_\_\_\_\_。

A 按值传递 B 按引用传递 C 按指针传递

难易程度 A 较易

认知程度 理解

预计答题时间 1 分钟

参考答案 A; B; A

17 设一个程序只有两个文件 main.cpp 和 function.cpp ,其中在文件 function.cpp 中有如下函

---

### 数原型说明

static int some\_function(int parameter);

请选出下面正确陈述(多重选择) \_\_\_\_\_

- A 在文件 main.cpp 中的函数可调用函数 some\_function。
- B 在文件 function.cpp 中的函数可调用函数 some\_function。
- C 函数 some\_function 可在 main.cpp 中实现，即在 main.cpp 给出该函数的定义性声明。
- D 函数 some\_function 可在 function.cpp 中实现。

难易程度 A 较易

认知程度 理解

预计答题时间 1 分钟

参考答案 B、D

18 设一个程序只有两个文件 main.cpp 和 function.cpp ,其中文件 function.cpp 的全部内容如下

```
extern int strange_var;
extern int function_one(int para_one);
static int square(int num);
int square(int num)
{
    return num * num;
}
int function_one(int para_one)
{
    int sum, index;
    for (sum = 0, index = 0; index < para_one; index++) sum = sum + square(index);
    strange_var = sum;
    return sum;
}
```

下面陈述中正确的是 (多重选择) \_\_\_\_\_。

- A 如果在 main.cpp 中包括编译指令#include “function.cpp”,则在该文件中可调用文件 function.cpp 中的函数 square。
- B 如果在 main.cpp 中包括语句 extern int square(int num),则在该文件中可调用文件 function.cpp 中的函数 square。
- C 如果整个程序要编译通过,则文件 main.cpp 中必包括如下两条语句之一  
extern int strange\_var = 0; 或 int strange\_var = 0;  
其中常量 0 可换成其他合法的 C++常量。
- D 在文件 main.cpp 中可包括如下语句 extern int function\_one(int para\_one);

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案 C、D

19 设有文件 main.cpp 的内容如下

```
int strange_var;
```



---

```

int function(int strange_para)
{
    int strange_int
    int condition = 0;
    if (condition) {
        float strange_var;
        // 此处为下面所说的 A 点。
    }
    // 此处为下面所说的 B 点。
    return 0;
}
// 此处为下面所说的 C 点。

```

则在A点可见的标识符包括（多重选择）\_\_\_\_\_，B点可见的标识符包括\_\_\_\_\_，C点可见的标识符包括\_\_\_\_\_。

- |   |                       |   |              |   |             |
|---|-----------------------|---|--------------|---|-------------|
| A | int 类型的 strange_var   | B | strange_para | C | strange_int |
| D | float 类型的 strange_var | E | condition    | F | function    |

难易程度      B    中等难度  
 认知程度          理解  
 预计答题时间      3    分钟  
 参考答案

B、C、D、E、F  
 A、C、D、E、F  
 A、F

20 设有文件 function.cpp 的内容如下

```

int var_one;
int function(int para_one)
{
    int var_two;
    static var_three
    .....
}

```

则变量var\_one是\_\_\_\_\_，变量var\_two是\_\_\_\_\_，变量var\_three是\_\_\_\_\_。

- |   |        |   |        |   |        |
|---|--------|---|--------|---|--------|
| A | 全局静态变量 | B | 局部静态变量 | C | 全局外部变量 |
| D | 局部外部变量 | E | 局部自动变量 |   |        |

难易程度      A    较易  
 认知程度          理解  
 预计答题时间      2    分钟  
 参考答案

C ;                  B ;                  E

名词解释

21 函数原型

- 
- 22 缺省参数
  - 23 标识符的作用域
  - 24 局部变量和全局变量
  - 25 变量生存期
  - 26 直接递归与间接递归

#### 程序改错题

指出下列函数原型说明中是否有错误，有则改正

27 `static int calc_middle(x, y, z);`

难易程度    A    较易

认知程度        识记

预计答题时间    0.5 分钟

参考答案        各个参数没有类型说明

28 `extern int calc_average(int 1_score, int 2_score, int 3_score);`

难易程度    A    较易

认知程度        识记

预计答题时间    0.5 分钟

参考答案        各个参数不是合法的标识符

29 `extren float 3rd_function(int x, int y, int z);`

难易程度    A    较易

认知程度        识记

预计答题时间    0.5 分钟

参考答案        函数名字不是合法的标识符

30 `extern char who_am_i(int index; int max_student);`

难易程度    A    较易

认知程度        识记

预计答题时间    0.5 分钟

参考答案        参数之间应该使用逗号分开

31 `int no_para_func();`

难易程度    A    较易

认知程度        识记

预计答题时间    0.5 分钟

参考答案        正确

32 `extern this_is_procedure;`

难易程度    A    较易

认知程度        识记

预计答题时间    0.5 分钟

参考答案        没有参数说明，C++语言中没有类似过程的概念。

---

33 int what\_to\_do(int x, int y, int z = 0, int w = 0);

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 正确

34 int how\_to\_do(int x = 0, int y = 0, int z, int w);

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 缺省参数右边还有未指定缺省值的参数

35 char look\_up(char word, char dictionary, char word);

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 参数重复说明

36 int check\_parameter(int int\_or\_float, int x, float x);

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 正确

37 指出如下程序中的错误

```
#include <iostream.h>

extern int ext_var_one = 1;
extern int ext_var_two;

int main()
{
    extern int ext_var_one;
    static int main_var_one;
    extern int ext_var_two = 0;

    main_var_one = ext_var_one + ext_var_two;
    cout << "\n" << main_var_one;
    return 1;
}
```

难易程度 B 中等难度

认知程度 理解

预计答题时间 2 分钟

参考答案 在 main 函数中不能有外部变量的定义性声明。

38 假定以下是一个完整的程序,即没有其它文件与该程序有关(除编译系统提供的头文件 `iostream.h` 外)。试指出该程序中的语法错误,并改正。

```
#include <iostream.h>

int main()
{
    int num;

    global_int = 1;
    num = 1;

    while (global_int < 10) num = num + 1;

    cout << "At last num = " << num;
    return 0;
}

extern int global_int;
```

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 3 分钟

参考答案

在 `main()` 函数中,变量 `global_int` 还没有定义,不能使用。  
不存在全局变量 `global_int` 的定义性声明。

39 假定以下是一个完整的程序,即没有其它文件与该程序有关(除编译系统提供的头文件 `iostream.h` 外)。试指出该程序中的语法错误,并改正。

```
#include <iostream.h>

int main()
{
    int num;

    num = 1;

    while (num < 10) num = num + function(num);

    cout << "At last num = " << num;
    return 0;
}

extern global_int = 0;

int function (int num)
{
    if ((global % 2) == 0 {
        int result;
        result = global_int * num;
    } else {
        int num = global_int;

        num = num * global_int;
        result = num + global_int;
    }
}
```

```
}  
    return result;  
}
```

难易程度     B   中等难度

认知程度         简单应用

预计答题时间     5     分钟

参考答案

在函数 func()的 if 语句中 else 分支的块语句中不能使用 result ,因为 result 的作用域仅限于 if 语句的 if 分支的块语句。

在函数 func()的返回语句中不能使用 result ,因为 result 的作用域仅限于 if 语句的 if 分支的块语句。

40 假定以下是一个完整的程序,即没有其它文件与该程序有关(除编译系统提供的头文件 iostream.h 外)。试指出该程序中的语法错误,并改正。

```
#include <iostream.h>  
  
int main()  
{  
    int num;  
  
    num = 1;  
  
    while (num < 10) num = num + function(num);  
  
    cout << "At last num = " << num;  
    return 0;  
}  
  
extern global_int = 0;  
  
int function (int num)  
{  
    int num;  
    int result;  
  
    if ((global % 2) == 0 {  
        result = global_int * num;  
    } else {  
        int num;  
        num = ::num * global_int;  
        result = num + global_int;  
    }  
    return result;  
}
```

难易程度     B   中等难度

认知程度         简单应用

预计答题时间     5     分钟

参考答案

在函数 func()中第三行重复定义变量 num ,与参数说明中的 num 重复。

在函数 func()的 if 语句中 else 分支的块语句中不能使用::num ,因为不存在全局的 num 变量

程序输出题

41 给出如下函数

```
void print_num(int first, int second)
{
    cout << first << " " << second << " " << first + second << "\n";
};
```

请问用一系列什么样的函数调用可输出（提示 可引入中间变量）：

```
1  1  2
3  5  8
13 21 34
```

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 5 分钟

参考答案 调用该函数的主函数可如下编写

```
int main()
{
    int first = 1, second = 1, temp, i;

    for (i = 1; i <= 3; i++) {
        temp = first + second;
        print_num(first, second);
        first = second + temp;
        second = first + temp;
    };
    return 1;
}
```

42 请给出下面程序的输出结果

```
#include <iostream.h>

int print_num(int first, int second)
{
    cout << first << " " << second << " " << first + second << "\n";
    return (first+second);
}

int main()
{
    int number_one, number_two;

    cout << "\n";
    number_one = 1; number_two = 1;
    number_one = print_num(number_one, number_two);
    number_two = print_num(number_one, number_two);
}
```

```
print_num(number_one, number_two);  
return 1;  
}
```

难易程度     B   中等难度

认知程度         简单应用

预计答题时间     5     分钟

参考答案

1 1 2

2 1 3

2 3 5

43 试给出下面程序的输出。

```
#include <iostream.h>  
  
int global_value = 1;  
  
void test_function(void)  
{  
    int local_value = 1;  
    static int static_value = 1;  
  
    cout << "\nThe global value: " << global_value;  
    cout << ", The local value: " << local_value;  
    cout << ", The static value: " << static_value;  
  
    global_value = global_value + 1;  
    local_value = local_value + 1;  
    static_value = static_value + 1;  
}  
  
int main()  
{  
    while (global_value < 4) test_function();  
    return 0;  
}
```

难易程度     B   中等难度

认知程度         简单应用

预计答题时间     5     分钟

参考答案

The global value: 1, The local value: 1, The static value: 1

The global value: 2, The local value: 1, The static value: 2

The global value: 3, The local value: 1, The static value: 3

44 请给出下面程序的输出结果。

```
#include <iostream.h>  
  
int function(int x, int y)  
{  
    if (y == 1) return x;
```

```

        else return (x + function(x, y-1));
    }

int main()
{
    cout << "\nThe result of function(5, 8) is " << function(5, 8);
    return 1;
}

```

难易程度     B   中等难度

认知程度         简单应用

预计答题时间     5     分钟

参考答案

The result of function(5, 8) is 40

45 请给出下面程序的输出结果

```

#include <iostream.h>

int function(int number, int is_first)
{
    int factor;

    if (number == 1) return 1;

    factor = 2;
    while (number % factor != 0) factor = factor + 1;
    if (is_first == 1) cout << "\n" << number << " = " << factor;
    else cout << " * " << factor;

    function(number / factor, 0);
    return 1;
}

int main()
{
    cout << "\nThe result of function(12, 1) is: ";
    function(12, 1);
    return 1;
}

```

难易程度     C   难度较大

认知程度         综合应用

预计答题时间     10     分钟

参考答案

The result of function(12, 1) is:

12 = 2 \* 2 \* 3



### 程序填空题

46 下面函数用来计算一个整数（即参数 number）的某一进制（即参数 base）表示下各位上数字的和，例如调用函数 bit\_sum(123, 10)，该函数应该返回值为 1+2+3=6，试填写下面函数的空，使其完成正确的功能。

```
int bit_sum(int number, int base)
{
    int sum;
    int temp_number;

    temp_number = number; sum = 0;
    while (temp_number > 0) {
        sum = sum + _____;
        temp_number = _____;
    }
    return sum;
}
```

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 5 分钟

参考答案

temp\_number % base

temp\_number / base

47 下面函数用来检查整数 number 是否是质数。请完成其中的空

```
int is_prime(int number)
{
    unsigned int factor;

    factor = 2;
    while (_____) {
        if (_____) return 0;
        factor = factor + 1;
    }
    return 1;
}
```

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 5 分钟

参考答案

factor \* factor <= number 或者 factor < number

number % factor == 0

48 下面函数用来检查整数 number 在 base 进制表示下，是否含有数字 bit，即 number 中是否存在某位的值是 bit，请完成其中的空。

```
int check_digit(unsigned int number, int bit, int base = 10)
{
    while (number > 0) {
        if (_____) return 1;
        number = _____;
    }
    return 0;
}
```

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 5 分钟

参考答案

```
number % base == bit
number / base
```

49 下面函数用来返回整数 number 在 base 进制下，从右往左数第 index 位的数字，请填写其中的空。

```
int digit(unsigned int number, int index, int base = 10)
{
    int count;

    count = 0;
    while (_____) {
        number = _____;
        count = count + 1;
    }

    return _____;
}
```

难易程度 B 中等难度

认知程度 简单应用

预计答题时间 5 分钟

参考答案

```
number > 0 && count < (index-1)
number / base
(number % base)
```

50 下面函数用来将一个整数 number 转换成相应的字符串输出，例如如果 number=1234，那么程序将输出字符串"1234"，请完成其中的空。

```

#include <iostream.h>

int int2str(int number, int base = 10)
{
    unsigned long max_power;
    int bit_num;

    max_power = 1;
    while (max_power < number) max_power = _____;
    max_power = max_power / base;

    cout << "\n";
    while (max_power > 0) {
        bit_num = _____;
        cout << bit_num;
        max_power = _____;
    }
    return 1;
}

```

难易程度     B   中等难度

认知程度         简单应用

预计答题时间     5   分钟

参考答案

```

max_power * base
(number / max_power) % base
max_power / base

```

51 下面程序定义了函数 factor\_sum 用来求一个整数的各个因子的和，主函数中利用该函数找出自然数 1-1000 中的完全数和亲和数对（所有亲和数 a, b 是指 a 的各个因子之和等于 b，而 b 的各个因子之和等于 a）。试完成其中的空。

```

#include <iostream.h>
#include <iomanip.h>

int factor_sum(int number)
{
    int factor, factor_square, sum;

    sum = 1; factor = 2; factor_square = factor * factor;
    while (factor_square <= number) {
        if (_____) {
            if (factor_square != number) {
                sum = sum + _____;
            } else sum = _____;
        }
    }
}

```

```

    }
    factor = factor + 1;    factor_square = factor * factor;
}
return sum;
}

int main()
{
    unsigned int number, other_number;
    const unsigned int max_number = 1000;

    cout << "\n";
    for (number = 2; number < max_number; number = number + 1) {
        other_number = factor_sum(number);
        if (_____) {
            // 发现完全数
            cout << "\nFind a complete number: " << number;
        } else {
            if (_____) {
                // 发现亲和数对
                cout << "\nFind <" << number << ", " << other_number << "> is twain number!";
            }
        }
    }
    return 0;
}

```

难易程度     C   难度较大

认知程度         总和和应用

预计答题时间     10   分钟

参考答案

```

number % factor == 0
factor + number / factor
sum + factor
other_number == number
factor_sum(other_number) == number

```

52 正读和反读都相同的数称为回文数。以下程序输入一个整数 `max_num`，输出从 0 到 `max_num` 中用二进制表示和十进制表示都是回文数的整数，其中函数 `is_circle_number` 判断一个数在某个进制表示下是否是回文数。例如，整数 313 就是该程序输出的一个数，因为它的二进制表示为 100111001。请填写以下程序中的空格，使其完成上面所说的功能。

```

#include <iostream.h>
#include <iomanip.h>           // Include file <iomanip.h> for function setw().

// 函数 is_circle_num 判断 number 在 radius 进制表示下是否是回文数，如果是返回 1，否则返回 0。

```

```

static int is_circle_num(int number, int radius);

int main()
{
    int num, max_num;

    cout << "Please input a integer (such as 1200) :";
    cin >> max_num;
    for (num = 0; num < max_num; num = num + 1) {
        if (is_circle_num(num, 10) == 1 && is_circle_num(num, 2) == 1) {
            // setw(6)使得输出每个回文数时占六位的宽度。
            cout << setw(6) << num;
            cout << " is a circle num in decimal and binary system!\n";
        }
    }
    return 0;
}

int is_circle_num(int number, int radius)
{
    int left_num, right_num;
    int max_power, left_power, right_power;

    max_power = 1;
    while (_____) {
        max_power = max_power * radius;
    }

    left_power = max_power;
    right_power = radius;

    while (_____) {
        right_num = _____;
        left_num = _____;

        if (right_num != left_num) return 0;

        right_power = right_power * radius;
        left_power = left_power / radius;
    }
    return 1;
}

```

难易程度      C    难度较大

认知程度          综合应用

预计答题时间    10    分钟

参考答案

```

(number / max_power) >= radius
left_power >= right_power
(number % right_power) / (right_power / radius)
(number / left_power) % radius

```

程序编写题

---

53 写两个函数计算从  $m$  中选  $n$  的组合数，即计算  $C_m^n$ ，一个函数不使用递归的方法计算，一个函数使用递归方法计算。函数原型可分别声明如下

```
int comb_loop(int m, int n);
```

```
int comb_recursion(int m, int n);
```

54 编写一个程序，由用户输入年、月、日，计算该日是该年的第几天。

55 利用 C++ 语言提供的库函数编写一个发牌程序，每次从一副扑克牌中随机发三张牌。

56 利用随机函数计算  $\pi$  的值。将随机数的前后两个值(均在 0-1 之间)作为点的  $x, y$  坐标。统计 1 万个点中，落入半径为 1 的  $1/4$  圆内的点数  $n$ ，从而计算出  $1/4$  的面积  $s=n/10000$ ，又知  $1/4$  的圆面积为  $\pi/4$ ，由此可计算出  $\pi$  的值。

---

## 第五章

### 填空、选择题

1 C++语言引入类主要起两个作用     A    和     B    。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 作为数据封装机制；B 作为类型定义机制

2 C++语言中的类用作数据封装机制，它将相关的数据以及作用在这些数据上的  A  组织在一起，实现了  B  。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 操作；B 信息隐藏

3 C++语言中的类用作类型定义机制，它可用于声明  A  并提供相应的类型检查功能，更进一步还允许用户通过  B  进一步定义子类型。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 对象；B 继承机制

4 C++语言类中定义的数据和函数分别称为  A  和  B  。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 数据成员；B 成员函数

5   A  可以改变局部变量、全局变量或其他类对象的值，但不允许修改本类中的数据成员的值。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 常量成员函数

6 按照成员访问方式，C++中的成员可分为三种   A  、  B  和  C  。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 公有成员；B 保护成员；C 私有成员

7 C++中类的构造函数的名字必须  A  相同，而且不能指定  B  ，即使是void

---

类型也不允许。

难易程度     A   较易

认知程度         识记

预计答题时间     0.5 分钟

参考答案         A   类名字；B   返回类型

8   在C++程序中，  A  和  B  的构造函数在主函数main()函数执行之前被自动调用。

难易程度     A   较易

认知程度         识记

预计答题时间     0.5 分钟

参考答案         A   全局对象；B   静态对象

9   C++类中的对象成员的构造函数的调用次序取决于这些对象成员  A  ，与它们在初始化列表中出现的次序无关。

难易程度     A   较易

认知程度         识记

预计答题时间     0.5 分钟

参考答案         A   在类中声明的次序

10  创建类C的一个新对象时，将先调用  A  的构造函数，然后再执行类C的构造函数。

难易程度     A   较易

认知程度         识记

预计答题时间     0.5 分钟

参考答案         A   成员对象

11  利用  A  我们可实现单实例类。

难易程度     A   较易

认知程度         识记

预计答题时间     0.5 分钟

参考答案         A   类的静态数据成员

名词解释

12  信息隐藏

难易程度     B   中等难度

认知程度         理解

预计答题时间     3   分钟

参考答案         将一个模块的内部数据或操作隐藏起来，其他模块只能通过该模块提供的公共界面来放这些内部数据的保护措施成为信息隐藏。

13  封装

难易程度     B   中等难度

认知程度         理解



---

预计答题时间 3 分钟

参考答案 将一组数据与这些数据上的相关操作组织在一起的措施称为封装

#### 14 类

难易程度 B 中等难度

认知程度 理解

预计答题时间 3 分钟

参考答案 类是对一群具有相同属性、表现相同行为的对象的描述。

#### 程序改错题

15 下列的类声明程序片段是否有语法错误，如有请改正

```
class POINT {  
    public:  
        int get_x();  
        int get_y();  
    private:  
        int x, y;  
}
```

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 类声明最后右花括号后少了分号作为类声明的结束。

16 下列的类声明程序片段是否有语法错误，如有请改正

```
class LINE {  
    public:  
        int draw();  
        int is_on_line(int x, int y);  
    private:  
        int start_x = 0, start_y = 0;  
        int end_x, end_y;  
};
```

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 类中数据成员不能初始化。

17 下列的类声明程序片段是否有语法错误，如有请改正

```
class RECTANGLE {  
    public:  
        int draw();  
        int is_in_rectangle(int x, int y);  
    private:
```

---

```
        int left_up_x, left_up_y;  
        int right_bottom_x, right_bottom_y;  
    };
```

难易程度     A   较易

认知程度         识记

预计答题时间     0.5   分钟

参考答案         正确。

18 下列的类声明程序片段是否有语法错误，如有请改正

```
class SQUARE {  
public:  
    void SQUARE(int x = 0, int y = 1, int len = 1);  
    ~SQUARE();  
    int draw();  
private:  
    int left_up_x, left_up_y;  
    int side_len;  
};
```

难易程度     A   较易

认知程度         识记

预计答题时间     0.5   分钟

参考答案         不能为构造函数指定返回类型。

19 下列的类声明程序片段是否有语法错误，如有请改正

```
class CIRCLE {  
public:  
    CIRCLE(int x = 0, int y = 0, int radius = 1);  
    ~CIRCLE();  
    int draw();  
private:  
    register int atom_x, atom_y;  
    auto int radius;  
};
```

难易程度     A   较易

认知程度         识记

预计答题时间     0.5   分钟

参考答案         类中数据成员不能指定存储类别。

20 下列的类声明程序片段是否有语法错误，如有请改正

```
class FIGURE {  
public:  
    figure();  
    FIGURE();  
    ~FIGURE(int have_drawed);
```

---

```
};
```

难易程度     A   较易  
认知程度         识记  
预计答题时间     0.5   分钟  
参考答案         不能为析构函数指定参数。

21-23     设有如下的类声明

```
class PRINTER {  
    unsigned char flag;  
    int port_address;  
    int print_page();  
public:  
    PRINTER(int address, unsigned char init_flag);  
    ~PRINTER();  
    int set_address(int new_address);  
    int set_flag(unsigned char new_flag);  
    int print_file();  
};
```

假定该类已被正确实现，试问下列使用该类的程序片段是否有错误（这些程序片段所在的程序已正确地使用#include 包含了类\_PRINTER 的声明），如有错误请改正

```
21   PRINTER LQ_1600K(0x3F8, '\0');  
     LQ_1600K.print_file();  
     LQ_1600K.print_page();
```

难易程度     A   较易  
认知程度         识记  
预计答题时间     0.5   分钟  
参考答案         不能为调用类的私有成员函数 print\_page()

```
22   PRINTER LASERJET();  
     LASERJET.flag = (unsigned char)(0xFF);  
     LASERJET.address = 0x3F9;  
     LASERJET.print_file();
```

难易程度     A   较易  
认知程度         识记  
预计答题时间     0.5   分钟  
参考答案         在声明对象 LASERJET 时要给出参数，不能为访问类的私有成员数据。

```
23   PRINTER DESKJET;  
     DESKJET.set_flag('\x7F');  
     DESKJET.set_address(0x3F8);  
     DESKJET.print_file();
```

难易程度     A   较易  
认知程度         识记

预计答题时间 0.5 分钟

参考答案 在声明对象 DESKJET 时要给出参数。

24 指出下面程序中的语法错误

```
#include <iostream.h>

class POINT {
public:
    POINT(int x, int y): x(x), y(y) {}
    int get_x() { return x; }
    int get_y() { return y; }
    void print()
    {
        cout << "Point: (" << get_x() << ", " << get_y() << ")";
    }
private:
    int x = 0;
    int y = 0;
};

class LINE {
public:
    LINE(int x0, int y0, int x1, int y1)
    {
        start.x = x0; start.y = y0;
        end.x = x1; end.y = y1;
    }
    POINT get_start() { return start; }
    POINT get_end() { return end; }
    void print()
    {
        cout << "Line from " << start.print();
        cout << " To " << end.print() << ".\n";
    }
private:
    POINT start, end;
};

int main()
{
    LINE line(0, 0, 50, 50);

    line.print();
    return 0;
}
```

```
}
```

难易程度 C 难度较大

认知程度 综合应用

预计答题时间 12 分钟

参考答案

类 POINT 的数据成员不能初始化

类 LINE 的构造函数需要提供成员对象 start, end 的初始化

在 LINE 的构造函数中不能访问类 POINT 的私有数据成员

类 POINT 的 print() 函数没有返回值, 因此不能直接用在类 LINEprint() 函数中的 cout 语句中以输出。

程序输出题

25 请给出下面程序的输出结果

```
#include <iostream.h>

class CLASS_NAME {
public:
    CLASS_NAME()
    {
        cout << "\nCall Construction of CLASS_NAME.";
    }
    int print()
    {
        cout << "\nThis is CLASS_NAME."; return 1;
    }
    ~CLASS_NAME()
    {
        cout << "\nCall Destruction of CLASS_NAME.";
    }
};

CLASS_NAME object_name;

int main()
{
    cout << "\nProgram begin...";
    object_name.print();
    cout << "\nProgram end...";
    return 1;
}
```

难易程度 B 中等难度

---

认知程度          简单应用

预计答题时间    5    分钟

参考答案

Call Construction of CLASS\_NAME.

Program begin...

This is CLASS\_NAME.

Program end...

Call Destruction of CLASS\_NAME.

26 请给出下面程序的输出结果

```
#include <iostream.h>

class MEMBER {
public:
    MEMBER(int identify = 0)
    {
        cout << "\nConstruct a MEMBER, which identify is " << identify;
        this->identify = identify;
    }
    ~MEMBER()
    {
        cout << "\nDestruct a MEMBER, which identify is " << identify;
    }
private:
    int identify;
};

class CONTAINER {
public:
    CONTAINER(): member_two(1)
    {
        cout << "\nConstruct a CONTAINER!";
    }
    ~CONTAINER()
    {
        cout << "\nDestruct a CONTAINER!";
    }
private:
    MEMBER member_one, member_two;
};

int main()
{
```

```
CONTAINER container;  
return 1;  
}
```

难易程度    B    中等难度

认知程度        简单应用

预计答题时间    5    分钟

参考答案

Construct a MEMBER, which identify is 0

Construct a MEMBER, which identify is 1

Construct a CONTAINER!

Destruct a CONTAINER!

Destruct a MEMBER, which identify is 1

Destruct a MEMBER, which identify is 0

27 请给出下面程序的输出结果

```
#include <iostream.h>  
  
class SINGLE_INSTANCE {  
public:  
    SINGLE_INSTANCE()  
    {  
        instance_count = instance_count + 1;  
        cout << "\nCan construction, and instance_count = " << instance_count;  
        if (instance_count > 1) cout << "\nWarning: more than one object instance!";  
    }  
    ~SINGLE_INSTANCE()  
    {  
        instance_count = instance_count - 1;  
    }  
private:  
    static int instance_count;  
};  
  
int SINGLE_INSTANCE::instance_count = 0;  
  
int main()  
{  
    cout << "\nProgram begin...";  
    SINGLE_INSTANCE object_one;  
    SINGLE_INSTANCE object_two;  
    cout << "\nProgram end...";  
    return 1;  
}
```

难易程度    B    中等难度

认知程度          简单应用  
预计答题时间      5    分钟  
参考答案

Program begin...

Can construction, and instance\_count = 1

Can construction, and instance\_count = 2

Warning: more than one object instance!

Program end...

28 请给出下面程序的输出结果。

```
#include <iostream.h>

const unsigned long MEGA_BYTE = (unsigned long)1024 * 1024;

class COMPUTER {
public:
    COMPUTER(unsigned long m_size, unsigned long h_size, int CD_speed)
    {
        memory_size = m_size;
        hard_disk_size = h_size;
        CD_ROM_speed = CD_speed;
        if (memory_size > (unsigned long)16 * MEGA_BYTE) {
            cout << "\nYou should install Windows 95!";
        } else {
            cout << "\nYou can only install Windows 3.1!";
        }
        if (hard_disk_size > (unsigned long)800 * MEGA_BYTE) {
            cout << "\nYou can install Visual C++ 4.0!";
        } else {
            cout << "\nYou can only install Borland C++ 4.5!";
        }
        if (CD_ROM_speed > 4) cout << "\nYou can see Video CD!";
        else cout << "\nYou can only listen Audio CD!";
    }

    ~COMPUTER()
    {
        cout << "\nYou should make good use of your computer!";
    }

    void print_configuration()
    {
        cout << "\nYour computer configuration is: ";
        cout << "\nMemory Size: " << (memory_size / MEGA_BYTE) << " Mbyte";
        cout << "\nHard Disk Size: " << (hard_disk_size / MEGA_BYTE) << " Mbyte";
        cout << "\nCD ROM Speed: " << CD_ROM_speed << " Speed";
    }

private:
    unsigned long memory_size;
    unsigned long hard_disk_size;
    unsigned int CD_ROM_speed;
};

int main()
```



```

{
    COMPUTER my_computer(32 * MEGA_BYTE, 1024 * MEGA_BYTE, 8);

    cout << "\nHi you get a good computer!!!";
    my_computer.print_configuration();
    return 0;
}

```

难易程度     C   较难

认知程度         综合应用

预计答题时间     8   分钟

参考答案

You should install Windows 95!

You can install Visual C++ 4.0!

You can see Video CD!

I get a good computer!!!

Your computer configuration is:

Memory Size: 32 Mbyte

Hard Disk Size: 1024 Mbyte

CD ROM Speed: 8 Speed

You should make good use of your computer!

29 请给出下面程序的输出结果。

```

#include <iostream.h>

const int WtMAIN = 0;     // The window type: MAIN window.
const int WtWORK = 1;     // The window type: WORK window.
const int WtMESSAGE = 2; // The window type: MESSAGE window.

class WINDOW {
public:
    WINDOW(int window_type);
    ~WINDOW();
    int get_window_number() { return count; }

private:
    static int count;
    int type;
};

int WINDOW::count = 0;     // Initialize the static data member of class _WINDOW

WINDOW::WINDOW(int window_type)
{
    count = count + 1;
    type = window_type;
    switch (type) {
        case WtMAIN: cout << "\nOpen a main window!"; break;
        case WtWORK: cout << "\nOpen a work window!"; break;
        case WtMESSAGE: cout << "\nOpen a message window!"; break;
        default: cout << "\nOpen a unknow type window!"; break;
    }
}

```

```

WINDOW::~~WINDOW()
{
    count = count - 1;
    switch (type) {
        case WtMAIN: cout << "\nClose a main window!"; break;
        case WtWORK: cout << "\nClose a work window!"; break;
        case WtMESSAGE: cout << "\nClose a message window!"; break;
        default: cout << "\nClose a unknow type window!"; break;
    }
}

class SCREEN {
public:
    SCREEN(): main_window(WtMAIN), work_window(WtWORK)
    {
        cout << "\nInitialize the screen!";
    }

    ~SCREEN()
    {
        cout << "\nClose the screen!";
    }

    int get_window_number() { return main_window.get_window_number(); }

private:
    WINDOW work_window;
    WINDOW main_window;
};

int main()
{
    SCREEN screen;
    WINDOW message_window(WtMESSAGE);

    cout << "\nThere are " << screen.get_window_number() << " window(s) on screen!";
    cout << "\nThere are " << message_window.get_window_number() << " message window(s)!";
    return 0;
}

```

难易程度 C 较难

认知程度 综合应用

预计答题时间 8 分钟

参考答案

Open a work window!

Open a main window!

Initialize the screen!

Open a message window!

There are 3 window(s) on screen!

There are 3 message window(s)!

Close a message window!

Close the screen!

Close a main window!

Close a work window!

30 给出程序

```
#include <iostream.h>
int main()
{
    cout << "\nHello, Everybody!!!";
    return 0;
}
```

请修改该程序使之能产生如下输出

```
Ok, I am coming...
Hello, Everybody!!!
Thanks, Goodbye!
```

但主函数 main()不能作任何修改。

难易程度 C 较难

认知程度 综合应用

预计答题时间 8 分钟

参考答案 可给出程序如下

```
#include <iostream.h>
class MY_CLASS {
public:
    MY_CLASS()
    {
        cout << "Hi, I am coming ... \n";
    }
    ~MY_CLASS()
    {
        cout << "Thanks! Goodbye! \n";
    }
};

MY_CLASS my_object;

int main()
{
    cout << "Hello, everyone! \n";
    return 0;
}
```

类界面设计题，以下各题只要求写出类界面，不需要实现。

31 设计一个平面几何中的位置类 POSITION，该类至少提供移动、求到另一点距离（包括到原点距离）、求 x 坐标、求 y 坐标等操作。

32 设计一个复数类 COMPLEX，尽量完善该类提供的操作。

33 设计一个日期类 DATE，尽量完善该类提供的操作。

---

34 设计一个有理数类 RATIONAL，尽量完善该类的操作。

---

## 第六章

### 填空、选择题

1 数组的每个元素都属于同一类型，该类型是数组的\_\_\_A\_\_\_，我们用\_\_\_B\_\_\_来标识数组的元素，数组的长度是\_\_\_C\_\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 基类型；B 数组下标；C 数组中元素的个数

2 C++语言中提供的空指针名字是\_\_\_A\_\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A NULL

3 设运行环境中int类型长度为2字节，且ptr是一个int类型的指针，且ptr的值是2400，ptr指向的整数变量的值是24，那么ptr+1的值是\_\_\_A\_\_\_，\*ptr+1的值是\_\_\_B\_\_\_。

难易程度 A 较易

认知程度 理解

预计答题时间 1 分钟

参考答案 A 2402；B 25

4 设运行环境中int类型长度为2字节，对以下声明的类A，sizeof(A)的值为 A 。

```
class a {  
public:  
    int get();  
    void set();  
private:  
    static int std;  
    int max, min;  
};
```

难易程度 B 中等难度

认知程度 理解

预计答题时间 2 分钟

参考答案 A 4

5 设运行环境中int类型的长度为2字节，现有声明语句int array[20][2]；，则sizeof(array)的值为\_\_\_A\_\_\_，sizeof(array[1])的值为\_\_\_B\_\_\_，sizeof(array[1][1])的值为\_\_\_C\_\_\_。

难易程度 B 中等难度

认知程度 理解

预计答题时间 2 分钟

参考答案 A 80；B 4；C 2

6 现有声明语句char name[30] = "Zhongshan University" , 那么sizeof(name)的值为A , strlen(name)的值为 B 。

难易程度 A 较易

认知程度 理解

预计答题时间 1 分钟

参考答案 A 30 ; B 20

7 现有 8 商品, 每种商品均有购进价、最低销价、最高销价三种价格。如果程序中需要一个二维双精度数组prices记录这些商品价格, prices第二维依次记录购进价、最低销价、最高销价, 则声明数组prices的语句为 A , 最后一种商品的最低销价为 B 。

难易程度 B 中等难度

认知程度 理解

预计答题时间 2 分钟

参考答案 A double prices[8][3] ; B prices[7][1]

8 organ是一个包含 100 个指向STUDENT类型对象的指针变量数组, 声明organ的语句是 A 。 ptr是一个指针类型的变量, 指向一个长度 100 的、基类型为STUDENT的数组, 声明ptr的语句是 B 。

难易程度 B 中等难度

认知程度 理解

预计答题时间 2 分钟

参考答案 A STUDENT \*organ[100] ; B STUDENT (\*ptr)[100] ;

9 C++语言中用来动态创建对象的运算符为 A , 用来动态撤消对象的运算符为 B 。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A new ; B delete

10 用关键字struct定义的结构类型的成员的缺省访问控制方式是 A , 而有class定义的类型类型的成员的缺省访问控制方式是 B 。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A public ; B private

11 C++语言中的每一个枚举常量代表一个 A 类型的值, 第一个枚举常量的值为 B , 当然我们也可显示地指定枚举常量的值。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 整数 ; B 0

12 在C++语言中我们可使用关键字 A 来引进类型别名。

---

难易程度     A   较易  
认知程度         识记  
预计答题时间    0.5  分钟  
参考答案     A   typedef

13  请说明以下语句的区别：

  声明 int \*ptr;  
  语句 \*ptr = 2400; 与语句 ptr = 2400; 的区别。

难易程度     A   较易  
认知程度         理解  
预计答题时间    1   分钟  
参考答案

14  请说明以下语句的区别：

  声明 int m = 10;  
  语句 \*mptr = &m; 与语句 int \*mptr; \*mptr = m; 的区别。

难易程度     A   较易  
认知程度         理解  
预计答题时间    1   分钟  
参考答案

15  请说明以下语句的区别：

  语句 char name[] = "Computer"; 与  
  语句 char name[] = {'C', 'o', 'm', 'p', 'u', 't', 'e', 'r' }; 的区别。

难易程度     A   较易  
认知程度         理解  
预计答题时间    1   分钟  
参考答案

16  请完成以下变量的声明

- (1)  指向字符的指针变量 char\_ptr ;
- (2)  10 个字符的数组 char\_array ;
- (3)  指向字符串的指针变量 string\_ptr ;
- (4)  10 个字符串的数组 string\_array ;
- (5)  指向字符串数组的指针变量 string\_array\_ptr ;

难易程度     B   中等难度  
认知程度         理解  
预计答题时间    5   分钟  
参考答案

程序改错题

17  请详细说明以下使用数组或指针的程序片段有什么语法错误

  int\* age\_ptr, test\_ptr;

---

```
int m = 3;
*age_ptr = &m; test_ptr = &m;
```

难易程度 B 中等难度  
认知程度 理解  
预计答题时间 2 分钟  
参考答案 \*age\_ptr 具有整数类型，不能接受地址值，同样 test\_ptr 是整数变量，而不是整数指针，从而也不能接受地址值。

18 请详细说明以下使用数组或指针的程序片段有什么语法错误

```
int* salary_ptr;
float* work_hours_ptr = 300.0;
salary_ptr = work_hours_ptr;
```

难易程度 B 中等难度  
认知程度 理解  
预计答题时间 2 分钟  
参考答案 指针赋值时的类型要严格匹配，因此不能将指向浮点类型的指针对值赋给指向整数类型的指针，另外指针 work\_hours\_ptr 不能用双精度常量 300.0 初始化。

19 请详细说明以下使用数组或指针的程序片段有什么语法错误

```
int int_array[10];
int* int_ptr = new int[10];
int_array++; int_ptr++;
int_array = m_ptr;
```

难易程度 B 中等难度  
认知程度 理解  
预计答题时间 2 分钟  
参考答案 数组名字是常量指针，不能使用++运算( int\_array++错 )，也不能赋值(int\_array = m\_ptr 错)。注 int\_ptr++没有错。

20 请详细说明以下使用数组或指针的程序片段有什么语法错误

```
extern int test_func(int);
int (*func)(int);
func = &test_func;
(*func)++;
```

难易程度 B 中等难度  
认知程度 理解  
预计答题时间 2 分钟  
参考答案 函数名字是常量指针，不能使用++运算( (\*func)++错 )。

21 请详细说明以下使用数组或指针的程序片段有什么语法错误

```
char* depart;
char name[30];
int int_array[3];
depart = "Computer";
```



---

```
    name = "Department";
    int_array = { 1, 2, 3 };
```

难易程度     B   中等难度  
认知程度         理解  
预计答题时间     2   分钟  
参考答案     数组名字名字是常量指针，不能使用赋值运算（name = "Department";错，同样 int\_array = { 1, 2, 3 };错）。

22 请详细说明以下使用数组或指针的程序片段有什么语法错误

```
char name[4][] = {"Computer", "Depart", "Software", "Engineer"};
int scores[][] = {{1, 2}, {3, 4}, {5, 6}};
float salary[][2] = {1, 2, 3, 4, 5, 6};
int max_number = 4;
double amount[max_number];
```

难易程度     B   中等难度  
认知程度         理解  
预计答题时间     2   分钟  
参考答案     不能确定数组 name、scores 的长度。声明数组 amount 时不能使用变量 max\_number 作为数组的长度。

24-29     以下程序片段都没有任何语法错误，甚至连警告错误也没有（假设这些程序片段都正确地包含了头文件 iostream.h 和 string.h），但它们在运行时都可能产生致命的错误，请指出这些错误，并说明为什么

```
24 int* age_ptr;
   int age = 30;
   *age_ptr = age;
```

难易程度     B   中等难度  
认知程度         理解  
预计答题时间     2   分钟  
参考答案     指针 age\_ptr 没有动态分配内存，因此不能对\*age\_ptr 进行赋值，因为这时 age\_ptr 指向的地址是随机的。

```
25 const int student_number = 20;
   int scores[student_number];
   for (int i = 1; i <= student_number; i++) scores[i] = 0;
```

难易程度     B   中等难度  
认知程度         理解  
预计答题时间     2   分钟  
参考答案     数组访问越界

```
26 const int name_len = 8;
   char name[name_len] = "Computer";
```

---

```
    char depart[name_len] = {'C', 'o', 'm', 'p', 'u', 't', 'e', 'r'};
    if (strcmp(name, depart) == 0) cout << "\nDepart is equal to name?!";
```

难易程度     B   中等难度  
认知程度         理解  
预计答题时间     2   分钟  
参考答案         数组访问越界

```
27  const int row = 2, col = 2;
      float matrix[row, col];
      for (int j = 0; j < row; j++)
          for (int k = 0; k < col; k++) *((matrix+row*j)+k) = 0.0;
```

难易程度     B   中等难度  
认知程度         理解  
预计答题时间     2   分钟  
参考答案         数组访问越界

```
28  char *department = "Computer Science Department";
      char *university = "Zhongshan University";
      strcpy(department, university);
      cout << "University: " << university << "\n";
      cout << "Depart: " << department << "\n";
      delete department;
      delete university;
```

难易程度     B   中等难度  
认知程度         理解  
预计答题时间     2   分钟  
参考答案         释放字符串常量所占的内存，会引起内存混乱

```
29  const int max_msg_len = 80;
      char *error_message = new char[max_msg_len+1];
      char *prompt_message = new char[max_msg_len+1];
      strcpy(error_message, "Use character pointer error!");
      prompt_message = error_message;
      cout << prompt_message << endl;
      delete prompt_message;
      delete error_message;
```

难易程度     B   中等难度  
认知程度         理解  
预计答题时间     2   分钟  
参考答案         prompt\_message 和 error\_message 互为别名，因此最后的两次 delete 将重复释放内存，因此内存混乱，而为 prompt\_message 分配的内存无法释放，造成内存垃圾。

30 请改正以下程序中的错误，使其能连续五次输出字符串 string

```
#include <iostream.h>
```

```

main()
{
    char string[30] = "Zhongshan University\n";
    char ptr = string;
    // 连续五次输出字符串
    for (int loop = 0; loop <= 4; loop = loop + 1)
        while (*ptr) {
            cout << *ptr;
            ptr = ptr + 1;
        };
    return 0;
};

```

难易程度    B    中等难度  
 认知程度        简单应用  
 预计答题时间    5    分钟  
 参考答案

31 试改正下面使用枚举类型程序的语法错误。

```

int main()
{
    enum COLOR_TYPE { CtWHITE = 2, CtBLACK = 1, CtBLUE } color;
    color = CtBLACK;
    switch (color) {
        case CtWHITE: cout << "\nThe color is white!"; break;
        case CtBLACK: cout << "\nThe color is black!"; break;
        case CtBLUE: cout << "\nThe color is blue!"; break;
    };
    return 0;
}

```

难易程度    B    中等难度  
 认知程度        理解  
 预计答题时间    2    分钟  
 参考答案

程序输出题

32-36 假定下面的各个函数 function()都由以下的 main()函数调用，试给出每次的输出。

```

#include <iostream.h>
// 函数 function()的适当原型说明
int main()
{
    int x = 5;
    int y = 18;
    // 对于 32、33、35 使用以下调用
    // function(x, y);
    // 对于 34、36 使用以下调用
    // function(x, (float *)(&y));

```

---

```
    cout << "The x is " << x << ", the y is " << y << "\n";  
    return 0;  
}
```

```
32 void function(int a, int b)  
{  
    a = a + 1;  
    b = b - 2;  
    cout << "The a is " << a << ", the b is " << b << "\n";  
    return;  
}
```

难易程度     A    较易

认知程度            理解

预计答题时间     1    分钟

参考答案

```
33 void function(int a, int &b)  
{  
    a = a + 1;  
    b = b - 2;  
    cout << "The a is " << a << ", the b is " << b << "\n";  
    return;  
}
```

难易程度     B    中等难度

认知程度            理解

预计答题时间     2    分钟

参考答案

```
34 void function(int a, int *b)  
{  
    a = a + 1;  
    *b = *b - 2;  
    cout << "The a is " << a << ", the b is " << *b << "\n";  
    return;  
}
```

难易程度     B    中等难度

认知程度            理解

预计答题时间     1    分钟

参考答案

```
35 void function(int &a, float &b)  
{  
    a = a + 1;
```

```

        b = b - 2;
        cout << "The a is " << a << ", the b is " << b << "\n";
        return;
    }

```

难易程度     B   中等难度

认知程度             理解

预计答题时间     3     分钟

参考答案

```

36 void function(int &a, float *b)
{
    a = a + 1;
    b = b - 2;
    cout << "The a is " << a << ", the b is " << *b << "\n";
    return;
}

```

难易程度     B   中等难度

认知程度             理解

预计答题时间     5     分钟

参考答案

37 给出下面程序的输出。

```

#include <iostream.h>

int main()
{
    const int max_number = 10;
    int index;
    int find_one, find_two;
    int array[max_number] = {20, 12, 32, 13, 19, 41, 10, 61, 23, 57 };

    find_one = array[1];
    find_two = array[2];
    for (index = 0; index < max_number; index = index + 1) {
        if (array[index] < find_one) find_one = array[index];
        if (array[index] > find_two) find_two = array[index];
    }
    cout << "\nOk, find_one = " << find_one << ", find_two = " << find_two;
    return 1;
}

```

难易程度     B   中等难度

认知程度             简单应用

预计答题时间     5     分钟

参考答案     (注 该程序实际上找数组中的最小元素和最大元素)

---

Ok, find\_one = 10, find\_two = 61

38 给出下面程序的输出。

```
#include <iostream.h>
int main()
{
    const int max_number = 10;
    int index;
    int result;
    int array[max_number];

    for (index = 0; index < max_number; index = index + 1) array[index] = index+1;
    result = 0;
    for (index = 0; index < max_number; index = index + 1)
        result = result + *(array + (2 * index % max_number));
    cout << "\nOk, At first, result = " << result;

    result = 0;
    for (index = 0; index < max_number; index = index + 1)
        result = result + *array + (2 * index % max_number);
    cout << "\nAnd then, result = " << result;

    return 1;
}
```

难易程度     B   中等难度

认知程度         简单应用

预计答题时间     8   分钟

参考答案        ( 注   该程序实际求数组中的各元素和 )

Ok, At first, result = 50

And then, result = 50

39 给出下面程序的输出。

```
#include <iostream.h>
int main()
{
    const int max_number = 10;
    const int dest_number = 17;
    int up, down, index;
    int array[max_number];

    for (index = 0; index < max_number; index = index + 1) array[index] = index+10;

    down = 0; up = max_number;
    while (down <= up) {
```

参考答案 (注 该程序实际是在有序数组中折半查找指定元素)

I check `array[7] = 17`, and it is `dest_number(=17)`, I find it!!

```
#include <iostream.h>
```

$$\};$$

```
LINK *new_node;
```

```
new_node->element = number;  new_node->next = NULL;
```

```
move = _____;
```

---

```

    }
    // 在上面循环中，我们没有比较过 number 与 head 中存放的整数，下面进行
    // 比较以决定新节点是插在 head 前面还是后面。如果插在 head 前面，那么新
    // 节点应该成为链表的头，因此需要改变 head。
    if (move == head && head->element > number) {
        _____; head = new_node;
    } else {
        // 否则插在 move 的后面，这时 move 的下一节点应该成为 new_node 的下一
        // 节点。
        new_node->next = move->next;
        _____;
    }
    // 注意 必须返回新的链表头，否则这里对 head 的改变不会影响到上层程序，从
    // 这里可看出以指针作为参数仍然是按值调用传递方式。在本函数中对 head 所指
    // 链表中节点内容的修改将影响上层程序，但对 head 本身的修改不会影响上层程序。
    return head;
}

```

难易程度      D    难度较大

认知程度            综合应用

预计答题时间      10    分钟

参考答案

```

    move->next != NULL
    move->next
    new_node->next = head
    move->next = new_node

```

41 打印链表中的整数，也即遍历整个链表。

```

int print(LINK *head)
{
    LINK *move;
    cout << "\n";
    _____;
    while (move != NULL) {
        cout << move->element << ' ';
        move = _____;
    }
    return 1;
}

```

难易程度      B    中等难度

认知程度            简单应用

预计答题时间      5    分钟

参考答案

```

    move = head
    move->next

```



---

42 释放链表所占用的空间，实际上也是遍历整个链表。

```
int free_link(LINK *head)
{
    LINK *move;

    _____;
    while (move != NULL) {
        head = _____;
        delete move;
        move = head;
    }
    return 1;
}
```

难易程度      B    中等难度

认知程度          简单应用

预计答题时间      5    分钟

参考答案

```
    move = head
    move->next
```

// 以下是主程序以演示上面函数。

```
int main()
{
    const int max_number = 10;
    int i;
    int array[max_number] = {20, 12, 32, 13, 19, 41, 10, 61, 23, 57 };

    LINK *head;

    head = new LINK;
    head->element = array[0];  head->next = NULL;
    for (i = 1; i < max_number; i = i + 1) head = insert(head, array[i]);
    print(head);

    return 1;
}
```

程序填空题

43 请填写下面程序中的空白，使程序能输出所给出的结果。

```
#include <iostream.h>
void bubble(int data[], int length)
{
```

```

    int segment;
    int loop;
    int temp;

    for (segment = 0; _____; segment++) {
        for ( _____; loop >= segment; loop = loop - 1) {
            if ( _____) {
                temp = data[loop];
                _____;
                data[loop+1] = temp;
            }
        }
    }
    return;
}

int main()
{
    const int max_nums = 8;
    int loop;
    int data[max_nums] = { 10, 3, 8, 12, 5, 8, 9, 2 };

    bubble(data, max_nums);
    cout << "After sorting:\n";
    for (loop = 0; loop <= max_nums-1; loop++) cout << data[loop] << ' ';
    return 0;
}

```

要求以上程序的输出结果如下

After sorting:

2 3 5 8 8 9 10 12

难易程度      C    较难

认知程度          简单应用

预计答题时间      8    分钟

参考答案

```

    segment <= length - 2
    loop = length - 2
    data[loop+1] < data[loop]
    data[loop] = data[loop+1]

```

44 在以下程序中，函数

```
fill(int square[MAX_LEN][MAX_LEN], int max_num)
```

将数字 1, 2, ..., max\_num, ..., max\_num<sup>2</sup>按逆时针填入max\_num × max\_num的矩阵中，下面是max\_num = 5 和max\_num = 6 时主程序的输出结果

Output(max\_num = 5):

```
1 16 15 14 13
2 17 24 23 12
3 18 25 22 11
4 19 20 21 10
5 6 7 8 9
```

Output(max\_num = 6):

```
1 20 19 18 17 16
2 21 32 31 30 15
3 22 33 36 29 14
4 23 34 35 28 13
5 24 25 26 27 12
6 7 8 9 10 11
```

请完成该函数中的空白，使程序运行能得到正确的结果。

```
#include <iostream.h>
#include <iomanip.h>

const int MAX_LEN = 20;
void full(int square[MAX_LEN][MAX_LEN], int max_num)
{
    int curr_num;
    int curr_level;
    int row, col;

    curr_num = 1; curr_level = 1;
    do {
        row = curr_level; col = curr_level;
        do {
            square[row][col] = curr_num; row++; curr_num++;
        } while (_____);
        do {
            square[row][col] = curr_num; col++; curr_num++;
        } while (_____);
        do {
            square[row][col] = curr_num; row--; curr_num++;
        } while (_____);
        do {
            square[row][col] = curr_num; col--; curr_num++;
        } while (_____);
        curr_level++;
    } while (curr_level <= (max_num / 2));
    if ((max_num % 2) != 0) square[max_num/2+1][max_num/2+1] = curr_num;
    return;
}

void print(int square[MAX_LEN][MAX_LEN], int max_num)
{
    int row, col;

    for (row = 1; row <= max_num; row++) {
```

```

        for (col = 1; col <= max_num; col++)
            cout << setw(4) << square[row][col];
        cout << "\n";
    }
    cout << "\n";
    return;
}

int main()
{
    int square[MAX_LEN][MAX_LEN];

    full(square, 6);
    print(square, 6);
    return 0;
}

```

难易程度 C 较难

认知程度 简单应用

预计答题时间 12 分钟

参考答案

```

row <= max_num - curr_level
col <= max_num - curr_level
row >= curr_level + 1
col >= curr_level + 1

```

45 下面函数返回一个数 number 在 radius 的反演数，例如十进制数 4567 的反演数为 7654，数 4567 在二进制表示下为 100011101011，其反演数为 1110101110001，即十进制的 7537，请填写该函数中的空。

```

unsigned int reserve(unsigned int number, unsigned int radius)
{
    unsigned int reserve_num;
    unsigned int left_num;
    unsigned int max_power, left_power, curr_power;

    max_power = 1;
    while ((number / max_power) >= radius) {
        max_power = _____;
    }

    left_power = max_power;
    curr_power = 1;
    reserve_num = _____;
    while (left_power > 0) {
        left_num = _____;
    }
}

```

```

        reserve_num = _____;

        left_power = left_power / radius;
        curr_power = curr_power * radius;
    }
    return reserve_num;
}

```

难易程度     C   较难

认知程度         简单应用

预计答题时间     12   分钟

参考答案

```

max_power * radius
0
(number / left_power) % radius
reserve_num + left_num * curr_power

```

46 下面函数将数 number 在 radius 进制下的表示转换成字符串，结果放在 result 中。请填写该函数中的空。

```

char *num2str(unsigned int number, unsigned int radius, char *result)
{
    unsigned int left_num, left_char, num_idx;
    unsigned int max_power, left_power;

    max_power = 1;
    while ((number / max_power) >= radius) {
        max_power = _____;
    }

    left_power = max_power;
    num_idx = 0;
    while (left_power > 0) {
        left_num = _____;

        if (left_num <= 9) left_char = char('0' + left_num);
        else left_char = char('A' + left_num - 10);
        result[num_idx] = _____;
        num_idx = _____;

        left_power = left_power / radius;
    }

    result[num_idx] = '\0';
    return result;
}

```

```
}
```

难易程度 C 较难

认知程度 简单应用

预计答题时间 12 分钟

参考答案

```
max_power * radius
(number / left_power) % radius
left_char
num_idx + 1
```

47-49 下面程序用来找任意位（由 bit\_num 决定，需小于 10）的水仙花数，其设计思想是用数组 power 存放预先计算出的从 0 到 9 的 bit\_num 次方，即 power[6] 等于 6 的 bit\_num 次方，用 tens 存放 10 的幂，即 tens[i] 是 10 的 i 次方。函数 amstrong 用来找水仙花数，其中 number 是整数数组，它分别记住 bit\_num 位数的每一位。主程序示范了求 6 位数的水仙花数，得出的结果为 548834。请完成以下各题中的空。

```
#include <iostream.h>
```

```
const int radius = 10;
```

```
const int MAX_BIT = 10;
```

47 下面函数用来求各个数的 bit\_num 次方，存放在数组 power 中。

```
void num_power(unsigned long power[radius], int bit_num)
```

```
{
```

```
    int i, j;
```

```
    for (i = 0; i < radius; i++) power[i] = _____;
```

```
    for (j = 1; j < bit_num; j++) {
```

```
        for (i = 1; i < radius; i++) power[i] = _____;
```

```
    }
```

```
    return;
```

```
}
```

难易程度 A 较易

认知程度 简单应用

预计答题时间 2 分钟

参考答案

```
i
power[i] * i
```

48 下面函数用来求 10 的各次幂

```
void ten_power(unsigned long tens[MAX_BIT], int bit_num)
```

```
{
```

```
    int i;
```

---

```

    tens[0] = 1;
    for (i = 1; i < _____; i++) tens[i] = _____;
    return;
}

```

难易程度      A    中等难度  
 认知程度            简单应用  
 预计答题时间      5    分钟  
 参考答案

```

    bit_num
    tens[i-1] * 10

```

#### 49 下面函数用来找水仙花数

```

void amstrong(unsigned long power[radius], unsigned long tens[MAX_BIT], int bit_num)
{
    int *number;
    int i;
    int finish;
    unsigned long sum, curr_num;

    // 初始化 number 数组。使其各个分量为 bit_num 位数的最小数。
    number = new int[bit_num];
    for (i = 0; i < bit_num; i++) number[i] = 0;
    number[bit_num-1] = _____;

    do {
        // 计算 number 中各个元素的 bit_num 次方的和，注意利用数组 power。
        sum = 0;
        for (i = 0; i < bit_num; i++) sum += _____;

        // 计算 number 数组所代表的数，注意利用数组 tens。
        curr_num = 0;
        for (i = 0; i < bit_num; i++) curr_num += _____;

        if (_____) cout << curr_num << " is an amstrong number.\n";

        // 判断是否到了 bit_num 位数的最大数。
        i = 0;    number[i]++;    finish = 0;
        while (number[i] > radius-1) {
            number[i] = 0;    i++;
            if (i >= bit_num) { finish = 1;    break;    }
            number[i]++;
        }
    } while (finish == 0);
}

```

```

        return;
    }
    难易程度    C    较难
    认知程度    简单应用
    预计答题时间    12    分钟
    参考答案

```

```

        1
        power[number[i]]
        tens[i] * number[i]
        sum == curr_num

```

```

int main()
{
    unsigned long power[radius];
    unsigned long tens[MAX_BIT];
    int bit_num = 6;

    cout << "\nProgram begin...";
    num_power(power, bit_num);
    ten_power(tens, bit_num);
    amstrong(power, tens, bit_num);

    return 0;
}

```

50 下面程序用来生成  $n$  阶的魔方阵 ( 这里的  $n$  为奇数 )。所谓  $n$  阶魔方阵是指这样一种方阵，它的每一行和每一列以及两个对角线上的  $n$  个自然数的和都相等，如下面就是一个 3 阶的魔方阵

8	1	6
3	5	7
4	9	2

试填写下面程序中的空格，使其能运行得到正确结果。

```

#include <iostream.h>
#include <iomanip.h>

const int MAX_LEN = 17+1;
static int fill_magic(int magic[][MAX_LEN], int magic_len);
static int print_magic(int magic[][MAX_LEN], int magic_len);

int main()
{
    int magic[MAX_LEN][MAX_LEN];
    int magic_len;

```



```

do {
    cout << "\nPlease input an odd possitive integer";
    cout << "(<= " << (MAX_LEN-1) << ", such as 7):";
    cin >> magic_len;
} while (magic_len <= 0 || magic_len > MAX_LEN || magic_len % 2 == 0);

fill_magic(magic, magic_len);
print_magic(magic, magic_len);
return 0;
}

int fill_magic(int magic[][MAX_LEN], int magic_len)
{
    int i, j, row, col, count;

    count = 1;
    row = 1;
    col = _____;
    for (i = 1; i <= magic_len; i++) {
        for (j = 1; j <= magic_len; j++) {
            magic[row][col] = _____ ;
            count++;
            if (--row < 1) row = _____;
            if (++col > magic_len) col = _____;
        }
        row += 2;
        if (row > magic_len) row -= magic_len;
        col--;
        if (col < 1) col += magic_len;
    }
    return 1;
}

int print_magic(int magic[][MAX_LEN], int magic_len)
{
    int row, col, count;

    cout << "\nThe " << magic_len << " * ";
    cout << magic_len << " magic matrix:\n";

    for (count = 1; count <= 4 * magic_len + 1; count++) cout << "-";
    for (row = 1; row <= magic_len; row++) {
        cout << "\n";
        for (col = 1; col <= magic_len; col++) {
            cout << setw(3) << magic[row][col];
        }
        cout << "\n";
    }
    for (count = 1; count <= 4 * magic_len + 1; count++) cout << "-";
    return 1;
}

```

难易程度     C   较难

认知程度     综合应用

预计答题时间     10   分钟

参考答案

$(\text{magic\_len} + 1) / 2$

---

```
count
row += magic_len
col -= magic_len
```

51-52 下面程序解决八皇后问题，给出该问题的所有解（92 个解，有些解是非本质的）。程序中定义了类 QUEEN 来求解该问题。该类中提供了求第一个解和求以后所有解的函数。从而上层程序可使用循环来得到所有解。求第一个解和求以后所有解的算法在本质上是一样的，只是回溯的起点不同而已。类 QUEEN 的成员函数 print\_solution() 用来打印一个解。试根据程序中给出的注释填写其中的空。

```
#include <iostream.h>

const int max_length = 16;
class QUEEN {
public:
    QUEEN(int length = 8);

    int get_first_solution();
    int get_next_solution();
    int print_solution();

private:
    int can_place(int row, int col);

    enum BOARD { EMPTY, HAVE_QUEEN };
    BOARD chess_board[max_length+1][max_length+1];
    int solution[max_length+1];
    int length;
};

QUEEN::QUEEN(int length)
{
    int row, col;

    for (row = 1; row <= length; row = row + 1)
        for (col = 1; col <= length; col = col + 1)
            chess_board[row][col] = EMPTY;

    for (row = 1; row <= length; row = row + 1) solution[row] = 0;
    this->length = length;
}
```

51 下面两个函数用来求所有解。

```
int QUEEN::get_first_solution()
```

---

```

{
    int row, col;

    row = 1; col = 1;
    while (row <= length) {
        // 判断在 row 行中一个可放皇后的位置。
        while (can_place(row, col) == 0) col = col + 1;
        if (col > length) {
            // 在 row 行中已经没有可放皇后的位置，这时需要回溯，重新在 row-1 行
            // 放皇后，这时必须清楚上次在 row-1 行中所放的皇后，然后从上次的列的
            // 的下一列重新试放皇后。
            row = _____;
            if (row == 0) return 0;          // 不能回溯，那么该问题已没有解。
            _____ = EMPTY;
            col = solution[row] + 1;
        } else {
            // 第 row 行已经找到合适位置放皇后，将该位置记录在数组 solution 中，
            // 然后推进到第 row+1 行，从该行的第一列开始试，这样一直到最后一行。
            chess_board[row][col] = HAVE_QUEEN;
            solution[row] = _____;
            row = row + 1; col = 1;
        }
    }

    return 1;
}

int QUEEN::get_next_solution()
{
    int row, col;

    // 在求下一个解时，应该从上一解的最后放置的皇后处的下一列开始重新求解。
    row = length;
    _____ = EMPTY;
    col = solution[row] + 1;

    while (row <= length) {
        while (can_place(row, col) == 0) col = col + 1;
        if (col > length) {
            row = _____;
            if (row == 0) return 0;
            _____ = EMPTY;
            col = solution[row]+1;
        } else {

```

---

```

        chess_board[row][col] = HAVE_QUEEN;
        solution[row] = _____;
        row = row + 1;  col = 1;
    }
}

return 1;
}

```

难易程度      C    较难  
 认知程度            综合应用  
 预计答题时间      10   分钟  
 参考答案

```

        row - 1
        chess_board[row][solution[row]]
        col
        chess_board[row][solution[row]]

```

```

int QUEEN::print_solution()
{
    int row, col;

    cout << "\n+";
    for (col = 1; col <= length; col = col + 1) cout << "----+";

    for (row = 1; row <= length; row = row + 1) {
        cout << "\n|";
        for (col = 1; col <= length; col = col + 1) {
            if (chess_board[row][col] == HAVE_QUEEN) cout << " Q |";
            else cout << "    |";
        }
        cout << "\n+";
        for (col = 1; col <= length; col = col + 1) cout << "----+";
    }
    cout << "\n";
    return 1;
}

```

52 下面函数用来当从第一行到第 row-1 行已经正确放置皇后后，判断在第 row 行第 col 列是否可放皇后，如果可以放返回 1，否则返回 0。

```

int QUEEN::can_place(int row, int col)
{
    int temp_row, temp_col;

    if (row == 1) return 1;

```

---

```

// 判断在同一列上是否已经放置了皇后
for (temp_row = _____; temp_row > 0; temp_row = temp_row - 1) {
    if (_____) return 0;
}

// 判断在左对角线上是否已经放置了皇后
for (temp_row = _____, temp_col = _____;
     temp_row > 0 && temp_col > 0;
     temp_row = temp_row - 1, temp_col = temp_col - 1) {
    if (_____) return 0;
}

// 判断在右对角线上是否已经放置了皇后
for (temp_row = _____, temp_col = col + 1;
     temp_row > 0 && temp_col <= length;
     temp_row = temp_row - 1, temp_col = temp_col + 1) {
    if (_____) return 0;
}

return 1;
}

```

难易程度     B   中等难度

认知程度         综合应用

预计答题时间     5     分钟

参考答案

```

row - 1
chess_board[temp_row][col] == HAVE_QUEEN
col - 1
chess_board[temp_row][temp_col] == HAVE_QUEEN

```

// 下面主程序示范怎样利用以上类来得到八皇后问题的所有解。

```

int main()
{
    QUEEN queen;
    int count;

    queen.get_first_solution();
    count = 1;
    cout << "\nSolution " << count << ": ";
    queen.print_solution();
    while (queen.get_next_solution() != 0) {
        count = count + 1;
        cout << "\nSolution " << count << ": ";
        queen.print_solution();
    }
}

```

---

```
    return 1;  
}
```

#### 问答题

53-54 请按下列要求分别定义数组类型

53 数组 budget，一年中各个月对应数组的一个分量，其值为每月的支出金额。

54 描述二十世纪每年人口数的数组 population。

55 数组 mark\_report 其分量对应课程 Algebra, English, Histor 和 Math 其值为成绩等级 (A、B、C、D 和 E)。

56-58 请按下列要求分别定义结构类型

56 房产管理中的结构类型房屋 house，包括房屋名称、房间总数、占地总面积、房屋总造价、每平方米售价。

57 仓库管理中的结构类型库存商品 good，包括商品编码、商品名称、商品规格型号、当前库存数量、当前库存成本价格、本期入库数量、本期出库数量。

58 图书管理中的结构类型图书 book，包括标准书号、书名、作者、出版社、出版时间、价格、检索分类码、检索关键字。

#### 程序编写题

59 编写一个函数 get\_scores()接受用户输入的语文、数学、物理、化学和英语五科成绩，在 main()中利用 get\_scores()接受输入 然后计算总成绩与平均成绩。要求 main()和 get\_scores()之间不得使用全局变量通信。

60 编写矩阵相乘程序。

---

## 第七章

### 填空、选择题

1 C++语言引入继承机制从父类派生出子类，这样子类可看成父类的 A 或 B。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 扩充；B 特殊化

2 作为类的构造机制，继承通过扩充、组合现有的类来构造新的类。扩充是指形成现有类的特例 A，组合是指抽取出若干现有类的共性形成新的抽象层次 B。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 派生类；B 基类

3 作为类型构造机制，如果类DERIVE继承类BASE，那么基类型为BASE的指针\_\_\_\_\_指向基类型为DERIVE的对象，而基类型为DERIVE的指针\_\_\_\_\_指向基类型为BASE的指针。

A 一定能 B 一定不能 C 不一定能 D 应该能

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 D； C

4 在面向对象程序设计中，类型赋值兼容性规则允许将 A 的对象赋给 B 。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 后代类；B 祖先类

5 在C++语言中允许\_\_\_\_\_可兼容基类。

A 公有派生类 B 私有派生类 C 受保护派生类

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A

6 创建一个派生类的对象时，如果基类带有构造函数则先调用 A，后调用 B。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 基类的构造函数；B 派生类的构造函数

---

7 使用\_\_A\_\_能将私有继承下来的父类的公有成员恢复到在子类中也具有公有访问控制方式。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 访问声明

8 如果在派生类中定义了一个函数原型与从父类继承的成员函数完全相同的成员函数,那么该函数的实现是对父类同名函数的\_\_A\_\_,这时如果在子类中要访问父类同名函数,那么需使用\_\_B\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 重定义; B 成员名限定符访问

9 在C++语言中,如果两个基类具有相同名字的成员,那么在派生类中这个名字将会产生\_\_A\_\_,解决该方法一是使用成员名限定符,二是\_\_B\_\_有名字冲突的成员。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 二义性; B 重定义

10 在多重继承情况下,多个基类构造函数的调用次序是按\_\_A\_\_调用的,与它们在派生类构造函数实现中的初始化列表出现的次序无关。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 基类在被继承时所声明的次序,从左至右依次

11 OOP语言如果允许多重继承,就必须考虑重复继承问题。C++语言中引入\_\_A\_\_解决派生类中数据成员的二义性问题,使得派生类只保留了被重复继承祖先类的唯一副本。C++语言中另一种解决重复继承二义性的方法是采用\_\_B\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 虚基类; B 成员名限定

12 在C++语言中,对虚基类构造函数的调用\_\_\_\_\_普通基类的构造函数。

A 总是先于 B 总是后于

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A

13 一个派生类对象实例的初始化时依次调用\_\_\_\_、\_\_\_\_、\_\_\_\_、\_\_\_\_。



- A 自身的构造函数                      B 虚基类的构造函数  
C 普通基类的构造函数                  D 对象成员的构造函数
- 难易程度      A 较易  
认知程度          识记  
预计答题时间      0.5 分钟  
参考答案          B ;      C ;      D ;      A

14 一个派生类具有多个基类时 ,该派生类的实例在初始化时按\_\_\_\_\_调用基类的构造函数。

- A 声明这些基类的次序                      B 在派生类构造函数初始化列表出现的次序
- 难易程度      A 较易  
认知程度          识记  
预计答题时间      0.5 分钟  
参考答案          A

15 一个类具有多个成员对象时 ,该派生类的实例在初始化时按\_\_\_\_\_调用成员对象的构造函数。

- A 声明这些成员对象的次序                      B 在该类构造函数初始化列表出现的次序
- 难易程度      A 较易  
认知程度          识记  
预计答题时间      0.5 分钟  
参考答案          A

#### 问答题

- 16 请分别说明类 A 的派生类和非派生类对 A 中成员的访问控制规则。  
17 请说明在不同的继承访问控制下，继承成员的访问控制规则。  
18 请说明在 C++语言中，在派生类中对继承成员可作哪些调整，其作用何在？  
19 试说明继承机制与提高软件可重用性之间的关系。

#### 程序改错题

20 指出下面程序中的错误

```
#include <iostream.h>
#include <string.h>

class BASE {
public:
    int set(char *my_name)
    {
        name = new char[strlen(my_name) + 1];
        strcpy(name, my_name);
        return 1;
    }
}
```

```

private:
    char *name;
};

class DERIVE: BASE {
};

int main()
{
    BASE base;
    DERIVE derive;

    base = derive;
    base.set("Computer Department");
    derive.set("Zhongshan University");
    return 1;
}

```

难易程度     B   中等难度

认知程度         理解

预计答题时间     2   分钟

参考答案     由于是私有继承，在主函数中不能将 derive 赋给 base，也不能通过对象 derive 调用 base 类的成员函数 set()

## 21 指出下面程序中的错误

```

#include <iostream.h>
#include <string.h>

class BASE {
public:
    BASE(char *my_base)
    {
        name = new char[strlen(my_base)+1];
        strcpy(name, my_base);
        return 1;
    }
private:
    char *name;
};

class DERIVE: public BASE {
public:
    int DERIVE(char *my_derive)
    {
        name = new char[strlen(my_derive)+1];
    }
}

```

```
        strcpy(name, my_derive);
    }
private:
    char *name;
};
```

难易程度     B   中等难度

认知程度         理解

预计答题时间     2   分钟

参考答案         在 derive 类的构造函数中要使用初始化列表显式调用 base 的构造函数。

## 22 指出下面程序中的错误

```
#include <iostream.h>
#include <string.h>

class BASE {
public:
    int set(char *my_base)
    {
        name = new char[strlen(my_base)+1];
        strcpy(name, my_base);
        return 1;
    }
private:
    char *name;
};

class DERIVE: public BASE {
public:
    int set(char *my_derive, char *my_base)
    {
        name = new char[strlen(my_derive)+1];
        strcpy(name, my_derive);

        BASE::name = new char[strlen(my_base)+1];
        strcpy(BASE::name, my_base);
    }
private:
    char *name;
};

int main()
{
    DERIVE derive;
    BASE base;
```

```

    derive.set("Computer Department");
    derive.set("Computer Department", "Zhongshan University");

    base = derive;
    base.set("Computer Department");
    base.set("Computer Department", "Zhongshan University");

    return 1;
}

```

难易程度     B   中等难度

认知程度         理解

预计答题时间     2   分钟

参考答案

由于 name 是 BASE 的私有数据，因此在 DERIVE 中不能访问。

不能通过 derive 调用类 BASE 的成员函数 set(char \*)，因为已经在类 DERIVE 中重定义成 set(char\*, char\*)。

不能通过 base 调用类 DERIVE 的成员函数 set(char \*, char \*)。

## 23 指出下面程序中的错误

```

#include <iostream.h>
#include <string.h>

class BASE_ONE {
public:
    int set(char *my_base)
    {
        name = new char[strlen(my_base)+1];
        strcpy(name, my_base);
        return 1;
    }
private:
    char *name;
};

class BASE_TWO {
public:
    int set(char *my_base)
    {
        name = new char[strlen(my_base)+1];
        strcpy(name, my_base);
        return 1;
    }
private:

```

```

    char *name;
};

class DERIVE: public BASE_ONE, public BASE_TWO {
};

int main()
{
    DERIVE derive;
    derive.set("Computer Department");
    return 1;
}

```

难易程度     B   中等难度

认知程度         理解

预计答题时间     2   分钟

参考答案         在主函数中通过 derive 调用成员函数 set 时发生二义性。

24 指出下面程序中的错误。

```

#include <iostream.h>
#include <string.h>

class BASE_ONE {
public:
    int set(char *my_base)
    {
        name = new char[strlen(my_base)+1];
        strcpy(name, my_base);
        return 1;
    }
private:
    char *name;
};

class BASE_TWO {
public:
    int set(char *my_base)
    {
        name = new char[strlen(my_base)+1];
        strcpy(name, my_base);
        return 1;
    }
private:
    char *name;
};

```

```

class DERIVE: virtual public BASE_ONE, virtual public BASE_TWO {
};

int main()
{
    DERIVE derive;
    derive.set("Computer Department");
    return 1;
}

```

难易程度     B   中等难度

认知程度         理解

预计答题时间     2   分钟

参考答案        在主函数中通过 derive 调用成员函数 set 时发生二义性。

## 25 指出下面程序中的错误

```

#include <iostream.h>
#include <string.h>
class PERSON {
public:
    PERSON(char *name)
    { int len;
      len = strlen(name);
      PERSON::name = new char[len+1];
      strcpy(PERSON::name, name);
    }
    ~PERSON() { delete name; }
    char* get_name() { return name; }
protected:
    char *name;
};

class STUDENT: PERSON {
public:
    void STUDENT(char *st_name, int score): score(score)
    { int len;
      len = strlen(st_name);
      name = new char[len+1];
      strcpy(name, st_name);
    }
    int get_score() { return score; }
protected:
    int score = 100;
};

int main()

```

```

{
    STUDENT soft("Zhao", 80);

    cout << "Student: " << soft.get_name();
    cout << "Score: " << soft.score << ".\n";
    return 0;
}

```

难易程度     C   较难

认知程度         简单应用

预计答题时间     6   分钟

参考答案

STUDENT 的构造函数不可有返回类型，并应提供对基类 PERSON 构造函数的调用。

STUDENT 中不能对数据成员 score 进行初始化。

在 main() 中 soft 的成员函数 get\_name() 和数据成员 score 不可访问。

## 26 指出下面程序中的错误

```

#include <iostream.h>
#include <string.h>

// Exercise 7-4 (a)
const int MAX_NAME_LEN = 30;

enum STATUS {
    IDLE = 0x0000,
    READY = 0x0001,
    BUSY = 0x0002,
    ERROR = 0x8000
};

class DEVICE {
public:
    DEVICE(STATUS status)
    {
        this->status = status;
    }
    void set_status(STATUS status)
    {
        this->status = status;
        return;
    }
private:
    STATUS status;
}

```

```

};

class PRINTER: public DEVICE {
public:
    int display_status()
    {
        cout << "The status of printer: " << status;
        return 1;
    }
};

class KEYBOARD: DEVICE {
public:
    KEYBOARD() : DEVICE(READY) { };

    int display_status()
    {
        cout << "The status of Keyboard: " << status;
    }
};

int main()
{
    PRINTER LQ_1600K;
    KEYBOARD keyboard_101;

    LQ_1600K.display_status();
    keyboard_101.set_status(ERROR);
    keyboard_101.display_status();

    return 0;
}

```

难易程度      C    较难

认知程度          简单应用

预计答题时间      6    分钟

参考答案

在类 PRINTER 和类 KEYBOARD 中不能访问类 DEVICE 的私有数据 status  
 在类 PRINTER 的构造函数中应提供对基类 DEVICE 构造函数的调用。  
 在 main()中不能通过对象 keyboard\_101 访问 DEVICE 的成员函数 set\_status()。

程序输出结果题

27 给出下面程序的输出结果

```
#include <iostream.h>
```



```

#include <string.h>

class BASE {
public:
    BASE()
    {
        cout << "\nConstruct BASE!";
    }
    ~BASE()
    {
        cout << "\nDestruct BASE!";
    }
};

class DERIVE: public BASE {
public:
    DERIVE()
    {
        cout << "\nConstruct DERIVE!";
    }
    ~DERIVE()
    {
        cout << "\nDestruct DERIVE!";
    }
};

int main()
{
    BASE base;
    DERIVE derive;
    return 1;
}

```

难易程度    B    中等难度

认知程度        理解

预计答题时间    3    分钟

参考答案

Construct BASE!

Construct BASE!

Construct DERIVE!

Destruct DERIVE!

Destruct BASE!

Destruct BASE!

28 给出下面程序的输出结果

```

#include <iostream.h>

```

```

#include <string.h>

class BASE {
public:
    BASE()
    {
        cout << "\nConstruct BASE!";
    }
    ~BASE()
    {
        cout << "\nDestruct BASE!";
    }
};

class DERIVE: public BASE {
public:
    DERIVE()
    {
        cout << "\nConstruct DERIVE!";
    }
    ~DERIVE()
    {
        cout << "\nDestruct DERIVE!";
    }
private:
    BASE derive_base;
};

int main()
{
    BASE base;
    DERIVE derive;
    return 1;
}

```

难易程度    B    中等难度

认知程度        理解

预计答题时间    5    分钟

参考答案

Construct BASE!

Construct BASE!

Construct BASE!

Construct DERIVE!

Destruct DERIVE!

Destruct BASE!

---

Destruct BASE!

Destruct BASE!

29 请给出下面程序的输出

```
#include <iostream.h>
#include <string.h>

class BASE_ONE {
public:
    BASE_ONE()
    {
        cout << "\nConstruct BASE_ONE!";
    }
    ~BASE_ONE()
    {
        cout << "\nDestruct BASE_ONE!";
    }
};

class BASE_TWO {
public:
    BASE_TWO()
    {
        cout << "\nConstruct BASE_TWO!";
    }
    ~BASE_TWO()
    {
        cout << "\nDestruct BASE_TWO!";
    }
};

class DERIVE: public BASE_ONE, public BASE_TWO {
public:
    DERIVE()
    {
        cout << "\nConstruct DERIVE!";
    }
    ~DERIVE()
    {
        cout << "\nDestruct DERIVE!";
    }
};

int main()
```

```
{  
    DERIVE derive;  
    return 1;  
}
```

难易程度    B    中等难度

认知程度        理解

预计答题时间    5    分钟

参考答案

Construct BASE\_ONE!

Construct BASE\_TWO!

Construct DERIVE!

Destruct DERIVE!

Destruct BASE\_TWO!

Destruct BASE\_ONE!

30 请给出下面程序的输出

```
#include <iostream.h>  
#include <string.h>  
  
class BASE_ONE {  
public:  
    BASE_ONE()  
    {  
        cout << "\nConstruct BASE_ONE!";  
    }  
    ~BASE_ONE()  
    {  
        cout << "\nDestruct BASE_ONE!";  
    }  
};  
  
class BASE_TWO {  
public:  
    BASE_TWO()  
    {  
        cout << "\nConstruct BASE_TWO!";  
    }  
    ~BASE_TWO()  
    {  
        cout << "\nDestruct BASE_TWO!";  
    }  
};  
  
class DERIVE: public BASE_ONE, virtual public BASE_TWO {
```

```

public:
    DERIVE()
    {
        cout << "\nConstruct DERIVE!";
    }
    ~DERIVE()
    {
        cout << "\nDestruct DERIVE!";
    }
};

int main()
{
    DERIVE derive;
    return 1;
}

```

难易程度     B   中等难度

认知程度            理解

预计答题时间     5   分钟

参考答案

Construct BASE\_TWO!

Construct BASE\_ONE!

Construct DERIVE!

Destruct DERIVE!

Destruct BASE\_ONE!

Destruct BASE\_TWO!

31 请给出下面程序的输出结果

```

#include <iostream.h>
#include <string.h>

class BASE_ONE {
public:
    BASE_ONE()
    {
        cout << "\nConstruct BASE_ONE!";
    }
    ~BASE_ONE()
    {
        cout << "\nDestruct BASE_ONE!";
    }
};

class BASE_TWO {

```

```

public:
    BASE_TWO()
    {
        cout << "\nConstruct BASE_TWO!";
    }
    ~BASE_TWO()
    {
        cout << "\nDestruct BASE_TWO!";
    }
};

class DERIVE: public BASE_ONE, virtual public BASE_TWO {
public:
    DERIVE()
    {
        cout << "\nConstruct DERIVE!";
    }
    ~DERIVE()
    {
        cout << "\nDestruct DERIVE!";
    }
private:
    BASE_TWO derive_base;
};

int main()
{
    DERIVE derive;
    return 1;
}

```

难易程度     B   中等难度

认知程度         理解

预计答题时间     8   分钟

参考答案

Construct BASE\_TWO!

Construct BASE\_ONE!

Construct BASE\_TWO!

Construct DERIVE!

Destruct DERIVE!

Destruct BASE\_TWO!

Destruct BASE\_ONE!

Destruct BASE\_TWO!

32 请给出下面程序的输出

```
#include <iostream.h>
```

```

class WINDOW {
public:
    WINDOW() {    count = count + 1; }
    ~WINDOW() {  count = count - 1;  }
    int get_win() { return count; }
private:
    static int count;
};

int WINDOW::count = 0;

class WORK_WIN: public WINDOW {
public:
    WORK_WIN()
    {
        cout << "Open a work window.\n";
    }
    ~WORK_WIN()
    {
        cout << "Close a work window.\n";
    }
};

class MSG_WIN: public WINDOW {
public:
    MSG_WIN()
    {
        cout << "Open a message window.\n";
    }
    ~MSG_WIN()
    {
        cout << "Close a message window.\n";
    }
};

class SCREEN {
public:
    SCREEN(): msg_win(), work_win()
    {
        cout << "Initialize the screen.\n";
    }
    ~SCREEN()
    {

```

```

        cout << "Clear the screen.\n";
    }
    int get_win()
    {
        return work_win.get_win();
    }
private:
    WORK_WIN work_win;
    MSG_WIN msg_win;
};

int main()
{
    SCREEN screen;
    WINDOW msg_win;

    cout << "There are " << screen.get_win() << " window(s) on screen.\n";
    cout << "There are " << msg_win.get_win() << " messge window(s).\n";
    return 0;
}

```

难易程度     C   较难

认知程度         理解

预计答题时间     12   分钟

参考答案

Open a work window.

Open a message window.

Initialize the screen.

There are 3 window(s) on screen.

There are 3 messge window(s).

Clear the screen.

Close a message window.

Close a work window.



---

## 第八章

1 在面向对象程序设计语言中，由程序员设计的多态性有两种最基本的形式   A   和   B  。

难易程度   A  较易

认知程度       识记

预计答题时间   0.5  分钟

参考答案    A  编译时多态性；B  运行时多态性

2 编译多态性是指在程序编译阶段时即可确定下来的多态性，主要通过使用   A   机制获得，在C++语言中有两大类型的重载   B   和   C   。

难易程度   A  较易

认知程度       识记

预计答题时间   0.5  分钟

参考答案    A  重载；B  函数重载；C  运算符重载

3 运行时多态性是指必须等到程序动态运行时才可确定的多态性，主要通过继承机制结合   A   获得。

难易程度   A  较易

认知程度       识记

预计答题时间   0.5  分钟

参考答案    A  动态绑定

4 虚函数和普通函数的区别在于函数名字与函数体之间的绑定方式不同，普通函数使用的是   A  ，而虚函数使用的是   B  。

难易程度   A  较易

认知程度       识记

预计答题时间   0.5  分钟

参考答案    A  静态绑定；B  动态绑定

5 编译程序根据   A   与   B   选择调用哪一个版本的重载函数。

难易程度   A  较易

认知程度       识记

预计答题时间   0.5  分钟

参考答案    A  实际参数个数；B  相应位置的类型

6 在声明一个对象时如果使用同一类类型的另一个对象来初始化，这时调用的是该类的拷贝构造函数，除此之外，拷贝构造函数还用于   A  、  B  。

难易程度   A  较易

认知程度       识记

预计答题时间   0.5  分钟

参考答案    A  在按值方式调用时，创建临时对象副本传递给函数；B  生成临时对象副本作为函数的返回值。

7 C++中提供的缺省拷贝构造函数采用浅复制策略，当类中包括指针类型的数据成员时，我们需要采用 A 策略。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 深复制

8 在C++中， A 只能用作派生其他类的基类，而本身不能创建对象实例。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 抽象类

9 在C++中，运算符看成一种特殊形式的函数，重载运算符有两种情况  A 和 B 。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 重载类成员运算符；B 重载友元运算符

### 名词解释

说明以下几组概念的区别

10 虚函数与纯虚函数

11 多态类与抽象类

12 静态绑定与动态绑定

13 早期绑定与晚期绑定

14 静态类型与动态类型

### 问答题

15 说明在存在函数重载情况下，编译程序选择函数调用版本的规则。

### 程序改错题

16 根据以下重载函数的定义判断主函数 main()中的函数调用是否合法。

```
#include <iostream.h>

void func(char ch)
{
    return ch + 1;
}

void func(unsigned char ch)
{
    return ch - 1;
}

int main()
{
```

```
cout << func('A');  
cout << func(65);  
return 0;  
}
```

难易程度     B   中等难度

认知程度         理解

预计答题时间     3     分钟

参考答案

17 请指出下面程序中的错误

```
#include <iostream.h>  
#include <string.h>  
  
class BASE {  
public:  
    virtual int set(char *name) = 0;  
private:  
    char *name;  
};  
  
class DERIVE: virtual public BASE {  
};  
  
int main()  
{  
    DERIVE derive;  
    BASE base;  
  
    derive.set("Computer Department");  
    base.set("Zhongshan University");  
    return 1;  
}
```

难易程度     B   中等难度

认知程度         理解

预计答题时间     3     分钟

参考答案     不能为抽象类创建对象实例。

程序输出结果题

18 指出下面程序的输出结果。

```
#include <iostream.h>  
  
void print(int x)  
{  
    cout << "Calling print(int) with " << x << ".\n";  
    return;  
}
```

```

}

void print(int x, int y)
{
    cout << "Calling print(int, int) with " << x << ", " << y << ".\n";
    return;
}

void print(double x)
{
    cout << "Calling print(double) with " << x << ".\n";
    return;
}

int main()
{
    print(800);
    print(3.14159);
    print(3, 4);
    return 0;
}

```

难易程度     B   中等难度

认知程度         简单应用

预计答题时间     5     分钟

参考答案

19 指出下面程序的输出结果。

```

#include <iostream.h>

class F00 {
public:
    F00(int i)
    {
        cout << "Constructing.\n";
        pointer = new int;
        *pointer = i;
        return;
    }
    F00(const F00& other)
    {
        cout << "Copy constructing.\n";
        pointer = new int;
        *pointer = *other.pointer;
        return;
    }
    ~F00()
    {
        cout << "Destructing.\n";
        delete pointer;
        return;
    }
    int get()
    {
        return *pointer;
    }
private:

```

```

        int *pointer;
    };

void display(F00 obj)
{
    cout << obj.get() << "\n";
    return;
}

int main()
{
    F00 obj1(15);
    F00 obj2 = obj1;

    display(obj2);
    return 0;
}

```

难易程度     B   中等难度

认知程度             简单应用

预计答题时间     5     分钟

参考答案

20 指出下面程序的输出结果。

```

#include <iostream.h>

class F00 {
public:
    F00()
    {
        cout << "Constructing.\n";
        return;
    }
    F00(const F00& other)
    {
        cout << "Copy constructing.\n";
        return;
    }
    ~F00()
    {
        cout << "Destructing.\n";
        return;
    }
};

F00 get_object()
{
    F00 obj;
    return obj;
}

int main()
{
    F00 obj;

    obj = get_object();
    return 0;
}

```

---

```
}
```

难易程度     B   中等难度  
认知程度         简单应用  
预计答题时间     5     分钟  
参考答案

21 指出下面程序的输出结果。

```
#include <iostream.h>

class FAIRY_TALE {
public:
    virtual void act1()
    {
        cout << "Princess meets Frog.\n";
        act2();
        return;
    }
    virtual void act2()
    {
        cout << "Princess kisses Frog.\n";
        act3();
        return;
    }
    virtual void act3()
    {
        cout << "Frog turns into Prince.\n";
        act4();
        return;
    }
    virtual void act4()
    {
        cout << "They live happily ever after.\n";
        act5();
        return;
    }
    virtual void act5()
    {
        cout << "- The End -\n";
        return;
    }
};

class UNHAPPY_TALE: public FAIRY_TALE {
public:
    virtual void act3()
    {
        cout << "Frog stays a Frog.\n";
        act4();
        return;
    }
    virtual void act4()
    {
        cout << "Princess runs away in disgust.\n";
        act5();
        return;
    }
};
```

```

virtual void act5()
{
    cout << "- The End of Unhappiness -\n";
    return;
}
};

int main()
{
    char selection;
    FAIRY_TALE *tale;

    cout << "Which tale would you like to hear (f/u)? ";
    cin >> selection;
    if (selection == 'f') tale = new FAIRY_TALE;
    else tale = new UNHAPPY_TALE;
    tale->act1();
    delete tale;
    return 0;
}

```

难易程度 C 较难

认知程度 简单应用

预计答题时间 10 分钟

参考答案

## 22 给出下面程序的输出结果

```

#include <iostream.h>
#include <string.h>

const int CODE_LEN = 20;
class DEPART {
public:
    DEPART(char *dep_code = "Math")
    {
        strncpy(code, dep_code, CODE_LEN); code[CODE_LEN] = '\0';
        cout << "Constructing depart: [" << code << "].\n";
    }
    DEPART(const DEPART& other)
    {
        strncpy(code, other.code, CODE_LEN); code[CODE_LEN] = '\0';
        cout << "Copy constructing depart: [" << code << "].\n";
    }
    ~DEPART()
    {
        cout << "Destructing depart: [" << code << "].\n";
    }
    DEPART operator =(const DEPART& other)
    {
        cout << "Calling operator =, set [" << code;
    }
}

```

```

        cout << "]" equal to [" << other.code << "].\n";
        strncpy(code, other.code, CODE_LEN); code[CODE_LEN] = '\0';
        return *this;
    }
    void print()
    {
        cout << "Depart " << code;
    }
private:
    char code[CODE_LEN+1];
};

class EMPLOYEE {
public:
    EMPLOYEE(char *emp_code = "Teacher",
              char *dep_code = "Computer"): depart(dep_code)
    {
        strncpy(code, emp_code, CODE_LEN); code[CODE_LEN] = '\0';
        cout << "Constructing employee: [" << code << "].\n";
    }
    ~EMPLOYEE()
    {
        cout << "Destructing employee: [" << code << "].\n";
    }
    void set_depart(DEPART depart)
    {
        EMPLOYEE::depart = depart;
    }
    void print()
    {
        cout << "Employee code: [" << code << "];
        cout << ", He belong to ";
        depart.print();
        cout << "\n";
    }
private:
    char code[CODE_LEN+1];
    DEPART depart;
};

int main()
{
    DEPART dep;
    EMPLOYEE emp;

```



```
emp.set_depart(dep);  
emp.print();  
return 0;  
}
```

难易程度 D 难度较大

认知程度 简单应用

预计答题时间 15 分钟

参考答案

Constructing depart: [Math].

Constructing depart: [Computer].

Constructing employee: [Teacher].

Copy constructing depart: [Math].

Calling operator =, set [Computer] equal to [Math].

Copy constructing depart: [Math].

Destructing depart: [Math].

Destructing depart: [Math].

Employee code: [Teacher], He belong to Depart Math

Destructing employee: [Teacher].

Destructing depart: [Math].

Destructing depart: [Math].

23 请给出下面程序的输出

```
#include <iostream.h>  
#include <string.h>  
  
const int MAX_LEN = 20;  
  
class PERSON {  
public:  
    PERSON(char *name = "Zhao", int salary = 300): salary(salary)  
    {  
        strncpy(PERSON::name, name, MAX_LEN);  
        PERSON::name[MAX_LEN] = '\0';  
        cout << "Constructing Person " << name << ".\n";  
    }  
    virtual void show()  
    {  
        cout << "Salary per month for Person " << name;  
        cout << " is " << salary << " yuan.\n";  
    }  
    ~PERSON()  
    {  
        cout << "Destructing Person " << name << ".\n";  
    }  
};
```

```

    }
protected:
    char name[MAX_LEN+1];
    int salary;
};
class EMPLOYEE: public PERSON {
public:
    EMPLOYEE(char *name = "Qian", int salary = 200):
        PERSON(name, salary)
    {
        cout << "Constructing Employee " << name << ".\n";
    }
    virtual void show()
    {
        cout << "Salary per week for Employee " << name;
        cout << " is " << salary << " yuan.\n";
    }
    ~EMPLOYEE()
    {
        cout << "Destructing Employee " << name << ".\n";
    }
};
class TEACHER: public EMPLOYEE {
public:
    TEACHER(char *name = "Sun", int salary = 10):
        EMPLOYEE(name, salary)
    {
        cout << "Constructing Teacher " << name << ".\n";
    }
    virtual void show()
    {
        cout << "Salary per hour for Teacher " << name;
        cout << " is " << salary << " yuan.\n";
    }
    ~TEACHER()
    {
        cout << "Destructing Teacher " << name << ".\n";
    }
};

int main()
{
    PERSON* person_ptr[3];
    PERSON person;

```

```

    EMPLOYEE employee;
    TEACHER teacher;
    int i;

    person_ptr[0] = &person;
    person_ptr[1] = &employee;
    person_ptr[2] = &teacher;
    for (i = 0; i < 3; i++) person_ptr[i]->show();

    return 0;
}

```

难易程度     D    难度较大

认知程度         简单应用

预计答题时间     15    分钟

参考答案

Constructing Person Zhao.

Constructing Person Qian.

Constructing Employee Qian.

Constructing Person Sun.

Constructing Employee Sun.

Constructing Teacher Sun.

Salary per month for Person Zhao is 300 yuan.

Salary per week for Emplpyee Qian is 200 yuan.

Salary per hour for Teacher Sun is 10 yuan.

Destructing Teacher Sun.

Destructing Employee Sun.

Destructing Person Sun.

Destructing Employee Qian.

Destructing Person Qian.

Destructing Person Zhao.

类界面设计题，以下各题只要求写出类界面，不需要实现。

24 设计一个复数类 COMPLEX，尽量合理使用运算符重载。

25 设计一个日期类 DATE，尽量合理使用运算符重载。

26 设计一个有理数类 RATIONAL，尽量合理使用运算符重载。

27 类 INT\_SET 描述了元素为整数的集合。集合中的元素可用链表或数组表示；INT\_SET 为其他类的对象提供了判断是否包括某元素的成员函数 has\_element()、以成员函数的形式重载了二元运算符+和\*，分别表示两个集合的并和交，以及重载了输入输出流库中的插入与提取运算符<<和>>。请写出类 INT\_SET 的类界面。

---

## 第九章 类属机制

### 填空、选择题

1 类属类必须经过  A  后才能成为可创建对象实例的类类型。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 实例化

2 在某些情况下,类属类可能要求实际类属参数必须具有某些特殊性质,这种类属类称为  A 。如果实际类属参数可是任何类型,则是  B 。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 受约束类属类; B 无约束类属类

3 类属类的实例化必须由  A  完成,而类属函数的实例化则是由编译程序根据函数调用的  B  自动完成的。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 程序员显式; B 实际参数类型

4 调用类属函数时,应保证函数的  A  与  B  完全匹配,编译程序在实例化过程中不作任何类型转换。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 实际参数类型; B 形式类属参数

### 问答题

5 什么是类属机制,通常在什么情况下使用?

6 试指出类属函数与重载函数的区别,并说明他们通常分别在什么情况下使用?

7 试说明在同时有类属函数和重载函数时,编译程序调用函数时选择函数版本的规则。

### 程序改错题

8 指出下面程序片段中的错误

```
template <class ITEM, class LINK>
int compare(ITEM source, ITEM target)
{
    if (source > target) return 1;
```

---

```
    else return 0;
}
```

难易程度     A   较易  
认知程度            识记  
预计答题时间     1   分钟  
参考答案     类属参数没有用到。

9-12     设有如下声明 ,判断各题中的函数调用是否合法? 如果合法则指出调用的是哪一个版本的函数。

```
template <class T>
void func(T x, T y)
{
    // ...
}

func(int i, int j)
{
    // ...
}
```

9    func('a', 'b');

难易程度     A   较易  
认知程度            理解  
预计答题时间     1   分钟  
参考答案     调用 func(T x, T y) , T 实例化为 char

10   func(10, 'c');

难易程度     A   较易  
认知程度            理解  
预计答题时间     1   分钟  
参考答案     调用 func(int x, int y) , 'c' 隐式类型转换为整数类型

11   func("string1", "string2");

难易程度     A   较易  
认知程度            理解  
预计答题时间     1   分钟  
参考答案     调用 func(T x, T y) , T 实例化为 char\*

12   func(10, "string");

难易程度     A   较易  
认知程度            理解  
预计答题时间     1   分钟  
参考答案     错误, 既不能实例化 func(T x, T y) 中的 T, 也不能将 char\* 类型转换为 int。

类界面设计题, 以下各题只要求写出类界面, 不需要实现。

---

13 类 SET 为一个类属集合。集合中的元素可用链表或数组表示；INT\_SET 为其他类的对象提供了判断是否包括某元素的成员函数 `has_element()`、以成员函数的形式重载了二元运算符+和\*，分别表示两个集合的并和交，以及重载了输入输出流库中的插入与提取运算符<<和>>。请写出类 SET 的类界面。

## 第十章及以后

## 填空、选择题

1 在C++语言中提供了\_\_A\_\_类来支持输入输出。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 实际参数类型；B 形式类属参数

2 C++语言将文件看作一个个字节的数据顺序组成，按照文件中数据的组织形式，可将文件分为\_\_A\_\_和\_\_B\_\_两类。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 文本文件；B 二进制文件

3 流类库支持格式化输入输出，格式化既可使用类 ios 类的成员函数，也可使用一类特殊的函数\_\_A\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 类操纵符

4 在C++中，文件也可用流来操作，使用文件之前必须\_\_A\_\_，使用完毕后应该\_\_B\_\_，C++既支持对文件的\_\_C\_\_访问，也支持对文件的\_\_D\_\_访问

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 打开文件；B 关闭文件；C 顺序；D 随机

5 已声明一个输入文件流对象 src\_file，执行 src\_file 的打开操作时与磁盘文件 C:\autoexec.bat 相连接，且以文本文件模式打开，则打开语句为\_\_A\_\_，紧接着判断该文件打开操作是否成功的逻辑表达式为\_\_B\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A src\_file.open("C:\\autoexec.bat"); B (!src\_file)

6 可靠性是软件质量的重要因素，它反映在两个方面\_\_A\_\_和\_\_B\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 正确性；B 健壮性

7 程序设计风范指导程序员如何编写好的程序，它可分为\_\_A\_\_、\_\_B\_\_、\_\_C\_\_和\_\_D\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A 过程程序设计；B 模块程序设计；C 类型程序设计；D 面向对象程序设计

8 面向对象程序设计语言的前驱是\_\_A\_\_，而在理论上比较完善的面向对象程序设计语言是\_\_B\_\_，导致面向对象程序设计兴起的语言及程序设计环境是\_\_C\_\_，而在Internet上最有前途的面向对象程序设计语言是\_\_D\_\_。

难易程度 A 较易

认知程度 识记

预计答题时间 0.5 分钟

参考答案 A Simula；B Eiffel；C Smalltalk；D Java

9 请给出下面程序的输出结果

```
#include <iostream.h>
#include <iomanip.h>
#include <string.h>

int main()
{
    const double pi = 3.14;
    char* subject = "Object-Oriented Programming";
    int length = strlen(subject);

    cout.width(20);
    cout << setprecision(2) << hex << "(" << length << ")" << dec << endl;

    cout.fill('.');
    cout.setf(ios::right);
    cout << setw(20) << "subject: " << subject << endl;

    cout.setf(ios::showpos | ios::left);
    cout << "(" << setw(10) << pi << ")" << endl;
    cout << "(" << length << ")" << endl;
    return 0;
}
```

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案



10-14 假设声明并初始化了以下变量：

```
int x = 321;
double y = 1.44;
char* z = "abcd";
```

下面各题要求写出一条或多条语句,利用 ios 类成员函数或输入 / 输出操纵符输出 x、y 和 z,使其满足规定的格式要求,其中“□”表示空格。

10 x 输出格式为“+□□□□321”;

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案

```
cout.setf(ios::internal | ios::showpos);
cout << setw(8) << x << endl;
```

11 x 输出格式为“321...”;

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案

```
cout.setf(ios::left); cout.fill('.');
cout << setw(6) << x << endl;
```

12 y 输出格式为“□□□□□□+1.4”;

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案

```
cout.setf(ios::showpos);
cout << setw(10) << setprecision(1) << y << endl;
```

13 y 输出格式为“1.4400”;

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案

```
cout.setf(ios::left | ios::showpoint);
cout << setprecision(4) << y << endl;
```

14 z 输出格式为“□□□abcd”。

难易程度 B 中等难度

认知程度 理解

预计答题时间 5 分钟

参考答案

```
cout.setf(ios::right);
```

```
cout << setw(7) << z << endl;
```