

一、选择题

1 关于运算符重载，下列表述中正确的是()。

- A. c++已有的任何运算符都可以重载
- B. 运算符函数的返回类型不能声明为基本数据类型
- C. 在类型转换符函数的定义中不需要声明返回类型
- D. 可以通过运算符重载来创建 c++中原来没有的运算符

参考答案：C

参考解析：重载运算符的规则如下：①c++不允许用户自己定义新的运算符，只能对已有的 c++运算符进行重载；②c++不能重载的运算符只有 5 个；③重载不能改变运算符运算对象的个数；④重载不能改变运算符的优先级和结合性；⑤重载运算符的函数不能有默认的参数；⑥重载的运算符必须和用户定义的自定义类型的对象一起使用，至少应有一个是类对象，即不允许参数全部是 c++的标准类型。故本题答案为 c。

2 在类的定义中，用于为对象分配内存空间，对类的数据成员进行初始化并执行其他内部管理操作的函数是()。

- A. 友元函数 B. 虚函数 C. 构造函数 D. 析构函数

参考答案：C

参考解析：构造函数在创建对象时系统自动调用，其功能是使用给定的值将对象初始化。

3 软件设计中模块划分应遵循的准则是()。

- A. 低内聚低耦合
- B. 高内聚低耦合
- C. 低内聚高耦合
- D. 高内聚高耦合

参考答案：B

参考解析：软件设计中模块划分应遵循的准则是高内聚低耦合、模块大小规模适当、模块的依赖关系适当等。模块的划分应遵循一定的要求，以保证模块划分合理，并进一步保证以此为依据开发出的软件系统可靠性强，易于理解和维护。模块之间的耦合应尽可能的低，模块的内聚度应尽可能的高。

4 如果类 A 被说明成类 B 的友元，则()。

- A. 类 A 的成员即类 B 的成员
- B. 类 B 的成员即类 A 的成员
- C. 类 A 的成员函数不得访问类 B 的成员
- D. 类 B 不一定是类 A 的友元

参考答案：D

参考解析：本题考查的是友元类的定义。类 A 为类 B 的友元类，则类 A 的所有成员函数都是类 B 的友元函数，但反之则不一定成立。

5 有如下类定义：

```
class Foo
{
public:
    Foo(int v): value(V) {}//①
    ~Foo() {}//②
private:
    Foo() {}//③

    int value=0; //④
};
```

其中存在语法错误的行是()。

A. ①B. ②C. ③D. ④

参考答案: D

参考解析: 本题考查对象的初始化, 类的数据成员是不能在声明类时初始化的, 所以本题答案为 D。

6 下面关于数组的描述错误的是()。

A. 在 C++ 语言中数组的名字就是指向该数组第一个元素的指针

B. 长度为 n 的数组, 下标的范围是 0~n-1

C. 数组的大小必须在编译时确定

D. 数组只能通过值参数和引用参数两种方式传递给函数

参考答案: D

参考解析: 本题考查的是数组的基本使用, 数组还可以通过对应的指针调用传递参数, 其余选项的说法都是正确的。

7 运算符重载是对已有的运算符赋予多重含义, 因此()。

A. 可以对基本类型(如 int 类型)的数据, 重新定义“+”运算符的含义

B. 可以改变一个已有运算符的优先级和操作数个数

C. 只能重载 c++ 中已有的运算符, 不能定义新运算符

D. C++ 中已有的所有运算符都可以重载

参考答案: C

参考解析: 重载运算符的规则如下: ①c++ 不允许用户自己定义新的运算符, 只能对已有的 c++ 运算符进行重载; ②c++ 不能重载的运算符只有 5 个; ③重载不能改变运算符运算对象的个数; ④重载不能改变运算符的优先级和结合性; ⑤重载运算符的函数不能有默认的参数; ⑥重载的运算符必须和用户定义的自定义类型的对象一起使用, 至少应有一个是类对象, 即不允许参数全部是 c++ 的标准类型。故本题答案为 c。

8 有如下类和对象的定义：

```
class Constants{  
  
public:  
  
static double getPI() {return 3.1416;}  
  
};
```

Constants constants;

下列各组语句中，能输出 3.1416 的是()。

A. cout

参考答案：D

参考解析：本题考查静态成员函数。静态成员函数由于没有 this 指针，所以不能访问本类中的非静态成员。引用静态成员函数有 2 种方式，一种是直接使用类来引用，即 Constants::getPI()，另一种是使用对象来引用，即 constants.getPI()。所以本题答案为 D。

9 程序流程图中带有箭头的线段表示的是()。

A. 图元关系 B. 数据流 C. 控制流 D. 调用关系

参考答案：C

参考解析：在数据流图中，用标有名字的箭头表示数据流。在程序流程图中，用标有名字的箭头表示控制流。所以选择 C。

10 下列程序的执行结果为()。

```
#include  
  
void main()  
  
{  
  
int a=3, b=0;
```

```
int*P = &a;

b=+a++;

cout<<*P<<"", "<

}
```

A. 3, 4 B. 4, 3 C. 3, 3 D. 4, 4

参考答案: B

参考解析: *P 为所指对象 a 的值, 语句 “b=+a++;” 等价于 “b=b+a; a=a+1;”。

11 对长度为 n 的线性表排序, 在最坏情况下, 比较次数不是 $n(n-1)/2$ 的排序方法是()。

A. 快速排序 B. 冒泡排序 C. 直接插入排序 D. 堆排序

参考答案: D

参考解析: 除了堆排序算法的比较次数是 $O(n\log_2 n)$, 其他的都是 $n(n-1)/2$ 。

12 字面常量 42、4.2、42L 的数据类型分别是()。

A. long, double, int
B. long, float, int
C. int, double, long
D. int, float, long

参考答案: C

参考解析: 本题考查整型变量的类型和浮点变量的类型, 42 默认为 int 型数据, 42 后面加上 L 后, 表示 long 型数据, 4.2 默认为 double 型数据。

13 下列叙述中正确的是()。

A. 栈是 “先进先出” 的线性表

B. 队列是“先进后出”的线性表

C. 循环队列是非线性结构

D. 有序线性表既可以采用顺序存储结构，也可以采用链式存储结构

参考答案：D

参考解析：栈是先进后出的线性表，所以 A 错误；队列是先进先出的线性表，所以 B 错误；循环队列是线性结构的线性表，所以 C 错误。

14 不能作为重载函数的调用的依据是()。

A. 参数个数 B. 参数类型 C. 函数类型 D. 函数名称

参考答案：D

参考解析：所谓重载函数是指同一个函数名可以对应多个函数的实现，编译器会根据参数个数、参数类型和函数返回值类型不同而自动调用相应的函数。

15 下列语句中，错误的是()。

A. `const int buffer=256;` B. `const double* point;` C. `int const buffer=256;` D. `double* const point;`

参考答案：D

参考解析：const 是一个 C++ 关键字，用于限定不允许改变的变量。选项 B 是指向常量的指针，定义时可以不初始化；选项 D 是指针常量，定义时必须初始化。故答案为 D。

16 下列运算符中，在 C++ 语言中不能重载的是()。

A. `*` B. `>=` C. `?:` D. `/`

参考答案：C

参考解析：不能被重载的运算符还包括：“.”、“.*”、“->*”和“?:”。

17 有如下说明：

`int a[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; *P=a;` 则数值为 9 的表达式是()。

A. *P+9B. 木(P+8)C. *P+=9D. P+8

参考答案: B

参考解析: 本题考查指向数组的指针, 本题中指针变量 P 指向数组 a, 那么 *P 的值就是 1, 即表示数组的第一个元素, 那么数值为 9 的表达式就是将指针向后移动 8 个地址, 即 P+8 指向的是数组元素值为 9, 所以数值为 9 的表达式是 *(P+8)。

18 执行下列语句段后, 输出字符 “*” 的个数是 ()。

```
for(int i=50;i>1;i-=2) Cout<<'*';
```

A. 24B. 25C. 26D. 50

参考答案: B

参考解析: 本题考查 for 循环语句, 题目中每执行完循环体后, i 都会减 2, 那么只有 50 到 2 之间的偶数才能输出 “*”, 所以总共输出 25 次。

19 算法的有穷性是指 ()。

- A. 算法程序的运行时间是有限的
- B. 算法程序所处理的数据量是有限的
- C. 算法程序的长度是有限的
- D. 算法只能被有限的用户使用

参考答案: A

参考解析: 算法原则上能够精确地运行, 而且人们用笔和纸做有限次运算后即可完成。有穷性是指算法程序的运行时间是有限的。

20 下面关于 break 语句的描述中, 不正确的是 ()。

- A. break 可以用于循环体内
- B. break 语句可以在 for 循环语句中出现多次
- C. break 语句可以在 switch 语句中出现多次

D. break 语句可用于 if 条件判断语句内

参考答案: D

参考解析: 本题考查的是 break 语句的使用, break 可以结束 switch 语句和 for 循环语句, 但是不能使用在条件判断语句内。

21 有如下类定义:

```
class Point
{
    int xx, yy;

public:
    Point(): xx(0), yy(0) {}
    Point(int x, int Y=0): xx(x), YY(Y) {}
};
```

若执行语句

```
Point a(2), b[3], *c[4];
```

则 Point 类的构造函数被调用的次数是()。

A. 2 次 B. 3 次 C. 4 次 D. 5 次

参考答案: C

参考解析: 本题考查默认构造函数和带参数的构造函数, 题目中定义一个对象 a(2) 以及对象数组 b[3], 共执行 4 次构造函数, 对象指针不调用构造函数。所以本题答案为 c。

22 有如下程序:

```
#include
```



```
using namespace std;
```

```
class Base1
```

```
{
```

```
public:
```

```
Base1(int D. {cout<
```

```
— Base1() {}
```

```
};
```

```
class Base2
```

```
{
```

```
public:
```

```
Base2(int D. {tout<
```

```
~Base2() {}
```

```
};
```

```
class Derived: public Base1, Base2
```

```
{
```

```
public:
```

```
Derived(int a, int b, int c, int D.: Base1(B., Base2(A., b1(D., b2(C. {}
```

```
private:
```

```
int b1:
```

```
int b2:
```

```
};
```

```
int main()

{

Derived d(1, 2, 3, 4);

return 0;

}
```

执行这个程序的输出结果是()。

A. 1234B. 2134C. 12D. 21

参考答案: D

参考解析: 本题考查派生类的构造函数和析构函数, 在定义一个派生类的对象时, 先调用基类的构造函数, 然后再执行派生类的构造函数, 对象释放时, 先执行派生类的析构函数, 再执行基类的析构函数。本题中定义一个派生类对象时, 分别调用基类的构造函数, 所以分别输出 21。

23 以下叙述正确的是()。

- A. 函数可以嵌套定义但不能嵌套调用
- B. 函数既可以嵌套调用也可以嵌套定义
- C. 函数既不可以嵌套定义也不可以嵌套调用
- D. 函数可以嵌套调用但不可以嵌套定义

参考答案: D

参考解析: 在 C++ 语言中, 所有的函数都是平行的, 即在定义函数时是互相独立的, 一个函数并不属于另一个函数, 即函数不能嵌套定义, 但可以互相调用。但是不能调用 main() 函数。

24 必须用一对大括号括起来的程序段是()。

A. switch 语句中的 case 标号语句

B. if 语句的分支

C. 循环语句的循环体

D. 函数的函数体

参考答案：D

参考解析：本题考查 c++ 基本控制结构，A、B、C 三个选项中的大括号不是必须的，而 D 选项中的函数体必须使用大括号。

25 软件按功能可以分为应用软件、系统软件和支撑软件(或工具软件)。下面属于应用软件的是()。

A. 学生成绩管理系统

B. C 语言编译程序

C. UNIX 操作系统

D. 数据库管理系统

参考答案：A

参考解析：软件按功能可以分为：应用软件、系统软件、支撑软件。操作系统、编译程序、解释程序、网络软件、数据库管理系统都属于系统软件。所以 B、C、D 都是系统软件，只有 A 是应用软件。

26 下面程序的运行结果为()。

```
#include
```

```
void main()
```

```
{
```

```
for(int a=0, x=0;!x&& a<=10;a++)
```

```
{
```

```
a++;
```

```
}  
  
cout<  
  
}
```

A. 10B. 11C. 12D. 0

参考答案: C

参考解析: 在进入最后一次循环时 a=10, 在循环体内执行 “a++;” 后 a=11, 执行完该次循环体计算 for 循环的增量部分后 a=12 小于 10, 循环结束。

27 有如下程序:

```
#include  
  
using namespace std;  
  
int i=1;  
  
class Fun{  
public:  
    static int i;  
    int value() {return i-1;}  
    int value() const {return i+1;}  
};  
  
int Fun::i=2;  
  
int main() {  
  
    int i=3;  
  
    Fun fun1;
```

```
const Fun fun2:
```

```
return 0;
```

```
}
```

若程序的输出结果是：

123

则程序中下画线处遗漏的语句是()。

A. cout<

参考答案：A

参考解析：本题考查构造函数、常成员函数和静态数据成员。外部同名的变量赋值，不能改变类的静态变量值，所以 Fun 对象中的 i 值为 2，所以本题答案为 A。

28 下列有关模板的叙述中，正确的是()。

A. 函数模板不能含有常规形参

B. 函数模板的一个实例就是一个函数定义

C. 类模板的成员函数不能是模板函数

D. 用类模板定义对象时，绝对不能省略模板实参

参考答案：B

参考解析：本题考查函数模板和类模板的基础知识，所以本题答案为 B。

29 运算符重载时不需要保持的性质是()。

A. 操作数个数 B. 操作数类型 C. 优先级 D. 结合性

参考答案：B

参考解析：重载运算符的规则如下：①c++不允许用户自己定义新的运算符，只能对已有的 c++运算符进行重载；②c++不能重载的运算符只有 5 个；③重载不能改变运算符运算对象的个数；④重载不能改变运算符的优先级和结合性；⑤重载运算符的函数不能有默认的参数；⑥重载的运算符必须和用户定义的自定义类型的对象一起使用，至少应有一个是类对象，即不允许参数全部是 c++的标准类型。所以答案为 B。

30 下列程序的输出结果为()。

```
#include  
  
{  
  
if(n<1)return 1;  
  
else return n+time(n-1); return 0;  
  
}  
  
void main() {  
  
cout<<  
  
}
```

A. 0B. 10C. 15D. 16

参考答案：B

参考解析：本题考查的是递归函数的使用，题中递归函数的结束条件为 n

31 有如下程序：

```
#include  
  
using namespace std;  
  
class MyClass{  
  
public:  
  
MyClass(int i=0) {cout<<1;}
```

```

MyClass(const MyClass&x) {cout<<2;}

MyClass&operator=(const MyClass&x) {cout<<3;return $this;}

~MyClass() {cout<<4;}

};

int main()

{

MyClass obj1(1), obj2(2), obj3(obj1);

return 0;

{

```

执行这个程序的输出结果是()。

A. 112444 B. 11114444 C. 121444 D. 11314444

参考答案: A

参考解析: 本题: 考查构造函数和复制构造函数以及重载运算符。本题中先后定义了两个对象 obj1, obj2, 依次输出 1, 2, 然后定义了对象 obj3, 调用复制构造函数, 输出 2, 最后执行三次析构函数, 输出 444。

32 在下列模式中, 能够给出数据库物理存储结构与物理存取方法的是()。

A. 外模式 B. 内模式 C. 概念模式 D. 逻辑模式

参考答案: B

参考解析: 数据库系统的三级模式是概念模式、外模式和内模式。概念模式是数据库系统中全局数据逻辑结构的描述, 是全体用户公共数据视图。外模式也称子模式或用户模式, 它是用户的数据视图, 给出了每个用户的局部数据描述。内模式又称物理模式, 它给出数据库物理存储结构与物理存取方法, 所以选择 B。

33 有如下程序:

```

#include using namespace std; Class Base{

public:

void output() {Cout<<1;}

virtual void Print() {Cout<<'B' ;} };

Class Derived: public Base{ public:

void output() {Cout<<2;} void Print() {Cout<<' D ';} };

int main() {

Base * ptr=new Derived; ptr->output();

ptr->Print(); delete ptr;

return 0; }

```

执行这个程序的输出结果是()。

A. 1B. 1D. 2B. 1D. 2

参考答案: B

【答案解析】 在某基类中声明为 virtual 并在一个或多个派生类中被重新定义的成员函数，本题中定义了一个指向派生类对象的基类指针 ptr，执行 ptr->output() 后，会执行基类的 output 函数，输出 1，由于 Print 是虚函数，所以 ptr->Print() 会执行派生类的 Print，即输出 D，所以本题答案为 B。

34 以下程序中，错误的行为是()。

①#include

②class A

③{

④public:

⑤ int n=2;

⑥ A(int val) {cout<

⑦ ~A() {};

⑧};

⑨void main() ⑩{

⑪ A a(0);

⑫}

A. ⑤B. ⑥C. ⑦D. ⑪

参考答案: A

参考解析: C++语言中规定在类体中不允许对所定义的数据成员进行初始化。

35 下列数据结构中, 属于非线性结构的是()。

A. 循环队列 B. 带链队列 C. 二叉树 D. 带链栈

参考答案: C

参考解析: 树是简单的非线性结构, 所以二叉树作为树的一种也是一种非线性结构。

36 有如下程序:

```
#include using namespace std; Class A{
public:
static int a,
void init(    ){。 a=1;}

A(int a=2){init(    );a++;} };

int A: : a: 0: A obj;
```

```
int main(    ) {

    Cout<

}


```

运行时输出的结果是()。

A. 0B. 1C. 2D. 3

参考答案: B

参考解析: 本题考查构造函数, 本题中对类中的 a 数据成员先是赋值为 0, 当定义了对象 obj 时, 执行构造函数, 将 a 的值变为了 1, 所以输出 1。

37 软件详细设计产生的图如下:



该图是()。

A. N—S 图 B. PAD 图 C. 程序流程图 D. E—R 图

参考答案: C

参考解析: N—S 图提出了用方框图来代替传统的程序流程图, 所以 A 不对。PAD 图是问题分析图, 它是继承程序流程图和方框图之后提出的又一种主要用于描述软件详细设计的图形表示工具, 所以 B 不对。E—R 图是数据库中的用于表示 E—R 模型的图示工具, 所以 D 不对。根据图中所示表示方法是进行软件详细设计时使用的程序流程图。

38 秘序调试的任务是()。

A. 设计测试用例 B. 验证程序的正确性 C. 发现程序中的错误 D. 诊断和改正程序中的错误

参考答案：D

参考解析：程序调试的任务是诊断和改正程序中的错误。

39 有如下程序：

```
#include
```

```
using namespace std;
```

```
class A{
```

```
public:
```

```
virtual void func1() {cout<<" A1 ";}
```

```
void func2() {cout<<" A2 ";}
```

```
};
```

```
class B: public A{
```

```
public:
```

```
void func1() {tout<<" B1 ";}
```

```
void func2() {tout<<" B2 ";}
```

```
};
```

```
int main()
```

```
{ A*P=new B;
```

```
P->func1();
```

```
P->func2();
```

```

delete p;

return 0;

}

```

执行这个程序的输出结果是()。

A. B182B. A1A2C. B1A2D. A1B2

参考答案: C

参考解析: 本题考查虚函数。虚函数指在基类中声明为 virtual 并在一个或多个派生类中被重新定义的成员函数。题中定义了一个派生类对象 8, 并使用类 A 的指针指向 B, 由于基类 A 中的 func1 为虚函数, 所以会执行派生类的 func1, 输出 11, 而基类中的 func2 不是虚函数, 所以 P->func2 会执行基类的 func2, 输出 A2。故答案为 c。

40 下列字符串中。不可以用做 C++ 标识符的是()。

A. y_2006B. _EST_HC. ReturnD. switch

参考答案: D

参考解析: 本题考查 C++ 标识符的命名规则。其规则有如下几点: ①所有标识符必须由字母、数字或下划线组成, 且必须由字母或下划线开头; ②所有标识符不能使用 C++ 已有的关键字; ③大、小写字母表示不同意义, 即代表不同的标识符。选项 C 中虽然 return 是关键字, 但因首字母大、小写不同, 也可用于标识符, 但不建议使用。故答案为 D。

二、基本操作题

41 请使用 VC6 或使用【答题】菜单打开考生文件夹 proj1 下的工程 proj1, 其中有枚举 DOGCOLOR、狗类 Dog 和主函数 main 的定义。程序中位于每个“//ERROR *****found*****”下的语句行有错误, 请加以改正。改正后程序的输出结果应该是:

There is a white dog named Hoh0. There is a blaCk dog named Haha. There is a motley dog named Hihi. 注意: 只修改每个//ERROR*****found*****女料下的那一行, 不要改动程序中的其他内容。

```
#include using namespace std; //狗的颜色: 黑、白、黄、褐、花、其他
```

```
enum DOGCOLOR{BLACK, WHITE, YELLOW, BROWN, PIEBALD, OTHER}; Class Dog{//  
狗类
```

```
DOGCOLOR Color; Char name[20]; static int Count; public:
```

```
Dog(Char name[], DOGCOLOR Color){ strcpy(this->name, name);
```

```
//ERROR*****found*****、
```

```
strcpy(this->Color, Color);
```

```
} 1 DOGCOLOR getColor()Const{return Color;}
```

```
//ERROR*****found*****
```

```
Const Char * getName()Const{return &finin9;} Const Char *  
getColorString()Const{
```

```
switch(Color){
```

```
Case BLACK: return "black"; Case WHITE: return "white";
```

```
Case YELLOW: return "yellow"; Case BROWN: return "brown"; Case PIEBALD:  
return "piebald";
```

```
}
```

```
return "mottley";
```

```
}
```

```
void show()Const{ Cout<<" There is a" <
```

```
} }; in main(){
```

```
//ERROR*****found*****
```

```
Dog dog1(" Hoh0" , WHITE), dog2(" Haha" , BLACK);dog3(" Hihi" , OTHER);  
dog1.show();
```

```
dog2.show(); dog3.show(); return 0;
```

```
}
```

参考解析：

```
(1) this->color=color;
```

```
(2) const char * getName() const {return *name;}
```

```
(3) Dog dog1("Hoho", WHITE), dog2("Haha",
```

```
BLACK), dog3("Hihi", OTHER);
```

三、简单应用题

42 请使用 VC6 或使用【答题】菜单打开考生文件夹 proj2 下的工程 proj2 (此工程中包含一个源程序文件 main.cpp，其中有一个“房间”类 Room 及其派生出的“办公室”类 Office 的定义，还有主函数 main 的定义。请在程序中“//****found****”下的横线处填写适当的代码并删除横线，以实现上述类定义。此程序的正确输出结果应为：

办公室房间号：308

办公室长度：5.6

办公室宽度：4.8

办公室面积：26.88

办公室所属部门：会计科

注意：只能在横线处填写适当的代码，不要改动程序中的其他内容，也不要删除或移动“//料料 found**料”。

```
#include
```

```
using namespace std;
```

```
class Room{ // “房间”类
```

```
int room_n0;//房间号
```

```
double length;//房间长度(m)
```

```

double width;//房间宽度(n1)

public:

Room(int the—room—n0,double the~length,double the—width):loom—no(the
—room—no), length(the—

length), width(the—width) {}

int theRoomNo()const{return iDonl—n0;}

//返回房间号

double theLength()const{return length;}//返回房间长度

double theWidth()const{return width;}//返回房间宽度

//*****found*****

double theArea()const{_____} //返回房间面积(矩形面积)

};

class Office:public Room{//“办公室”类
char* depart;//所属部门

public:

Office(int the_room_n0,double the_length, double the_width, const char
本 the_depart)

//*****found*****

: _____{

depart=new char[strlen(the depart)+1];

//*****found*****

strcpy(_____);

```

```

}

~Office() {delete[]depart;}

const char*theDepartment()const{return depart;}//返回所属部门

};

int main(){

//*****found*****

Office_____

cout<<" 办公室房间号: " <

<<" 办公室长度: " <

<<" 办公室宽度: " <

<<" 办公室面积: " <

<<" 办公室所属部门: " <

return 0;

}

```

参考解析:

- (1)returtl length*width;
- (2)Room(the_room_no, the_lengih, the_width)
- (3)depart, the_depart
- (4)an_office(308, 5.6, 4.8, " 会计科")

四、综合应用题

43 请使用 VC6 或使用【答题】菜单打开考生目录 proj3 下的工程文件 proj3，此工程中包含一个源程序文件 proj3.cpp，补充编制 c++ 程序 proj3.cpp，其功能是读取文本文件 in.dat 中的全部内容，将文本存放到 doc 类的对象 myDoc 中。然后将 myDoc 中的字符序列反转，并输出到文件 out.dat 中。文件 in.dat 的长度不大于 1000 字节。

要求：

补充编制的内容写在“//*****”与“//*****”两行之间。实现将 myDoc 中的字符序列反转，并将反转后的序列在屏幕上输出。不得修改程序的其他部分。

注意：程序最后已将结果输出到文件 Out.dat 中，输出函数 writeToFile 已经给出并且调用。

```
//proj3.cpp

#include

#include

#include

using namespace std;

class doc
{
private:
    char *sir; //文本字符串首地址
    int length; //文本字符个数
public:
    //构造函数，读取文件内容，用于初始化新对象，filename 是文件名字符串首地址
    doc(char *filename);

    void reverse(); //将字符序列反转
```

```
一 doc();

void writeToFile(char*filename);

};

doc: : doc(char*filename)

{

ifstream myFile(filename);

int len=1001, tmp;

atr=new char[len];

length=0;

while((trap=myFile.get())!=EOF)

{

str[length++]=trap;

}

str[length]='\0';

myFile.close();

}

void doc: : reverse() {

//将数组 atr 中的 length 个字符中的第一个字符与最后一个字符交换，第二个字
符与倒数第二个

//字符交换……

//*****333*****
```

```

//*****666*****

}

doc: : ~doe()

{

delete[]str;

}

void doc: : writeToFile(char*filename)

{

Ofstream outFile(filename);

outFile<

outFile.close();

}

void main()

{

doc m_Doc(" in.dat" );

myDoc.reveme();

myDoc.writeToFile(" out.dat" );

}

```

参考解析:

```
int j, J;//定义两个整数 1 临时变量 i. j.
```

```
for(i=0, j=length-1;i
```

```
chartemp=str[i];

//把 str[i]中的值保存在临时变量 temp

str[i]=str[j];

//把 Str[j]值赋给 Str[i]，实现字符前后替换

str[j]=temp;

//把保存在临时变量 temp 中的值再赋值给 Str[j]

}
```

【考点分析】

本题主要考查的是 doc 类、构造函数、成员函数和析构函数。

【解题思路】

题目要求将 myDoc 中的字符序列反转，在 main 函数中我们看到 myDoc 是 doc 类，根据 doc 类的定义可以知道它是把读取文件的字符串存到 str 动态数组中。reverse 函数实现将数组 str 中的 length 个字符中的第一个字符与最后一个字符交换，第二个字符与倒数第二个字符交换，依次类推。使用变量 i 和 j，分别表示第一个字符和最后一个字符的下标，定义 temp 作为中间变量进行交换。

第 43 题 笔记记录我的笔记(0) | 精选笔记(1) 标签: 试题内容(1) 答案解析(0)