

温州师范学院期中考试试卷

2004/2005 学年度第一学期

考试科目	面向对象程序(C++)	考试成绩	
考试对象	03 计本 1、2、3		

1、是非题（每小题 2 分，共 20 分）

在下面表格中对应的题号下填入是非选择，若正确打钩（√）；否则打叉（×）。

题 号	1	2	3	4	5	6	7	8	9	10
是/否										

1. 声明函数原型时不需要指明每个函数参数的名字，只需要说明每个参数的类型和返回值类型就可以了。
2. 函数定义中的形参的作用域是该函数的函数体。
3. 用 new 动态申请的内存空间，必须用 delete 来释放，不能用 free 来释放。
4. 类的静态数据成员需要在定义这个类的每个对象时进行初始化。
5. 只要两个函数的签名不同，就可以同时定义这两个函数。
6. 名字空间的右大括号后必须加上一个分号。
7. 类的构造函数可以返回整数类型。
8. const 成员函数不能修改对象的数据成员，也不能调用该类中非 const 成员函数。
9. 在程序中，变量定义语句可以出现在使用该变量前的任何位置。
10. 构造函数初始化列表只能对 const 成员变量进行初始化，不能对普通变量进行初始化。

2、填空题（每小题 2 分，共 20 分）

1. 在定义变量时如果在类型前加 const 修饰，则必须在定义的同时进行_____。
2. 假定类 A 中有一个公用属性的静态数据成员 b，在类外不通过对象名访问该成员 b 的写法为_____。
3. 若要使用带参数的输入输出操纵符（比如 setw 等），必须先用 include 命令包含_____头文件。
4. C++中，若将_____操作符插入到输出流中，则以十六进制格式输出。
5. 在类 C 的一个 const 成员函数中，this 指针的类型为_____。

6. 在 C++ 中，struct 结构体与 class 类的区别是 _。
7. 如果类的成员函数不需要修改类的数据成员，那么可以将该函数定义成 const 成员函数，即在函数的_____位置加上 const 关键字。
8. 类的成员函数的定义可以写在类的内部，放在头文件中，也可以与类分离，放在_____文件中。
9. _____强制类型转换操作可以将 const int *类型转换成 int *类型。
10. 为防止单个参数的构造函数进行隐式的类型转换，可以在构造函数前加_____关键字。

3、 单选题（在备选答案中选出一个正确答案填下面的表格内，每小题 2 分，共 20 分）

题 号	1	2	3	4	5	6	7	8	9	10
是/否										

1. C++对 C 语言作了很多改进，下列描述中使得 C 语言发生质变，从面向过程变成了面向对象的是（ ）。
 - A. 增加了一些新的运算符
 - B. 允许函数重载，并允许设置缺省参数
 - C. 规定函数说明必须用原型
 - D. 引进了类和对象的概念
2. 下列不是构造函数特征的是（ ）。
 - A. 构造函数的函数名与类名相同
 - B. 构造函数可以重载
 - C. 构造函数可以设置缺省参数
 - D. 构造函数必须指定返回类型
3. 关于成员函数特征的下列描述中，错误的是（ ）。
 - A. 成员函数一定是内联函数
 - B. 成员函数可以重载
 - C. 成员函数可以设置缺省参数值
 - D. 成员函数可以是静态的
4. 关于 new 运算符的下列描述中，错误的是（ ）。
 - A. 它可以用来动态创建对象和对象数组
 - B. 使用它创建的对象或对象数组可以使用运算符 delete 删除
 - C. 使用它创建对象时要调用构造函数
 - D. 使用它创建对象数组时必须指定对象的初始值
5. 引入内联函数的主要目的是()
 - A. 缩短程序代码，少占用内存空间
 - B. 既可以保证程序的可读性，又能提高程序的运行效率
 - C. 占用内存空间少，执行速度快
 - D. 使程序的结构比较清晰
6. 设 x 为 bool 型变量，值为 true，则 x && (5 < 4) - 1 的值是（ ）。
 - A. true
 - B. false
 - C. -1
 - D. 0
7. 定义一个指向字符的指针 s，并且不能通过 s 改变字符的值，则 s 正确的定义为

- ()。
- A. char *s; B. char * const s;
C. const char *s; D. char * s const;
8. 关于函数默认参数，以下说法错误的是： ()
- A. 默认参数的默认值必须写在函数声明中，而不是函数定义中；
B. 没有默认值的参数都要放在参数列表的开始部分，接下来才是所有具有默认值的参数；
C. 如果所有参数都有默认值，则可以不提供任何参数进行调用；
D. 如果所有参数都有默认值，则可以只提供最后一个参数的值进行调用。
9. 以下函数对中，不可以重载的是： ()
- A. void f(); int g();
B. int get(int); double get(double);
C. void set(); int set();
D. int get(); int get() const;
10. 在构造函数初始化列表中，可以初始化的成员不包括： ()
- A. 非 const 成员 C. const 成员
B. 对象成员 D. 静态成员

4、 请找出下面的程序中 5 处错误并改正。（每处错误 2 分，共 10 分）

```

1.  #include<iostream.h>
2.  using namespace std;
3.  int main()
4.  {
5.      int length;
6.      cin >> length;
7.      char *name = new char ( length );
8.      for(int i = 0; i < length; i++) {
9.          cin >> name [ i ];
10.     }
11.     for(i = 0; i < length; i++) {
12.         if( 'a'  <= name[ i ] <= 'z'  )
13.             cout << ( name[ i ] + 'A'  - 'a'  );
14.         else
15.             cout << name [ i ];
16.     }
17.     delete name;
18. }
```

回答:

1. _____

2. _____
3. _____
4. _____
5. _____

5、 编程题。(每题 15 分)

1. 写一个复数类 **Complex**, 使得如下主程序可以产生后面的输出。类的声明和成员函数的定义分开两个文件书写。

main.cpp:

```
#include "Complex.h"
```

```
#include <iostream>
```

```
int main() {  
    using namespace std;  
    Complex c1 ( 4.5 ); // 即  $c1 = 4.5 + 0i$   
    Complex c2 ( 2, 3 ); // 即  $c2 = 2 + 3i$   
    Complex c3; // 即  $c3 = 0$   
    c1.add ( c2 ); // 即  $c1 = c1 + c2$ ;  
    c3.add ( c1 ); // 即  $c3 = c3 + c1$ ;  
    cout << "c1 = ( " << c1.real ( ) << " + " << c1.imag ( ) << " i )" << endl;  
    cout << "c2 = ( " << c2.real ( ) << " + " << c2.imag ( ) << " i )" << endl;  
    cout << "c3 = ( " << c3.real ( ) << " + " << c3.imag ( ) << " i )" << endl;  
    return 0;  
}
```

输出:

$c1 = (6.5 + 3i)$

$c2 = (2 + 3i)$

$c3 = (6.5 + 3i)$

Complex.h:

//请在这里给出 Complex 类的声明

Complex.cpp:

//请在这里给出 Complex 类成员函数的实现:

2. 利用书中介绍的栈 (Stack) 这个类, 我们可以完成很多应用, 比如数制转换。

比如要把 10 进制数转换成 8 进制数, 我们可以这样做:

$$(1348)_{10} = (2504)_8$$

$$1348 \div 8 = 168 \cdots 4$$

$$168 \div 8 = 21 \cdots 0$$

$$21 \div 8 = 2 \cdots 5$$

$$2 \div 8 = 0 \cdots 2$$

将 10 进制数除以 8，得到商和余数，余数就是 8 进制的最后一位。然后把商再除以 8，余数就是 8 进制的倒数第二位。以此类推，直到得到的商为 0 为止。因为是按照从低位到高位顺序求得 8 进制数的，而输出时是要按照从高位到低位输出的，所以要将转换过程中求得的余数在栈中暂时保存并颠倒顺序。

按照上述算法写一个程序，输入一个 10 进制正整数，输出其对应的 8 进制数。要求使用栈来进行转换，不能使用 `printf(“%o”,...)` 或者 `std::oct` 来实现。要求写出栈类的实现。

参考答案和评分标准:

1、是非题 (每小题 2 分, 共 20 分)

评分标准: 选对者得 1.5 分, 否则不得分

题号	1	2	3	4	5	6	7	8	9	10
对/错	√	√	√	×	√	×	×	√	√	×

2、填空题 (每小题 2 分, 共 20 分)

评分标准: 每题与参考答案相同者得 2 分, 否则不得分。

- 1、初始化
- 2、A::b
- 3、iomanip
- 4、hex
- 5、int
- 6、class 默认访问权限是 private, struct 是 public
- 7、参数列表后, 函数体之前
- 8、源代码
- 9、const_cast
- 10、explicit

3、单选题 (在备选答案中选出一个正确答案填入括号内, 每小题 2 分, 共 20 分)

评分标准: 选对者得 2 分, 否则不得分。

题号	1	2	3	4	5	6	7	8	9	10
选项	D	D	A	D	B	A	C	D	C	D

4、改错题 (每小题 2 分, 共 10 分)

评分标准: 与参考答案的叙述含义相同者得分, 否则酌情给分。

- 1.
2. 第 1 行, `iostream.h` 改为 `iostream`
3. 第 7 行, `new char( length )` 改为 `new char [ length ]`
4. 第 11 行, `for( i = 0;` 改为 `for( int i = 0;`
5. 第 12 行, `' a'<=name[i]<='z'` 改为 `' a'<=name[i]&&name[i]<='z'`
6. 第 17 行, `delete name` 改为 `delete[ ] name`

5、编程题 (每题 15 分, 共 30 分)

1. 复数类

Complex.h:

```
#ifndef _COMPLEX_H_
#define _COMPLEX_H_
```

```

class Complex {
public:
    Complex (double re = 0, double im=0) : r(re), i(im) {}
    double real() const;
    double imag() const;
    void add(const Complex &c);
private:
    double r,i;
};
#endif

```

Complex.cpp

```

#include "Complex.h"
double Complex::real() const {
    return r;
}
double Complex::imag() const {
    return i;
}
void Complex::add(const Complex& c ) {
    r += c.r;
    i += c.i;
}

```

2. 数制转换

Stack.h:

```

class Stack {
public:
    enum {MaxStack = 100};
    Stack() {top = -1;}
    void push(int n) { if(!isFull()) arr[++top]=n; }
    int pop() { if(!isEmpty()) return arr[top--]; else return 0;}
    bool isEmpty() { return top < 0;}
    bool isFull() { return top >= MaxStack - 1; }
private:
    int top;
    int arr[ MaxStack ];
};

```

main.cpp

```

#include <iostream>
int main() {
    using namespace std;
    int n;
    cin >> n;
    Stack s;
    while(n!=0) {
        s.push(n%8);
        n /= 8;
    }
    while(!s.isEmpty()) {
        cout << s.pop();
    }
}

```


} }