

南京理工大学课程考试试卷 (学生考试用)

课程名称: C++程序设计 学分: 4 教学大纲编号: 06039701
试卷编号: 06039701-23A 考试方式: 闭卷 满分分值: 100 考试时间: 120 分钟
组卷日期: 2024 年 1 月 14 日 组卷教师(签字): C++教学组 审定人(签字): 靳从

一、选择题 (每题 1 分, 共 20 分)

1. 下列选项中能够作为 C++ 合法整型常量的是【1】。
A. 1010B B. 031 C. x12 D. 32H
2. 若有定义 “float a[2][3];”, 则 sizeof(a) 的值是【2】。
A. 6 B. 12 C. 24 D. 48
3. 与表达式 “x>0 && x<=10” 相反的表达式为【3】。
A. x<=0 || x>10 B. x<=0 && x>10 C. x<=0 || x<10 D. x>0 && x>10
4. 以下给字符数组赋初值, 正确的是【4】。
A. char s1[4] = {"ABCD"}; B. char s2[4] = { "A","B","C","D" };
C. char s3[][4] = { "A","B","C","D"}; D. char s4[4][] = {'A','B','C','D'};
5. 有如下代码 “int a[10]={1,3,4,5,7,8,10},*p=&a[5];”, 则*(p-1)的值为【5】。
A. 4 B. 5 C. 7 D. 8
6. 以下函数默认值定义错误的是【6】。
A. fun(int x, int y = 0) B. fun(int x = 100) C. fun(int x = 0, int y) D. fun(int x = f()) (f已定义)
7. 若定义了函数 “double* fun()”, 则该函数的返回值为【7】。
A. 实数型 B. 实数的地址 C. 函数的地址 D. 指向函数的指针
8. 若有以下说明和定义 “typedef int* INTEGER; INTEGER a,*b;”, 以下正确的是【8】。
A. a是int型变量 B. a是基类型为int的指针变量
C. b是基类型为int的指针变量 D. 用INTEGER代替类型名int
9. 已知一个函数的声明为 “void fun(int x[], int n)”, main 函数中有定义 “int a=1,b[2]={1,2};”, 若在 main 函数中调用 fun 函数, 则以下调用语句正确的是【9】。
A. fun(*a, b); B. fun(b[2], *b); C. fun(&a, b); D. fun(&b[0], b[a]);
10. 若有以下语句 “char s1[] = "123", s2[] = { '1','2','3' };”, 则下面 选项是正确的【10】。
A. s1数组和s2数组的长度相同 B. s1数组的长度大于s2数组的长度
C. s1数组等价于s2数组 D. s1数组的长度小于s2数组的长度
11. 设有说明语句 “int a[2][4] = {{0,2},{3,4}}, *p = (int*)a;”, 则*(p+4)的值为【11】。
A. 0 B. 2 C. 3 D. 4

12. 有结构体定义 “struct Stu { int id; int age; } s,*p = &s;”, 对成员 age 的非法引用的是【12】。
A. s.age B. Stu.age C. p->age D. (*p).age
13. 设有函数原型说明 “double (*p)(int, double);”, 则 p 是【13】。
A. 一个实数 B. 指向一个实数的指针 C. 该函数的地址 D. 指向一个函数的指针
14. 有枚举类型定义 “enum Fruit { apple, orange, banana = 1, peach, grape };”, peach 的值是【14】。
A. 2 B. 3 C. 4 D. 5
15. 假定 MyClass 为一个类, 则执行 “MyClass a, &b = a, c[3], *p[1];” 语句后, 自动调用构造函数的次数是【15】。
A. 3 B. 4 C. 5 D. 6
16. 若有 “class A {int x; public: int *p; A(){p=&x;}}a;”, 如果要给 x 赋值为 1, 下面选项正确的是【16】。
A. a.p = 1; B. a->p = 1; C. a.*p = 1; D. *a.p = 1;
17. 以下选项中, 满足类 A 的复制构造函数在类内的函数声明形式是【17】。
A. A(const &); B. A(A&); C. A(A); D. void A(A & a);
18. C++ 设置虚基类的目的是【18】。
A. 简化程序 B. 消除二义性 C. 提高运行效率 D. 减少目标代码
19. 已知无返回值的 fun 函数是一个类的常成员函数, 下列表示中正确的是【19】。
A. const void fun(); B. void const fun(); C. void fun(const); D. void fun() const;
20. 若以二进制读方式打开一个文件, 则文件的打开模式应为【20】。
A. ios::app | ios::in B. ios::app | ios:: out C. ios:: binary | ios::in D. ios::app | binary:: out

二、基本概念填空题 (每题 2 分, 共 20 分)

1. 定义了宏 “#define S(x,y) x/2+y”, 执行语句 “a = S(4 + 3, 5 + 7) * 3;” 后变量 a 的值为【1】。
2. 已有声明 “float a = 10;”, 表达式 “a/2.5” 的值的数据类型是【2】。
3. C++ 规定在字符串的结尾加上一个【3】, 以便系统据此判断字符串是否结束。
4. 为了得到一个包括 10 个整数的数组空间并由指针 p 指向它, 应使用程序语句【4】。
5. 如果调用函数时需要改变实参的值, 可以采用【5】调用方式。
6. 为了实现换行功能, 除了使用语句 “cout << endl;” 外, 还可以使用语句 “【6】”。
7. 已有语句 “const char* const p = "Hello";”, 所定义的指针 p 和它所指的内容都不能被【7】。
8. 面向对象程序设计的三大特性是封装性、继承性和【8】。
9. 当一个类对象的成员函数被调用时, 该成员函数的【9】指向调用它的对象。
10. 纯虚函数的声明形式是在虚函数声明后面加上【10】。

三、阅读程序填空题（每小题 3 分，共 30 分）

1. 以下程序的输出结果是【1】

```
#include <iostream>
using namespace std;
int main()
{
    int a = 0;
    for (int i = 0; i < 8; i++) {
        switch (i) {
            case 0: a += 5; break;
            case 1: ++a; break;
            default: --a;
        }
        if (i % 3) continue;
        cout << a << "\t";
    }
    return 0;
}
```

2. 以下程序的输出结果是【2】

```
#include <iostream>
using namespace std;
int main()
{
    const char* a[] = {"123","456","789"};
    const char** p = a;
    p++;
    cout << *p << endl;
    return 0;
}
```

3. 以下程序的输出结果是【3】

```
#include <iostream>
using namespace std;
static int n = 3;
void fun(int i)
{
    n--;
    static int n = 0;
    n += i;
    cout << n << "\t";
}
int main()
{
    for (int i = 0; i < 3; i++)
        fun(n);
    return 0;
}
```

4. 以下程序的输出结果是【4】

```
#include <iostream>
using namespace std;
void fun(int* x, int& y, int n = 0)
{
    int a = ++y;
    *x += n;
    a += *x;
    cout << a << "\t";
}
int main()
{
    int a = 6, b = 9, n = 2;
    fun(&a, b);
    fun(&a, b, n);
    cout << b << endl;
    return 0;
}
```

5. 以下程序的输出结果是【5】

```
#include <iostream>
using namespace std;
int fun(int n, int k)
{
    if (k > n) return 0;
    else if (k == 0 || k == n) return 1;
    else return fun(n - 1, k) + fun(n - 1, k - 1);
}
int main()
{
    int n = 3, k = 2;
    cout << fun(n, k);
    return 0;
}
```

6. 以下程序的输出结果是【6】

```
#include <iostream>
using namespace std;
int a = 0;
int add(int x, int y = 1)
{
    return x + y + a++;
}
int fun(int (*p)(int, int), int m, int n)
{
    int b = p(m, n) + add(a);
    cout << a << "\t";
    return b;
}
int main()
{
    int n = fun(add, 3, 5);
    cout << n << endl;
    return 0;
}
```

7. 以下程序的输出结果是【7】

```
#include <iostream>
#include<cstring>
using namespace std;
int main()
{   char str[] = "a a3bccd d";
    char* ptr = str + 1;
    while (*ptr) {
        for (int i = 0; i < ptr - str; i++) {
            if (*ptr == str[i]) {
                strcpy(ptr, ptr + 1);
                ptr--;
                break;
            }
        }
        ptr++;
    }
    cout << str << endl;
    return 0;
}
```

8. 以下程序的输出结果是【8】

```
#include <iostream>
using namespace std;
class Base {
public:
    Base(int n, int m = 7) : x(n), y(n + m) { cout << x << "\t"; }
    void fun() { cout << y << "\t"; }
private:
    int x, y;
};
class A {
public:
    A(int n = 3) : k(n) { cout << k << "\t"; }
private:
    int k;
};
class Derived : public Base {
public:
    Derived(int n = 1, int m = 2) : a2(m), Base(n + m), z(n) {}
    void fun() { cout << z << "\t"; }
private:
    int z;
    A a1, a2;
};
void fun(Base& b) { b.fun(); }
int main()
{   Derived d(5);
    fun(d);
    return 0;
}
```

9. 以下程序的输出结果是【9】

```
#include <iostream>
using namespace std;
class A {
public:
    A(int x) : n(x) { cout << n << "\t"; }
    void f() { cout << n * n << "\t"; }
private:
    int n;
};
class B : virtual public A {
public:
    B(int x) : A(x) { cout << x << "\t"; }
};
class C : virtual public A {
public:
    C(int x) : A(x) { }
};
class D : public B, public C {
public:
    D(int x) : A(x + 1), B(x), C(x) {}
};
int main()
{   D d(5);
    d.f();
    return 0;
}
```

10. 以下程序的输出结果是【10】

```
#include <iostream>
using namespace std;
class Base {
public:
    Base(int n) : x(n) {}
    void f() { cout << x * x << "\t"; }
    virtual void g() { cout << x++ << "\t"; }
private:
    int x;
};
class Derived : public Base {
public:
    Derived(int n) : Base(n++), x(n) {}
    void f() { cout << x++ << "\t"; }
    void g() { cout << x * x << "\t"; }
private:
    int x;
};
int main()
{   Derived d(3);
    Base b = d, * bp = &d;
    b.g(); bp->f(); bp->g();
    return 0;
}
```

四、程序完善题（每空 2 分，共 20 分）

1. 以下程序用来计算两个大整数的相加结果。大整数的每个数字用数组来存放，下标 0 的数组元素存放该整数的个位数字，下标 1 的数组元素存放该整数的十位数字，如此类推。例如，第一个大整数的值为 258738，第二个大整数的值为 9260563，则程序输出两个整数相加的结果为 9519301。

```
#include <iostream>
using namespace std;
#define N 20
int main()
{
    int a[N] = { 8,3,7,8,5,2 };
    int b[N] = { 3,6,5,0,6,2,9 };
    int sum[N], i, carry = 0;
    for (i = 0; i < N; i++) {
        sum[i] = 【1】;
        if (sum[i] < 10)
            carry = 0;
        else {
            carry = sum[i] / 10;
            sum[i] = 【2】;
        }
    }
    while (--i)
        if (sum[i]) 【3】;
    for (; i >= 0; --i)
        cout << sum[i];
    cout << endl;
    return 0;
}
```

2. 以下程序的功能为：在字符串中找出最长的字母型子字符串。例如，字符串“Fruit \$12 Apple # Watermelon # Cherry”中最长的字母型子字符串为“Watermelon”。

```
#include <iostream>
using namespace std;
int main()
{
    char str1[] = "Fruit $12 Apple # Watermelon # Cherry", str2[32];
    char* p = str1, * maxp = str1, * p1 = str1, * p2 = str2;
    int n = 0, maxn = 0;
    while (*p1) {
        char c = 【4】;
        if (c >= 'a' && c <= 'z' || c >= 'A' && c <= 'Z') {
            if (【5】 >= maxn) {
                maxn = n; maxp = p;
            }
        }
        else {
            n = 0; p = 【6】;
        }
    }
}
```

```
while (maxn--)
    *p2++ = 【7】;
*p2++ = '\0';
cout << str2 << endl;
return 0;
}
```

3. 以下函数的功能为：在链表的中间结点后面插入一个结点。其中，用一个快指针和慢指针来查找链表的中间结点。每当快指针连续两步慢指针就走一步，当快指针走完时慢指针所指向的结点即为中间结点。

```
Node* insert(Node* list, Node* node)
{
    if (list == NULL) return node;
    Node* fast = list;
    Node* slow = 【8】;
    while (fast->next) {
        fast = fast->next;
        if (【9】)
            break;
        fast = fast->next;
        slow = slow->next;
    }
    【10】 = slow->next;
    slow->next = node;
    return list;
}
```

五、编程题（10 分）

定义一个字符串类 Str，将一个字符串的顺序颠倒后，并将从指定位置开始的所有数字字符拼接到另一个字符串的尾部。例如，字符串 1 为“abcdef”，字符串 2 为“A2G1%34&2D#”，将字符串 2 的顺序颠倒后为“#D2&43%1G2A”，如果指定从 5 个字符位置开始，将后面的数字字符“4312”拼接字符串 1 的尾部，即得到新的字符串“abcdef4312”。程序编写的具体要求如下：

- （1）私有成员
char* p: 用来存放字符串 1 和拼接后的新字符串。
- （2）公有成员
Str(char* s): 构造函数，根据参数字符串 s 的长度为成员变量 p 动态分配内存空间，并将 s 的所有字符拷贝至成员变量 p 中。
~Str(): 析构函数，释放动态分配的内存空间。
void operate (char* s, int n): 将参数字符串 s 的顺序颠倒后，并将从第 n 个字符开始的所有数字字符拼接成员变量 p 所指向的字符串之后，形成一个新的字符串。注意，必须为 p 重新分配内存空间。
void show(): 输出成员变量 p 的所有字符。
- （3）在主函数中首先定义字符串 1 和字符串 2，显示两个字符串的内容和字符串 2 的指定开始位置，然后用字符串 1 生成类 Str 的对象，再通过对象调用成员函数 operate()完成字符串处理功能，最后调用成员函数 show()输出处理后的新字符串。