

## 南开大学理工科《面向对象程序设计 2-1》

## 2010 级期末考试试卷 (A 卷) 2011 年 1 月

任课教师\_\_\_\_\_学院\_\_\_\_\_系别(班级)\_\_\_\_\_姓名\_\_\_\_\_学号\_\_\_\_\_

题 目	一	二	三	四	卷面成绩
分 数					

得 分

## 一、单项选择(每空 1 分, 共 20 分)

(注意选项填写到下面表格中!)

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

- 下面语句的输出结果为 ( a )。  

```
int n = 'a';
cout << n;
```

A) 97                      B) a                      C) 报错                      D) 65
- 在以下选项中, ( c ) 不是字符串常量。  

A) "string"                B) "str"                      C) '\n'                      D) "n"
- 已知: "int x=-10;" 则语句 "cout << (x%3);" 的输出结果为 ( a )。  

A) -1                      B) 1                          C) -2                          D) 2
- 下列标识符中, 合法的标识符为 ( a )。  

A) \_NO1                      B) main                      C) m3\$                      D) 3stu
- 判断字符变量 c 是数字或大写字母的逻辑表达式为 ( a )。  

A) c>='A' && c<='Z' || c>='0' && c<='9'  
 B) c<'A' || c>'Z' && c<'0' || c>'9'  
 C) c>='0' && c<='Z'  
 D) !(c>='A' && c<='Z' || c>='0' && c<='9')
- 已知: "int a[][1]={1,2,3};" 则 "cout << a[1];" 语句会在屏幕上输出 ( b )。  

A) 1 2 3                      B) 地址值                      C) 1                          D) 2
- 以下程序运行后的输出结果是 ( 0 )。  

```
#include <iostream>
using namespace std;
```

```

int main()
{
    int k=1;
    while(--k)
        k-=3;
    cout<<k;
    return 0;
}

```

A) 1                      B) 0                      C) -2                      D) 死循环

8. 下列语句正确的是 ( b )。

A) int a,\*p; p=a;                      B) int a[3],\*p; p=a;  
C) int a,\*p; a=p;                      D) int a[3],\*p; a=p;

9. 已知: “int a[]={1,2,3,4,5,6}; int\* p=a+2;” 则 “cout<<p[2]<<endl;” 的输出结果为 ( d )。

A) 2                      B) 3                      C) 4                      D) 5

● 10. 下列程序段执行后, x 的值为 ( b )。

```

x=0;
for(i=0;i<10;i++){
    if(i) x++;
}

```

A) 0                      B) 1                      C) 9                      D) 10

11. 在 32 位计算机中, 一个指向行的指针变量占用内存的字节数是 ( b )。

A) 1                      B) 4                      C) 8                      D) 由行长度决定

12. 已知: char str[] = "Hello"; 则 sizeof(str) 的值为 ( c )。

A) 4                      B) 5                      C) 6                      D) 7

13. 函数原型中, 可以不用给出的是 ( b )。

A) 函数参数的类型                      B) 函数的参数名  
C) 函数类型                      D) 函数名

14. 在一个被调用函数中, 关于 return 语句使用的描述, 错误的是 ( d )。

A) 被调用函数中可以不用 return 语句  
B) 被调用函数中可以使用多个 return 语句  
C) 被调用函数中, 如果有返回值, 就一定要有 return 语句  
D) 被调用函数中, 一条 return 语句可返回多个值给调用函数

15. 下面关于 const 指针的语句中, 正确的是 ( c )。

A) const int a=10; int b=5; const int \*p = &a; p=&b;  
B) const int a=10; int b=5; int \*const p=&a; p=&b;  
C) const int a=10; int b=5; const int \*p=&a; \*p=b;  
D) const int a=10; int b=5; int \*const p=&a; \*p=b;

16. 以下二维数组的初始化语句中, 正确的是 ( c )。

A) int a[3][2] = {{1,2},{2,3}};    B) int a[3][] = {1,2,3};  
C) int a[][2] = {1,2,3};            D) int a[][] = {1,2,3};

17. 已知 “int a=10, b=5, \*p; int \*&r=p; p=&a; \*r=15; p=&b; cout<<a;”, 则输出结果为 ( a )。

- A) 15                      B) 10                      C) 5                      D) 程序有错误

● 18. 已知 “char \*s=“123456”; cout<<\*s+2;”, 输出结果为 ( c )。

- A) 3                      B) 3456                      C) 51                      D) 52

19. 已知 “double \*p = new double(3);”, 若分配成功, 则指针 p 在堆中分配到的内存为 ( c ) 个字节。

- A) 1                      B) 3                      C) 8                      D) 24

20. 下列关于函数的描述正确的是 ( c )。

- A) 函数必须要有参数和返回值  
B) 函数体中只能含有一条 return 语句  
C) 函数通过 return 语句至多能够返回一个值  
D) 函数必须定义在先, 调用在后

得 分

## 二、填空题 (每空 1 分, 共 20 分)

(注意将结果填写到下面相应横线上!)

① \_\_\_\_\_ ② \_\_\_\_\_ ③ \_\_\_\_\_ ④ \_\_\_\_\_ ⑤ \_\_\_\_\_

⑥ \_\_\_\_\_ ⑦ \_\_\_\_\_ ⑧ \_\_\_\_\_ ⑨ \_\_\_\_\_ ⑩ \_\_\_\_\_

(11) \_\_\_\_\_ (12) \_\_\_\_\_ (13) \_\_\_\_\_ (14) \_\_\_\_\_ (15) \_\_\_\_\_

(16) \_\_\_\_\_ (17) \_\_\_\_\_ (18) \_\_\_\_\_ (19) \_\_\_\_\_ (20) \_\_\_\_\_

- 变量的引用就是变量的 ① 别名。
- 使用 new 分配的内存必须使用 delete ② 释放, 否则会造成内存泄露。
- 数制就是用一组固定的 ③ 数码 和一套统一的 ④ 规则 来表示数值的方法。
- 由高级程序设计语言编写的程序叫 源程序 ⑤, 它必须经过编译器翻译成计算机能够识别的目标程序, 再由连接器连接成可执行程序。
- 设 char c='a'; int x=2; double y=2.9; , 则表达式 y+c/x 的值是 ⑥ 50.9。
- 已知: char \*pc[5] = {"Beijing", "Tianjin", "Shanghai"}; 则使用 cout<<sizeof(pc); 语句将在屏幕上输出 ⑦ 20, 使用 cout<<sizeof(pc[0]); 语句将在屏幕上输出 ⑧ 4, 使用 cout<<strlen(pc[0]); 语句将在屏幕上输出 ⑨ 7。
- 局部变量存放在内存的 ⑩ 栈 区。
- 已知: int a[2][3], (\*p0)[3]=a, \*p1[2]={a[0], a[1]}, \*\*p2=p1+1, \*p3=&a[0][0]; 则通过 p0 访问元素 a[1][1] 应写为 (11) p0[1][1], 通过 p1 访问元

素  $a[1][1]$  应写为 (12)  $*(p1[1]+1)$  , 通过  $p2$  访问元素  $a[1][1]$  应写为 (13)  $*(p2+1)$  , 通过  $p3$  访问元素  $a[1][1]$  应写为 (14)  $*(3+4)$  。

9. 要使一个全局变量在声明该变量的源文件中可以访问, 但不能被其他源文件所访问, 则应将该变量声明为 静态全局 变量, 在声明该变量时应加上 static 关键字。

10. 常量 500.0 默认的数据类型是 double 。

11. 已知 “char  $a[10]$  = “abc”,  $*b$  = “def”;”, 则将  $b$  所指向的字符串内容拷贝到数组  $a$  中的语句为 strcpy(a,b) , 将  $b$  所指向的字符串内容添加到数组  $a$  尾部的语句为 (19) strcat(a,b) , 比较数组  $a$  中保存的字符串内容与  $b$  所指向的字符串内容是否相同的语句为 (20) strcmp(a,

b) 。

得 分

### 三、读程序写结果 (每空 3 分, 共 30 分)

(注意将结果写在下面相应横线上!)

① \_\_\_\_\_ ② \_\_\_\_\_ ③ \_\_\_\_\_ ④ \_\_\_\_\_ ⑤ \_\_\_\_\_

⑥ \_\_\_\_\_ ⑦ \_\_\_\_\_ ⑧ \_\_\_\_\_ ⑨ \_\_\_\_\_ ⑩ \_\_\_\_\_

1. 写出下面程序的运行结果。

```
#include <iostream>
using namespace std;
int main()
{
    int i;
    for (i=1; i<=100; i++)
        if(i%2==0 && i%3==0 && i%7==0)
            cout<<i<<<endl;
    return 0;
}
```

运行结果为: 42

②84

2. 写出下面程序的运行结果。

```
#include <iostream>
```

```
using namespace std;
int main()
{
    int i=1;
    while (i<=15)
    {
        i++;
        if (i%5!=2)
            continue;
        else if (i%3)
            continue;
        else cout<<i;
    }
    cout<<endl;
    return 0;
}
```

运行结果为: 12③

3. 写出下面程序的运行结果。

```
#include <iostream>
using namespace std;
int mul(int n)
{
    static int m=2;
    m *= n;
    return m;
}
int add(int n)
{
    static int m;
    m += n;
    return m;
}
int main()
{
    cout<<mul(3)<<endl;
    cout<<add(3)<<endl;
    return 0;
}
```

运行结果为: 6④  
3⑤

4. 写出下面程序的运行结果。

```
#include <iostream>
using namespace std;
void fun(int n, char ch)
{
    int i;
    for (i=0;i<n;i++)
        cout<<ch;
    cout<<endl;
}
int main()
{
    char c1='a', c2='d';
    fun(c2-c1, c1);
    return 0;
}
```

运行结果为: aaa⑥

5. 写出下面程序的运行结果。

```
#include <iostream>
using namespace std;
int main()
{
    int a[][4] = {1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4};
    int i=0, s=1;
    for (; i<4; i++)
        s *= a[i][(i+1)%3];
    cout<<s<<endl;
    return 0;
}
```

运行结果为: ⑦12

6. 写出下面程序的运行结果。

```
#include <iostream>
using namespace std;
int main()
{
    int a[][3] = {1,2,3,4,5,6,7,8,9,10,11,12};
    int (*p1)[3] = a+2;
    int *p2[4] = {a[0], a[1], a[2], a[3]};
    int **p3=p2+1;
    cout<<*(p1[1]+1)<<" "<<*(p2[2]+1)<<" "<<*(p3[1]+2)<<endl;
    return 0;
}
```

}

运行结果为: 11, 8, 9 ⑧

7. 写出下面程序的运行结果。

```
#include <iostream>
using namespace std;
int main()
{
    char a[3][10] = {"apple", "orange", "pear"};
    char *p = &a[1][1];
    cout<<a[2]<<" "<<a[0][2]<<endl;
    cout<<p<<" "<<*p<<endl;
    return 0;
}
```

运行结果为: pear p⑨  
Range r⑩

得 分

## 四、程序填空 (每空 2 分,共 30 分)

(注意将结果写在下面相应横线上!)

① \_\_\_\_\_ ② \_\_\_\_\_ ③ \_\_\_\_\_

④ \_\_\_\_\_ ⑤ \_\_\_\_\_

⑥ \_\_\_\_\_ ⑦ \_\_\_\_\_ ⑧ \_\_\_\_\_ ⑨ \_\_\_\_\_ ⑩ \_\_\_\_\_

(11) \_\_\_\_\_ (12) \_\_\_\_\_ (13) \_\_\_\_\_

(14) \_\_\_\_\_ (15) \_\_\_\_\_

1. 已知数列  $a_n = 2 \cdot n! + 3 \cdot (n+1)!$ , 函数 fun(n)的功能是输出数列的前 n 项。请填空。

```
#include <iostream>
using namespace std;
void f(int n)
{
    int a, b, c, i;
    a=b=1;
    for (i = 1; i <= n; i++)
```

```

    {
        b*=i①;
        a*=i+1②;
        c = 2*b+3*a;
        cout<<c<<" ";
    }
    cout<<endl;
}
int main()
{
    f(8);
    return 0;
}

```

2. 下面程序是利用冒泡排序法将数组中的数据按从小到大升序排序。请填空。

冒泡排序法的基本思路是：

第 1 轮：对于长度为  $n$  的一维数组。从前到后依次比较两个相邻的数，如果小的在前面，就保持不变；如果小的在后面，则交换两个相邻的数把小的交换到前面。经过第 1 轮的比较和交换，便把最大的数排到了最后一个位置。

第 2 轮：由于下标  $n-1$  的位置存储的是最大的数，因此，第 2 轮比较仅对数组中下标范围  $0 \sim n-2$  的前  $n-1$  个元素进行冒泡排序。

依次类推，可以完成后面的排序。如果在某轮比较中，没有发生任何数据交换，则说明当前数组元素已排序，不需再进行后续的比较。

```

#include <iostream>
using namespace std;
int main()
{
    int a[] = {5, 12, 23, 15, 9, 61, 98, 20, -21, -13};
    int i, j, temp, size;
    int swap=0 ③;
    size = sizeof(a)/sizeof(int);
    for (i = 0; i < size-1 && swap==0; i++)
    {
        swap = 1;
        for (j = 0; j < size-i-1; j++)
            if (a[j]>a[j+1] ④)
            {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
                swap=0;
            }
    }
}

```



```

    }
    for (i = 0; i < size; i++)
        cout<<a[i]<<" ";
    return 0;
}

```

3. 下面程序的功能是将两个字符串连接、形成一个新字符串，并将新字符串输出到屏幕上。请填空。

```

#include <iostream>
using namespace std;
int main()
{
    char *a = "abcde", *b = "fgh", *s=NULL;
    int n1=strlen(a), n2=strlen(b), i;
    _____ s= new char[n1+n2]; // 为 s 分配内存
    for (i=0; i<n1; i++)
        s[i] = a[i];
    for (i=0; i<=n2; i++)
        _____ s[n1+i] = b[i];
    cout<<s<<endl;
    _____ delete []s; // 释放 s 所指向的内存
    return 0;
}

```

4. 对二维数组 a[4][4]所表示的矩阵，输出对角线元素（即输出 10、21、32、43）。请填空。

```

#include <iostream>
using namespace std;
int main()
{
    double a[4][4]={10,11,12,13},{20,21,22,23},{30,31,32,33},{40,41,42,43}};
    int i;
    for (i=0; i<4; i++)
        cout<< _____ a[i][i] ⑧ <<endl;
    return 0;
}

```

5. 将用户输入的字符串前 4 个字符（不足 4 个则循环取）取出生成一个新字符串，并将其输出到屏幕上。如：输入“ABCDEF”，则输出“ABCD”；输入“ABC”，则输出“ABCA”；输入“AB”，则输出“ABAB”；输入“A”，则输出“AAAA”。请填空。

```

#include<iostream>
using namespace std;
int main()
{
    char a[20], b[20];

```

```

int n, i;
cout<<"请输入一个字符串: ";
cin>>a;
n = strlen(a);
for (i=0; i<4; i++)
    b[i] = ⑨ a[i%n];
b[i]='\0';
cout<<b<<endl;
return 0;
}

```

6. 下面程序的功能是用户从键盘上输入两个操作数和一个运算符 (+或-), 根据运算符进行相应的运算并输出计算结果。例如, 输入 9+3, 输出 9+3=12。如果输入的不是加法或减法运算符, 则输出 “不支持该运算”。请填空。

```

#include <iostream>
using namespace std;
int main()
{
    double x,y;
    char ch;
    cout<<"请输入算式 (格式:3.5+2.3、3.5-2.3): "<<endl;
    cin>>⑩x>>ch>>y;
    switch(ch) ⑪
    {
        case '+': cout<<x<<'+<<y<<='<<x+y<<endl; break;
        case '-': cout<<x<<'-<<y<<='<<x-y<<endl; break;
        default: ⑫ cout<<"不支持该运算"<<endl;
    }
    return 0;
}

```

7. 下面程序的功能是判断用户输入的数是否为素数, 若是素数则输出, 否则不输出。请填空。

```

#include <iostream>
using namespace std;
int main()
{
    int n, i;
    cin>>n;
    for (i=2;; i++) if (n%i == 0) break;
    if (⑬ i==n) cout<<n<<"是素数"<<endl;
    return 0;
}

```

8. 下面是求一个数的绝对值的函数。请填空。

```
(14) double abs(double x)
{double y;
  y= x>=0?x:-x (15);
  return y;
}
```