

中山大学

二〇〇五年港澳台人士攻读博士学位研究生入学考试试题

科目代码: 676

科目名称: 高级软件设计

考试时间: 4 月 16 日 下午

考生须知

全部答案一律写在答题纸上, 答在试题纸上的不得分!
答题要写清题号, 不必抄题。

一、填空题 (25 小题, 共 50 分) 按顺序把答案写在答题纸上, 注明题号或序号。
(以下每小题 1 分, 共 10 分)

1. 在 C++ 中, 用数组、指针、和 _____ 作为函数参数, 能够将参数值带回。
2. 在 C++ 中, 虽然友元提供了类之间数据进行访问的一种方式, 但它破坏了面向对象程序设计的 _____ 特性。
3. 在 C++ 中, 构造派生类对象时, 总是先从 _____ 的初始化开始的。
4. 拷贝构造函数是在用一个对象初始化另一个对象时被调用, 系统缺省的拷贝构造函数的工作方法是 _____。
5. 类是用用户定义的类型, 具有类类型的变量称作 _____。
6. 在 5 个运算符“*(乘号)”, “<”, “!”, “&&”中, 优先级由高到低的顺序是 _____。

7. 在 C++ 中, 类定义一般用 class 关键字, 不过用 struct 关键字也可以定义类, 它们定义类的区别在于 _____。所以, 它们只能访问静态的数据成员。

8. 静态的成员函数没有隐含的 _____。

9. 与“(a==0)”等价的 C++ 表达式是 _____。

10. 若 a=6, b=4, c=2, 则表达式“!(a-b)+c-1&&b+c/2”的值是 _____。
(以下每小题 2 分, 共 20 分)

11. 在面向对象的设计中, 通过 _____ 实现数据隐藏; 通过 _____ 实现代码的复用。

12. 在下面的程序段中, 语句“const int* c=&a;”和语句“int* const d=&b;”的含义分别是 _____。

```
const int a=78;
int b=28;
const int* c=&a;
int* const d=&b;
```

13. 用 new 申请有 10 个元素的指针数组 str, 假定数组元素是指向字符型数据的指针, 该 C++ 语句为 _____。

14. 在 C++ 中构造一个对象时, 其数据成员在构造函数中初始化。对于内嵌的对象、_____、_____数据成员需要在构造函数中初始化列表表中初始化。

15. 在类的定义中, 说明为 protected 的数据成员称为保护成员。保护数据成员在类的定义中, 说明为 protected 的数据成员称为保护成员。保护数据成员在类的定义中, 说明为 protected 的数据成员称为保护成员。保护数据成员在类的定义中, 说明为 protected 的数据成员称为保护成员。

考试完毕, 试题和草稿纸随答题纸一起交回。

员具有双重作用: 对于其派生类而言, _____; 而对于其外部的程序而言, _____。

16. C++ 中有两种数据类型: _____ 和 _____ 可以使用 signed 修饰符。

17. 举出 C++ 中两种流程控制语句 _____ 和 _____。

18. 预编译指令由三种, 它们是: 文件包含、_____ 和 _____。

19. C++ 中, 对象保存在内存中, _____ 内存是自动分配和释放的, 而 _____ 内存需要用户自己申请和释放。

20. 在 C++ 函数中, 可用 return 语句带回一个值。如果有多个返回值, 可用 _____ 等带回。

(以下每小题 4 分, 共 20 分)

21. 下面程序的运行结果是 _____。

```
#include <stdio.h>
void main()
{
    char a='0', b='9';
    float x;
    x=(b-a)/(F-'B');
    printf("%d\n", (int)(3.14*x));
}
```

22. 以下程序的输出结果是 _____。

```
#include <stdio.h>
void main()
{
    int i;
    for (i=0; i<=5; i++) {
        if (i%2) printf("***");
        else continue;
        printf("#");
    }
    printf("$\n");
}
```

23. 下面程序的运行结果是 _____。

```
#include <iostream.h>
#define N 100
class CStack
{
public:
    CStack() {top=0; cout<<"Hello ";}
    ~CStack() {cout<<"Bye";}
    void push(int i);
    int pop();
private:
    int stack[N];
    int top;
    void CStack::push(int i)
    {
        if (top==N)
            cout<<"Overflow";
        return;
    }
};
```

```

}
else{
    top++;
    stack[top]=i;
}
}
int CStack::pop()
{
    int temp;
    if (top==0){
        cout<<"Underflow";
        return 0;
    }
    else{
        temp=stack[top];
        top--;
        return temp;
    }
}
void main()
{
    CStack *ptr=new CStack;
    ptr->push (10);
    ptr->push(50);
    cout <<ptr->pop()<<" ";
    cout << "OK!"<<endl;
}
}

```

24. 下面程序的运行结果是_____。

```

#include <iostream.h>
class B
{
public:
    B(){cout<<"B::B0 construction "<<endl;}
    virtual ~B(){cout<<"B::~B0 destruction "<<endl;}};
class D:public B
{
public:
    D(int i){cout<<"D::D0 construction "<<endl;
        buf=new char[i];
    }
    virtual ~D()
    {
        delete []buf;
        cout<<"D::~D0 destruction "<<endl;
    }
private:
    char *buf;
};
void fun(B *a)
{
    delete a;
}
void main()
{

```

```

B *a=new D(25);
fun(a);
}
}
25. 下面程序的运行结果是_____。
#include <stdio.h>
int a[]={2,5,6,8,10};
int *p[]={a,a+1,a+2,a+3,a+4};
void main()
{
    printf("%d\\t%d\\t%d\\n",a[4],*(a+2),*p[1]);
    printf("%d\\t%d\\t%d\\n",*(p+1)+a[2],*(p+4)-*(p+0),*(a+3)%a[4]);
}

```

二、问答题 (每小题 5 分, 共 20 分)

1. 在定义拷贝构造函数时, 为什么通常还要定义一个重载的赋值运算符?
2. 简述局部作用域、全局作用域和类作用域的异同。
3. 虚函数是否一定要有 virtual 关键字? 什么叫纯虚函数和抽象类? 多态调用需要满足怎样的条件?
4. 数组和链表是两种重要的线性数据结构, 使用链表有什么优点?

三、找出下面程序 (或程序段) 中的语法错误, 并予以纠正 (每小题 4 分, 共 8 分)

(1) 程序功能是将某年某月的几号转换成这一年的第多少天。

```

#include <stdio.h>
struct date {int y; int m; int d;}
int daytab[2][12]={{0,31,28,31,30,31,30,31,31,30,31,30,31},
{0,31,29,31,30,31,30,31,31,30,31,30,31}};
void main()
{
    int n;
    struct date d;
    int days (struct date *pd);
    printf("Input: Year=? Month=? Day=? \n");
    scanf("%d%d%d",&y,m,d);
    n=days(&d);
    printf("N=%d days\n",n);
}
int days (struct date *pd)
{
    int i,day,leap;
    day=pd.d;
    leap=pd.y%4==0&&pd.y%100!=0 || pd.y%400==0;
    for(i=1;i<pd->m;++i)
        day+=daytab[leap][i];
    return(day);
}

```

(2) 程序功能是打印 Object 类型变量的分量 a。

```
#include <iostream.h>
struct Object{int a;int b;};
void main()
{
    Object& MyFunction(int a=20,int b);
    Object& rMyObj = MyFunction(5);
    cout << "rMyObj.a = " << rMyObj.a << endl;
    delete &rMyObj;
}
Object& MyFunction(int a=20,int b)
{
    Object* o = new Object;
    o->a = a;
    o->b = b;
    return o;
}
```

四、(8分) 下列 shape 类是一个表示形状的抽象类，area() 为求图形面积的函数，total() 则是一个通用的用以求不同形状的图形面积总和的函数。请从 shape 类派生梯形类 (trapezoid)、圆形类 (circle)，并给出具体的求面积函数

```
class shape{
public:
    virtual float area()=0
};

float total(shape *s[],int n)
{
    float sum=0.0;
    for(int i=0;i<n;i++)
        sum+=s[i]->area();
    return sum;
}

class Circle:public Shape
{
public:
    Circle(double r){radius=r;}
    double Area()const{return 3.1416*radius*radius;}
private:
    double radius;
};

class Trapezoid:public Shape
{
public:
    Trapezoid(double top,double bottom,double high)
    {T=top;B=bottom;H=high;}
    double Area()const{return (T+B)*H*0.5;}
private:
    double T,B,H;
};
```

五、(6分) 函数 binary() 实现折半查找，即查寻给定的单词 word 是否在关

键字表 tab 中 (关键字按字典顺序排列)，折半查找每次把 word 与 tab 表中相应部分的位于中间位置的关键字进行比较，最终结果：或者与某个关键字相同，或者与所有关键字都不相同。

```
#include <string.h>
struct key{char *keyword; int count;};
struct key *binary(char *word,struct key tab[],int n)
//word: a searching word
//tab: keyword table
//n: the sum of keywords
{
    int cond;
    struct key *low=tab;
    struct key *high=&tab[n-1];
    struct key *mid;
    while(low<=high){
        mid=
        if((cond=strcmp(word,mid->keyword))<0)
            high=mid-1;
        else if (cond>0)
            ;
        else return mid;
    }
    return(NULL);
}
```

六、(8分) 下面程序实现汉诺塔游戏。规则是：三个立柱 (分别为 A、B、C)，开始 A 上串有 n 个 (用户输入值) 大小不等的圆盘，大的在下，小的在上。要求借助于 B 把它们从 A 移到 C。每次只能移一个盘，而且三个柱上的盘总是大的在下，小的在上

```
#include <stdio.h>
int i=0;
void main()
{
    int n;
    void movetower(int m,char from, char to,char usg);
    for(;;){
        printf("input the number of disks of hanoi tower:");
        scanf("%d",&n);
        if (n==0)
            printf("n\n");
        printf("The moving step is as below:\n");
        movetower(n,'A','C','B');
        printf("Total:%d\n",i);
    }
}

void movetower (int n,char from, char to,char usg)
{
    void movetodisk(char source,char destination);
    if (
        movetower(n-1,from,usg,to);
        movetower(n-1,
    );
}
```

```
else movedisk(from,to);  
}  
void movedisk(char source,char destination)  
{  
    i++;  
    printf("%c----->%c\n",source,destination);  
}
```