

清华大学计算机系 C++ 期末考试题及答案

一、 填空题 (25 小题, 共 50 分)

(以下每小题 1 分, 共 10 分)

1. 在 C++ 中, 函数的参数有两种传递方式, 它们分别是值传递和 **地址或指针或引用传递**。
2. 当一个成员函数被调用时, 该成员函数的 **this 指针** 指向调用它的对象。
3. 在基类和派生类中, 派生类可以定义其基类中不具备的数据和操作。对两个有相同名字的数据成员进行访问时, 如果没有 **作用域分隔符限定时**, 对此数据成员的访问将出现歧义。
4. 拷贝构造函数使用 **引用** 作为参数初始化创建中的对象。
5. 在公有继承的情况下, 基类数据成员在派生类中的访问权限 **保持不变**。
6. 描述命题“A 小于 B 或小于 C”的表达式为 **$A < B \mid A < C$** 。
7. 用 new 申请某一个类的动态对象数组时, 在该类中必须能够匹配到 **没有形参的或缺省参数** 构造函数, 否则应用程序会产生一个编译错误。
8. 静态数据成员在类外进行初始化, 且静态数据成员的一个拷贝被类的所有对象 **共享**。
9. 为了避免可能出现的歧义, C++ 对 if...else 语句配对规则规定为: else 总是与 **与最近的 if** 配对。
10. 设“int a=3, b=4, c=5;”, 表达式“(a+b) > c && b == c”的值是 **0**。

(以下每小题 2 分, 共 20 分)

11. 面向对象的程序设计有四大特征, 它们是**抽象、封装、继承、多态**。
12. 在 Visual C++ 中, 定义重载函数时, 应至少使重载函数的参数个数或参数类型 **不同**; 在基类和派生类中, 成员函数的覆盖是指 **派生类成员函数与在基类被覆盖的成员函数名、参数个数、参数类型和返回值类型均相同**。
13. 构造函数与析构函数除功能不同外, 在定义形式上, 它们的区别还包括构造函数名与类名相同, 而析构函数名是在类名前加一个~、**析构函数没有参数、析构函数可以定义为虚函数**。
14. 动态联编要满足两个条件, 它们是 **被调用的成员函数是虚函数、用指针或引用调用虚函数**。
15. 在 C++ 类中, 有一种不能定义对象的类, 这样的类只能被继承, 称之为 **抽象类**, 定义该类至少具有一个 **纯虚函数**。
16. 在 C++ 类中, const 关键字可以修饰对象和成员函数, const 对象不能 **被修改**, const 成员函数不能 **修改类数据成员**。
17. 举出 C++ 中两种用户自定义的数据类型: **类、枚举**。
18. C++ 中没有字符串类型, 字符串是通过 **字符数组** 来表示的, 每一个字符串都有一个结尾字符 **\0**。
19. C++ 中没有输入输出语句, 输入输出是通过 **输入输出库** 实现的, 写出一条打印整型

变量 n 的输出语句: `cout<<n;`

20. 举出 C++中两种代码复用的方式: `继承`、`复用`。

(以下每小题 4 分, 共 20 分)

21. 下面程序的运行结果是 `3`。

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

22. 下面程序的运行结果是 `2 5 8 11 14`。

```
#include "iostream.h"
void main( )
{
    int i=1;

    while (i<=15){
        i++;
        if (i%3!=2) continue;
        else cout <<"i="<<i<<endl;
    }
}
```

23. 下面程序的运行结果是_____。

```
#include "iostream.h"
class test
{
private:
    int num;
    float fl;
public:
    test( );
    int getint( ){return num;}
    float getfloat( ){return fl;}
```

1.单选: 10*2

2.简答: 4*5

3.程序分析: 5*8

4.程序设计: 2*10

考点:

(1) 运算符重载

(2) 文件 IO 流

(3) 异常处理

```

    ~test( );
};

test::test( )
{
    cout << "Initalizing default" << endl;
    num=0;fl=0.0;
}

test::~~test( )
{
    cout << "Desdtructor is active" << endl;
}

void main( )
{
    test array[2];
    cout << array[1].getint( )<< " " << array[1].getfloat( ) <<endl;
}

```

```

Initalizing default
Initalizing default
0 0
Desdtructor is active
Desdtructor is active

```

24. 下面程序的运行结果是_____。

```

#include <iostream.h>

class A
{
public:
    A() {cout<<"A::A()  called.\n";}
    virtual ~A() {cout<<"A::~~A()  called.\n";}
};

class B:public A
{
public:
    B(int i){
        cout<<"B::B()  called.\n";
    }
};

```

```

    buf=new char[i];}
    virtual ~B()
    {
        delete []buf;
        cout<<"B::~B() called.\n";
    }
private:
    char *buf;
};

```

```

void fun(A *a)
{
    delete a;
}

```

```

void main()
{
    A *a=new B(15);
    fun(a);
}

```

A::A() called.

B::B() called.

B::~B() called.

A::~A() called.

25. 下面程序的运行结果是_____。

```

#include <stdio.h>
int a[ ]={1,3,5,7,9};
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]);
}

```

9 5 3

8 4 7

二、 问答题（每小题 5 分，共 20 分）

1. 若程序员没有定义拷贝构造函数，则编译器自动生成一个缺省的拷贝构造函数，它可能会产生什么问题？

解答要点：当对象含有指针数据成员，并用它初始化同类型的另一个对象时，缺省的拷贝构造函数只能将该对象的数据成员复制给另一个对象，而不能将该对象中指针所指向的内存单元也复制过去。这样，就可能出现同一内存单元释放两次，导致程序运行出错。

2. 简述成员函数、全局函数和友元函数的差别。

解答要点：以下几点必须说清楚：

成员函数是在类内部定义的，作用域在类的内部，成员函数可以访问类的数据成员（公有、保护和私有数据成员），可以调用该类的其它成员函数（公有、保护和私有成员函数），可以调用全局函数。如果友元函数是另一个类的公有成员函数，则该类的成员函数也只能通过那个类的对象调用，不能调用那个类的保护和私有成员函数。非本类成员函数（其它类成员函数或全局函数）可以通过该类的对象访问该类的公有数据成员和调用该类的的公有成员函数。

不是在类中定义的成员函数都是全局函数。

如果某一个函数（全局函数或类的成员函数）定义为另一个类的友元函数，需要在那个类中用 friend 关键字声明，友元函数并不是类的成员，它的定义自然是在那个类的外面。

3. 简述结构化的程序设计、面向对象的程序设计的基本思想。

解答要点：结构化的程序设计将数据和对数据的操作分离，程序是由一个个的函数组成的，面向对象的程序设计将数据和操作封装在一起，程序是由一个个对象组成的，对象之间通过接口进行通信，它能够较好地支持程序代码的复用。

4. 结构 struct 和类 class 有什么异同？

解答要点：struct 和 class 都可以定义类，但是缺省访问权限说明时，struct 的成员是公有的，而 class 的成员是私有的。在 C++ 中，struct 可被 class 代替。

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) 程序功能是倒序输出各给定的字符串。

```
#include <stdio.h>
void main()
{
    char str[5][ ]={"First","Second","Third","Forth","Fifth"};
    char *cp[ ]={str[4],str[3],str[2],str[1],str[0]};
    int i;

    while(i<=5)
    {
        printf("%c ",*(cp+i));
```

```

        i++;
    }
}

```

- ① "char str[5][]={"First","Second","Third","Forth","Fifth"};"应为
"char str[5][10]={"First","Second","Third","Forth","Fifth"};"
- ② "while(i<=5)"应为"while(i<5)"
- ③ "printf("%c ",*(cp+i));"应为"printf("%s",*(cp+i));"
- ④ "int i;"应为"int i=0;"

(2) 程序功能是将各个平方根值放入数组中。

```

#include <stdio.h>
void main()
{
    int max,a,i;
    scanf("%d%d",max,a);
    double x[max];

    for (i=0;i<max;i++)
        x[i]=sqrt(a*i);
}

```

- ① 增加"#include <math.h>"
- ② "scanf("%d%d",max,a);"应为"scanf("%d%d",&max,&a);"
- ③ "double x[max];"改为:
"double *x=new double[max];"
...
"delete []x;"

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 类派生三角形类 (triangle)、矩形类 (rectangle)，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 Triangle:public Shape
{
public:
    Triangle(double h,double w) {H=h;W=w;}
    double Area() const{return H*W*0.5;}
private:
    double H,W;
};

class Rectangle:public Shape
{
public:
    Rectangle(double h,double w) {H=h;W=w;}
    double Area()const{return H*W;}
private:
    double H,W;
};

```

五、（6分）完成顺序查找函数 f_seq()。其过程是：从表头开始，根据给定的模式，逐项与表中元素比较。如果找到所需元素，则查找成功，并打印出它在表中的顺序号。如果查找整个表仍未找到所需对象，则查找失败

```

#include <stdio.h>
void f_seq(char *list[],char *object,int len)
//list 指针数组，指向字符串
//object 模式串
//len 表的长度
{
    char **p;
    int strcmp(char *s,char *t);
    p=list;
    while (____①____) //p<list+len
        if (strcmp(*p,object)==0)
            break;
        else ____②____; //p++
}

```

```

    if (p<list+len)
        printf( "Success! **% d\n",p-list);
    else printf("Unsuccess!\n");
}
int strcmp(char *s,char *t)
{
    for (;*s==*t; s++,t++)
        if (*s=='\0')
            return(0);
        return(____③____); //s-t 或*s-*t 或 1
}

```

六、（8 分）完成使链表逆置函数 reverse，若有链表：

链表结点的结构如下：

```

struct node
{
    int num;
    struct node *next;
}

struct node* reverse(struct node *head)
//head 链表头结点
{
    struct node *p,*temp1,*temp2;
    if(head==NULL____①____) return head; //| head->next==NULL
    p=head->next;head->next=NULL;
    while(____②____) //p!=NULL 或 p
    {
        temp1=head;
        ____③____; //head=p;
        temp2=p;
        p=p->next;
        ____④____; //temp2->next=temp1;或 head->next=temp1;
    }//Match while statement
    return head; //返回逆置后的链表的头结点
}

```


一、选择填空题（共 20 小题，每小题 2 分，共 40 分）。

1. 关于 C++与 C 语言关系的描述中，（ 4 ）是错误的。
 - a.C 语言是 C++语言的一个子集
 - b.C 语言与 C++语言是兼容的
 - c.C++语言对 C 语言进行了一些改进
 - d.C++语言和 C 语言都是面向对象的
2. 按照标识符的要求，（ 1 ）符号不能组成标识符。
 - a.连接符
 - b.下划线
 - c.大小写字母
 - d.数字字符
3. 为了避免嵌套的 if-else 语句的二义性，C 语言规定 else 总是与（ 3 ）组成配对关系。
 - a.缩排位置相同的 if
 - b.在其之前未配对的 if
 - c.在其之前未配对的最近的 if
 - d.同一行上的 if
4. 在"int a[][3]={ {1},{3,2},{4,5,6},{0}};"中，a[2][2]的值是（ 3 ）。
 - a.1
 - b.0
 - c.6
 - d.2
5. 设"char **s; "，以下正确的表达式是（ 2 ）。
 - a.s="computer";
 - b.*s="computer";
 - c.**s="computer";
 - d.*s='c';
6. 对于"int *pa[5]; "的描述中，（ 4 ）是正确的。
 - a.pa 是一个指向数组的指针，所指向的数组是 5 个 int 型元素
 - b.pa 是一个指向某数组中第 5 个元素的指针，该元素是 int 型变量
 - c.pa [5]表示某个元素的第 5 个元素的值
 - d.pa 是一个具有 5 个元素的指针数组，每个元素是一个 int 型指针
7. 在下列表示引用的方法中，（ 1 ）是正确的。

已知：int m=10;

 - a.int &x=m;
 - b.int &y=10;
 - c.int &z;
 - d.float &t=&m;
8. 下列 for 循环的次数为（ 2 ）。

for (i=0, x=0; !x&& i<=5; i++)

 - a.5
 - b.6
 - c.1
 - d.无限
9. 对于 C/C++语言的函数，下列叙述中正确的是（ 1 ）。

- a.函数的定义不能嵌套，但函数调用可以嵌套
- b.函数的定义可以嵌套，但函数调用不能嵌套
- c.函数的定义和调用都不能嵌套
- d.函数的定义和调用都可以嵌套

10. 在一个被调用函数中，关于 `return` 语句使用的描述，（ 4 ）是错误的。

- 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，一个 `return` 语句可以返回多个值给调用函数

11. 在一个函数中，要求通过函数来实现一种不太复杂的功能，并且要求加快执行速度，选用（ 1 ）。

- a.内联函数
- b.重载函数
- c.递归调用
- d.嵌套调用

12. 使用 `fseek` 函数可以实现的操作是（ 1 ）。

- a.改变文件指针的当前位置
- b.文件的顺序读写
- c.文件的随机读写
- d.以上都不对

13. 下列存储标识符中，（ 3 ）的可见性与存在性不一致。

- a.外部类
- b.自动类
- c.内部静态类
- d.寄存器类

14. 在如下结构定义中，不正确的是（ 2 ）。

a.struct student
 {int no;
 char name[10];
 float score;
 };

b.struct stud[20]
 {
 int no;
 char name[10];
 float score;
 };

c.struct student
 {
 int no;
 char name[10];
 float score;
 } stud[20];

d.struct

```

{
    int no;
    char name[10];
    float score;
}stud[100];

```

15. 将两个字符串连接起来组成一个字符串时，选用（3）函数。

- a.strlen()
- b.strcpy()
- c.strcat()
- d strcmp()

16. (4)不是构造函数的特征

- 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.构造函数必须指定类型说明

17. 已知：类 A 中一个成员函数说明如下：

```
void Set(A&a);
```

其中，A&的含义是(3)。

- a.指向类 A 的指针为 a
- b.将 a 的地址值赋给变量 Set
- c.a 是类 A 对象的引用，用来作函数 Set() 的参数
- d.变量 A 与 a 按位与作为函数 Set() 的参数

18. 已知：print()函数是一个类的常成员函数，它无返回值，下列表示中，（1）是正确的。

- a.void print() const;
- b.const void print();
- c.void const print();
- d.void print(const);

19. 关于虚函数的描述中，（3）是正确的。

- a.虚函数是一个 static 类型的成员函数
- b.虚函数是一个非成员函数
- c.基类中说明了虚函数后，派生类中将其对应的函数可不说明为虚函数
- d.派生类的虚函数与基类的虚函数具有不同的参数个数和类型

20. 关于 new 运算符的下列描述中，（4）是错的。

- a.它可以用来动态创建对象和对象数组
- b.使用它创建的对象和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初始值

二、问答题（15 分）

- 1、（3 分）虚析构函数有什么作用？
- 2、（3 分）拷贝构造函数在哪几种情况下调用？
- 3、（4 分）函数重载与函数覆盖有什么不同，它们与多态有什么关系？
- 4、（3 分）C++继承是如何工作的？
- 5、（2 分）类与对象有什么区别？

三、（6 分）分析下面程序的运行结果

```
#include <iostream.h>
class B
{
public:
    B(){}
    B(int i){b=i;}
    virtual void virfun()
    {
        cout<<"B::virfun() called.\n";
    }
private:
    int b;
};
class D:public B
{
public:
    D(){}
    D(int i,int j):B(i){d=j;}
private:
    int d;
    void virfun()
    {
        cout<<"D::virfun() called.\n";
    }
};
void fun(B *obj)
{
    obj->virfun();
}
void main()
{
    D *pd=new D;
    fun(pd);
}
```

四、（9 分）下面的程序可以统计命令行第一个参数中出现的字母个数，请填充下面空白，完成程序。

```
#include <stdio.h>
#include <ctype.h>
void main(int argc,  argv[;])
{
    char *str;
    int count=0;
    if(argc<2)exit(1);
```

```

        str=  ;
while(*str)
    if(isalpha( )) count++;
printf("\n 字母个数: %d\n",count);
}

```

提示: int isalpha(int ch)函数功能是检查 ch 是否是字母

五、(8 分) 定义一个字符栈类 Stack (包括类的实现)。数据成员包括一个存放字符的数组 stck[] 和一个栈指针 tos。栈数组的尺寸由常量 SIZE 确定。栈的基本操作为 Push () 和 Pop ()。

六、(10 分)完成下面的函数, 对有 n 个元素的数组 a, 使数组元素按逆序排列。

```

void inverse(int *a, int n)
{
}

```

七、(12 分) 下面的函数统计子字符串 substr 在字符串 str 中出现的次数, 如果 substr 在 str 中不出现, 则返回值 0。请完成该函数。

```

int str_count(char *substr, char *str)
{
}

```

一、选择填空题

- 1、 D 2、 A 3、 C 4、 C 5、 B 6、 D 7、 A 8、 B 9、 A 10、 D
11、 A 12、 A 13、 C 14、 B 15、 C 16、 D 17、 C 18、 A 19、 C 20、 D

二、问答题

1、 虚析构函数有什么作用?

解答要点:

对象销毁时, 需要调用析构函数。在多态调用时, 是用基类的指针访问派生类的对象。如果析构函数是非虚函数, 则基类指针只能访问基类的析构函数, 而不能访问派生类的析构函数, 导致派生类对象销毁时, 没有调用派生类的析构函数, 只是调用了基类的析构函数。如果把析构函数定义成虚函数, 则可克服这个问题。

2、 拷贝构造函数在哪几种情况下调用?

解答要点:

用一个对象初始化另一个对象时

当用对象作为函数参数传递时

当函数返回对象时

3、 函数重载与函数覆盖有什么不同, 它们与多态有什么关系?

解答要点:

函数重载是指函数名相同, 而函数的参数个数或类型不同; 覆盖是指在派生类中成员函数与基类成员函数的函数名、参数个数、类型与返回值均相同; C++中正是通过虚函数的覆盖, 实现多态的功能。

4、 C++继承是如何工作的？

解答要点：

继承使得派生类能够使用基类的公有和保护成员，从而实现代码的复用，派生类可以增加成员，也可以隐藏和覆盖基类的成员。对于公有继承，基类成员的访问权限在派生类保持不变。

5、 类与对象有什么区别？

解答要点：

类是类型，是对象的抽象，对象是类的具体实例。一个类可以有多个对象，每个对象都有自己的存储单元，而类不占存储单元。

三、运行结果为： D::virfun() called.

四、填空

① char * ② argv[1] ③ *str++

五、（8分） 定义一个字符栈类 Stack（包括类的实现）。数据成员包括一个存放字符的数组 `stck[]` 和一个栈指针 `tos`。栈数组的尺寸由常量 `SIZE` 确定。栈的基本操作为 `Push（）` 和 `Pop（）`。

```
const int SIZE=27;
class Stack
{
public:
    Stack():tos(0){};
    void Push(char ch);
    char Pop();
private:
    char stck[SIZE];
    int tos;
};

void Stack::Push(char ch)
{
    if(tos==SIZE)
        cout<<"\nStack is full\n";
    else{
        stck[tos]=ch;tos++;
    }
}

char Stack::Pop()
{
    if(tos==0){
        cout<<"\nStack is empty\n";
```

```

        return 0;
    }
    tos--;
    return stek[tos];
}

```

六、(10 分)完成下面的函数，对有 n 个元素的数组 a，使数组元素按逆序排列。

```

void inverse(int *a, int n)
{
}

```

```

int i,*p;
p=new int[n];

for(i=0;i<=n-1;i++)
    p[i]=a[i];
for(i=0;i<=n-1;i++)
    a[i]=p[n-i-1];
delete []p;

```

七、(12 分) 下面的函数统计子字符串 substr 在字符串 str 中出现的次数，如果 substr 在 str 中不出现，则返回值 0。请完成该函数。

```

int str_count(char *substr, char *str)
{
}

```

```

int count=0;
char *pChar;

if(substr==NULL||str==NULL) return count;
while(*str!='\0'){
    pChar=substr;
    while(*pChar==*str){
        pChar++;
        if(*pChar=='\0'){
            count++;break;
        }
        else str++;
    }//Match while(*pCh...) statement
    str++;
} //Match while(*str...) statement
return count;

```