

共:【5大题】 【50小题】 【满分100.0分】 考试时间: 【120分钟】

题型	单选题	多选题	填空题	判断题	编程题	实训题	简答题	总分
应得分	25.0	10.0		15.0		30.0	20.0	100
实得分	25.0	6.0		15.0		30.0	20.0	96
评卷人								

一、单选题 (共25题; 共25.0分)

1.下列关于内联函数描述正确的是() (1.0分)

- ☐ 内联函数必须通过关键字inline来定义
- ☒ 内联函数是通过编译器来实现的
- ☐ 内联函数函数体的后一条语句必须是return语句
- ☐ 内联函数在调用时会发生控制转移

2.下列哪种情况适宜采用inline定义函数: (1.0分)

- ☒ 函数代码少、频繁调用
- ☐ 函数体含有循环语句
- ☐ 函数代码多、不常调用
- ☐ 函数体含有递归语句

3.C++中所有在函数中定义的变量以及形式参数,都属于: (1.0分)

- ☐ 静态变量
- ☐ 全局变量
- ☐ 函数
- ☒ 局部变量

4.下面概念中,不属于面向对象方法的是() (1.0分)

- ☐ 类
- ☐ 对象
- ☒ 过程开发
- ☐ 继承

5.C++从C的面向过程变成为面向对象的主要原因是() (1.0分)

- ☐ 规定函数说明符必须用原型
- ☐ 增加了一些新的运算符
- ☒ 引进了类和对象的概念
- ☐ 允许函数重载, 并允许设置缺省参数

6.以下程序段中循环体的执行次数是() (1.0分)

```
1. int i,j;  
2. for (i=0,j=1; i<=j+1; i+=2, j--)  
3.     cout << i << endl;
```

- ☒ 1
- ☐ 3
- ☐ 0
- ☐ 2

7.以下程序段执行后sum的值是 () (1.0分)

```
1. int i,sum;  
2. for(i=1;i<16;i++)  
3.     sum+=i;  
4. cout<< sum ;
```

- ☐ 0
- ☐ 15
- ☒ 不确定
- ☐ 14

8.在类定义的外部, 可以被访问的成员有 (1.0分)

- ☒ public的类成员
- ☐ 所有类成员
- ☐ public或private的类成员
- ☐ private的类成员

9.下面关于C++中 `sizeof` 的说法正确的是：（1.0分）

- ☐ 若 `long x = 229`; 则 `sizeof(x) = 229`
- ☐ `sizeof`是C++的库函数
- ☒ 若 `long long x = 110`; 则 `sizeof (x)` 等于 `sizeof (long long)`
- ☐ 如果要确定int类型的长度，应该用 `sizeof int`

10.下面关于数组初始化的相关说法，错误的是（1.0分）

- ☐ 若初始化语句为 `long arr[10] = {0} ;`，则数组元素全部初始化为0
- ☐ 初始化数组时，提供的值可以少于数组的元素数目。
- ☒ 若初始化语句为 `long arr[10] = {1} ;`，则数组元素全部初始化为1
- ☐ 如果只对数组的一部分进行初始化，则编译器将把其他元素设置为0。

11.假定系统中 `unsigned short` 类型的位宽是2个字节，若程序中声明了这样一个数组：（1.0分）

组： `unsigned short bugs[5] = {};`，则： `sizeof(bugs)` 的值是：

- ☒ 10
- ☐ 2
- ☐ 0
- ☐ 5

12.下面关于 模板类 `vector` 的说法，错误的是（1.0分）

- ☐ 要使用`vector`对象，必须包含头文件： `#include <vector>`
- ☐ 可以在运行阶段重新设置`vector`对象的长度，可在末尾附加新数据，还可在中间插入新数据
- ☐ `vector`包含在名称空间 `std` 中，因此可使用 `using` 编译指令、 `using` 声明或 `std::vector`
- ☒ `vector`类内部使用`new`和`delete`来管理内存，但是这种工作需要程序员自己来完成。

13.下面关于C++结构体的说法，错误的是（1.0分）

- ☐ 声明结构体变量的时候才会开辟内存空间
- ☐ 结构体的类型定义只描述结构体的组成，不分配内存
- ☐ 结构体变量的成员可以使用 点运算符(.) 来访问
- ☒ 在C++中声明结构体变量的时候必须使用 `struct` 关键字

14.程序中，有这样一条语句： `int * p_dogs`，则下面的说法，错误的是：（1.0分）

- ☐ `*p_dogs` 是一个int类型的变量
- ☐ `p_dogs` 是一个指向int类型的指针变量

- ☐ p_dogs 是一个指向int类型的指针变量
- ☒ *p_dogs 是一个指针变量
- ☐ p_dogs 的类型是 int*
-

15.关于 new 和 delete 这两个运算符，下面的用法，正确的是 (1.0分)

- ☐ int * ptr = new int[10]; delete ptr;
- ☐ int jugs = 10; int * pj = &jugs; delete jugs;
- ☒ int * ptr = new int[10]; delete[] ptr;
- ☐ int * ps = new int; delete ps; delete ps;
-

16.C++程序中有这样一条语句： int * ptr = new int[10]，则下面的说法错误的是： (1.0分)

- ☐ ptr[3] 的值等于 *(ptr+3)
- ☐ ptr指向数组的第一个元素
- ☒ 如果sizeof(int)的值是4，则sizeof(ptr)的值是40
- ☐ *ptr 的值等于 ptr[0]
-

17.下面的程序，输出结果为： (1.0分)

```
1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5.     int x = 100;
6.     {
7.         int x = 300;
8.         int y = 200;
9.     }
10.    cout<<x<<"", "<<y<<endl;
11.    return 0;
12. }
```

- ☐ 100,0
- ☐ 100,200
- ☒ 编译不通过
- ☐ 300,200
-

18.下面的程序，输出结果为： (1.0分)

```

1. #include <iostream>
2. using namespace std;
3. void maxfun(int x, int y, int m)
4. {
5.     m = x>y?x:y
6. }
7. int main()
8. {
9.     int a=10, b=5, k=1;
10.    maxfun(a,b,k);
11.    cout<<"k="<<endl;
12.    return 0;
13. }

```

- ☒ k=1
- ☐ k=10
- ☐ 编译不通过
- ☐ k=5

19.下面的程序，输出结果为： (1.0分)

```

1. #include <iostream>
2. using namespace std;
3. void maxfun(int x, int y, int* m)
4. {
5.     (*m) = x>y?x:y
6. }
7. int main()
8. {
9.     int a=10, b=5, k=1;
10.    maxfun(a,b,&k);
11.    cout<<"k="<<endl;
12.    return 0;
13. }

```

- ☐ k=1
- ☒ k=10
- ☐ 编译不通过
- ☐ k=5

20.下面的程序，输出结果为： (1.0分)

```

1. #include <iostream>
2. using namespace std;
3. void maxfun(int x, int y, int& m)

```

```

4. {
5.     m = x>y?x:y
6. }
7. int main()
8. {
9.     int a=10, b=5, k=1;
10.    maxfun(a,b,&k);
11.    cout<<"k="<<endl;
12.    return 0;
13. }

```

- ☐ k=1
- ☐ k=10
- ☒ 编译不通过
- ☐ k=5

21.下面的函数原型声明，错误的是 (1.0分)

- ☐ int gcd(int m, int n);
- ☒ int gcd(int m=2, int n);
- ☐ int gcd(int m=9, n=90);
- ☐ int gcd(int m, int n=10);

22.下面的程序输出结果是： (1.0分)

```

1. #include <iostream>
2. using namespace std;
3. int add(int x=5, int y=6)
4. {
5.     return x+y;
6. }
7. int main
8. {
9.     cout<<add(10,20)<<","<<endl;
10.    cout<<add(10)<<","<<endl;
11.    cout<<add()<<endl;
12.    return 0;
13. }

```

- ☐ 编译不通过
- ☒ 30,16,11
- ☐ 16,20,11
- ☐ 30,

16,

11

23.下面的一对函数，不属于函数重载的是（1.0分）

- ☒ `int add(int a, int b);`
`int add(int, int)`
- ☐ `int add(int a, int b);`
`double add(double a, double b)`
- ☐ `int add(int a, float b);`
`float add(float a, int b);`
- ☐ `int add(int a, int b);`
`int add(int a, int b, int c)`

24.下面关于C++的类的定义的相关说法，错误的是：（1.0分）

- ☐ 当你在定义类的时候，不提供构造函数，则C++编译器会给你提供一个默认构造函数
- ☐ 在定义类时，声明为public,private或protected的成员的次序任意
- ☒ 当你在定义类的时候，必须提供析构函数
- ☐ 在一个类体中，不一定都包含public,private和protected的部分，可以只有public、private、protected的部分或任意组合

25.设函数 fun 的定义形式为 `void fun(char ch,float x){...}`，则以下对函数 fun 的调用语句（1.0分）中，正确的是（）。

- ☐ `fun('65',2.8);`
- ☐ `fun("abc",3.0);`
- ☒ `fun(32,32);`
- ☐ `t=fun('D',16.5);`

二、多选题（共5题；共10.0分）

26.下面关于C++变量命名的说法，正确的是：（2.0分）

- ☐ C++的关键字可以用作变量名
- ☒ 在C++中，变量的类型是对变量占用的内存空间的大小和存储方式做出的规定
- ☒ 变量名称的第一个字符不可以是数字
- ☒ 变量名称中只能使用**字母字符、数字、下划线()**，但是不能使用**横线(-)**。

27.下面关于C++的重载函数的说法，正确的是：（2.0分）

- ☐ 两个函数如果是重载的，则它们的返回值必须相同

- ☐ 两个函数如果是重载的，则它们的返回值必须相同
- ☒ 两个函数如果是重载的，则它们的函数名称必须相同
- ☒ 重载函数的参数可以带有默认值
- ☒ 两个函数如果是重载的，则它们的形参列表必须不相同

28.关于C++的类和对象，下列说法正确的是（2.0分）

- ☒ 类与类之间可以通过封装而具有明确的独立性
- ☐ 对象与对象之间可以通过消息进行通信和联络
- ☒ 类与类之间必须是平等的关系，而不能组成层次关系或包含关系
- ☒ 类用于描述事物的属性和对事物的操作

29.下面关于C++中const常量的说法，正确的是：（2.0分）

- ☒ const常量被初始化后其值就固定了，编译器不允许修改常量的值。
- ☐ 若 `const long double FA = 666;` 则 FA是个整型的常量。
- ☒ 定义在一个函数内的const常量在另一个函数中是不可见的
- ☒ const常量必须在声明的同时进行初始化。

30.下面的C++代码，在遵循C++11标准的编译器上，能通过编译的是：（2.0分）

- ☒

```
1. class Point { int x; float y};
2. class Line
3. {
4.     Point a;
5.     Line ln;
6. };
```
- ☐

```
1. class Point;
2. class Line
3. {
4.     Point a;
5. };
```
- ☒

```
1. class Point { int x; float y};
2. class Line
3. {
4. public:
5.     Point a;
6.     Line* ln;
7. private:
8.     double length=0.0;
9. };
```




```
1. class Point { int x; float y};  
2. class Line  
3. {  
4.     Point a;  
5. };
```

三、判断题 (共15题; 共15.0分)

31.C++ 整数类型的位宽排序: $\text{char} < \text{short} \leq \text{int} \leq \text{long} \leq \text{long long}$ (1.0分)

☒ 正确 ☐ 错误

32.在C++中, 对布尔变量赋值, 任何非零值都被转换为true, 而零被转换为false (1.0分)

☒ 正确 ☐ 错误

33. #define 预编译指令的工作原理类似于查找替换: 在预编译过程中把所有符号全部替换成对应的字面常量 (1.0分)

☒ 正确 ☐ 错误

34.结构体变量每个成员分别占有自己的内存单元 (1.0分)

☒ 正确 ☐ 错误

35.共用体变量所占的内存长度等于最长的成员所占的内存长度 (1.0分)

☒ 正确 ☐ 错误

36.若定义了一个枚举类型 `enum color { red=3, blue, green, orange}`, 则第三个枚举量 `green` 的值为 2。 (1.0分)

☐ 正确 ☒ 错误

37.在C++中, 指针是一个变量, 其存储的是值的地址, 而不是值本身 (1.0分)

☒ 正确 ☐ 错误

38.在C/C++的内存模型中：“每个字节都有一个唯一的地址。”（1.0分）

☒ 正确 ☐ 错误

39.C/C++数据对象模型：“数据对象的地址是它占据的第一个字节的地址。”。（1.0分）

☒ 正确 ☐ 错误

40.关于C++的cin对象的 `get()` 和 `getline()`，`getline()` 将丢弃换行符，`get()` 将换行符保留在输入序列中（1.0分）

☒ 正确 ☐ 错误

41.C++11新增了模板类 `array`，`vector` 的数组长度可以在运行期间动态改变，但是 `array` 对象的数组长度在运行期间是固定的。（1.0分）

☒ 正确 ☐ 错误

42.在C++中，常规数组元素的内存空间不是连续分配的（1.0分）

☐ 正确 ☒ 错误

43.这两条C++程序语句 `float * pf; *pf = 100.100F;` 可以编译通过（1.0分）

☐ 正确 ☒ 错误

44.类的每一个对象都会拥有自己的非静态数据成员，但是成员函数是存放在公共代码区域，所有对象共用的。（1.0分）

☒ 正确 ☐ 错误

45.在C++中，用new运算符创建的数据实体都被存放在自由存储区(即 堆内存)。（1.0分）

☒ 正确 ☐ 错误

四、简答题（共2题；共20.0分）

46.请简要概述：面向对象编程和面向过程编程的相同点和不同点（10.0分）

相同点：

面向对象和面向过程编程在解决问题的过程中都要有‘操作对象’即数据，都是对数据进行一定的操作后得到符合要求的结果然后进行输出；

进行编程解决问题的过程相同，都是先进行问题分析，提取操作对象，建立数学模型，用编程语言

进行编程解决问题的过程相同，都是先进行问题分析、提取操作对象、建立数学模型、用编程语言进行求解、得到结果后利用规律进行实际问题的解答；

不同点：

面向过程的编程是 由一个程序完成一整个任务，面向对象是 一个或一种对象做一件事最后共同完成一个任务；

面向对象的编程写出来的程序更加优美，也更容易修改，比如：有一个商家需要一个商品结算的程序，用两种方法编程出来的代码不会有太大差别。到了国庆商店价格有活动，过程编程在程序中加了一个判断语句进行处理，对象编程只需要在类中做相应的修改，活动越来越多，过程编程的代码越来越长越冗杂，而对象编程的代码仍能保持优美且修改方便；

面向过程是以完成一个任务需要哪些步骤进行编程的，而面向对象是把问题解决的每一个阶段抽象出一个类来，由该类创建的对象完成。

47.请列举你学到的关于C++类与对象中构造函数和析构函数的相关知识点（10.0分）

构造函数：

- 1>构造函数的名字和类的名字相同；
- 2>构造函数如果程序员不定义的话，编译器会自动定义一个默认构造函数；
- 3>构造函数可以重载，即可以创建多个构造函数；
- 4>构造函数可以利用初值列表进行赋初始值，比如

```
class complex
{
    private:
        int re,im;
    public:
        complex(int x,int y):re(x),im(y){ }
}
```

- 5>构造函数也可以定义成内联的(inline),在类内部定义的函数一般都是默认为内联。(如果有的函数太大即使程序员定义成内联的，编译器也不会设置为内联的。)

析构函数：

- 1>析构函数使用来回收已经创建的类型对象；
- 2>析构函数的定义为，~类名() {} 比如

```
class complex
{
    private:
        int re,im;
    public:
```

```
complex(int x,int y):re(x),im(y){ }  
//析构函数  
~complex(){}  
}
```

3> 在程序设计中创建含有 指针类型的数据成员 的类，一定要定义析构函数；

4> 调用一个类类型的对象的析构函数后，这个对象所占用的内存空间会被回收；

5> 可以在析构函数的函数体内加一些语句提示这个对象已经被回收；

五、实训题 (共3题；共30.0分)

48.通过所有算例测试即为回答正确 (10.0分)

阶段成绩

通过所有算例测试即为回答正确

第1关 设计一个长方形类 10.0分

49.通过所有测试用例就算成功完成此题目 (10.0分)

阶段成绩

通过所有测试用例就算成功完成此题目

第1关 删除已排序数组中的重复项 10.0分

50.通过所有测试用例就算完成本题目。 (10.0分)

阶段成绩

通过所有测试用例就算完成本题目。

第1关 阿尔法乘积 10.0分