

- 一二章 1. 下列 C++ 标点符号中表示行注释开始的是 C 。
- A) # B) ; C) // D) }
2. 下列字符串中可以用作 C++ 标示符的是 A 。
- A) _256 B) class C) delete D) 5
char
3. 下列关于 C++ 函数的描述中正确的是 C 。
- A) 主函数是程序的入口，它是由用户定义的函数调用。 B) 函数是 C++ 中面向对象程序设计框架的基本组成元素。
C) 函数在调用前必须先被声明。 D) 每个用户定义函数只能被其他函数调用一次。
4. 下列关于 C++ 关键字的说法中正确的是 D 。
- A) 关键字是用户为程序中各种需要命名的“元素”所起的名字。 B) 关键字是对程序中的数据进行操作的一类单词。
C) 关键字是在程序中起分割内容和界定范围作用的一类单词。 D) 关键字是 C++ 中预先定义并实现一定功能得一类单词。
5. 每个 C++ 程序都必须由且仅有一个 B 。
- A) 预处理命令 B) 主函数 C) 函数 D)
语句
6. 下面关于对象概念的描述中， A 是错误的。
- A) 对象就是 C 语言中的结构变量 B) 对象代表着正在创建的系统中的—个实体
C) 对象是一个状态和操作（或方法）的封装体 D) 对象之间的信息传递是通过消息进行的
7. C++ 对 C 语言做了很多改进，下列描述中 D 使得 C 语言发生了质变，即从面向过程变成面向对象。
- A) 增加了一些新的运算符 B) 允许函数重载，并允许设置默认参数
C) 规定函数说明必须用原型 D) 引进类和对象的概念
8. 下面关于类概念的描述中， D 是错误的。
- A) 类是抽象数据类型的实现 B) 类是具有共同行为的若干对象的统一描述体
C) 类是创建对象的样板 D) 类就是 C 语言中的结构体类型
9. C++ 语言是以 B 语言为基础逐渐发展而演变而成的一种程序设计语言。
- A) Pascal B) C C) Basic
D) Simula67
10. C++ 语言中规定函数的返回值的类型是由 D 。
- A) return 语句中的表达式类型所决定的 B) 调用该函数式的主调用函数类型所决定的
C) 调用该函数式系统临时确定的 D) 在定义该函数式所指的数据类型所决定的
11. 下列关于 C++ 与 C 语言关系的描述中错误的是 C 。
- A) C++ 是 C 语言的超集 B) C++ 是 C 语言进行了扩充
C) C++ 是 C 语言都是面向对象的程序设计语言 D) C++ 包含 C 语言的全部语法特征
12. 下列字符串中可以用做 C++ 标示符的是 A 。
- A) 521 B) computer~1 C) case D) 3var
13. 面向对象程序设计思想的主要特征中不包括 D 。
- A) 封装性 B) 多态性 C) 继承性 D) 功能分解，逐步求精
14. 下列 C++ 标点符号中表示一条预处理命令开始的是 A 。
- A) # B) // C) } D) ;

15. 下列关于 C++ 类的描述中错误的是 C。
- A) 类与类之间可以通过一些手段进行通信和联络 B) 类用于描述事物的属性和对事物的操作
C) 类与类之间必须是平等的关系，而不能组成层次关系 D) 类与类之间可以通过封装而具有明确的独立性
16. 下列 C++ 标点符号中表示一条语句结束的是 A。
- A) ; B) // C) # D) {
17. 下列 C++ 标点符号中表示复合语句结束的标记符是 C。
- A) # B) // C) } D) ;
18. 下列 C++ 标点符号中表示复合语句开始的标记符是 A。
- A) { B) // C) } D) ;
19. 下列关键字中， B 即在 C 语言中使用，又在 C++ 语言中使用。
- A) inline B) break C) private D) public
20. 下列字符串中不可以用作 C++ 标示符的是 C。
- A) 521 B) computer123 C) int D) flower
21. 下列不正确的选项是 D。
- A) C++ 语言是一种面向对象的程序设计语言，它支持面向对象思想中的 3 个主要特征 B) 标点符号是在程序中起分割内容和界定范围作用的一类单词
C) iostream 是一个标准的头文件，定义了一些输入输出流对象
D) 类与类之间不可以进行通信和联络
22. 下列不正确的选项是 B。
- A) 封装是一种信息隐藏技术 B) 标识符是由字母、数字、下划线组成的字符串，必须以数字或下划线开头
C) 编译是由源程序文件转换到目标文件的过程 D) 一个 C++ 程序可以认为是函数串
23. 下列与流程控制有关的关键字是 A。
- A) goto B) static C) int D) sizeof
24. 下列正确的选项是 A。
- A) 继承是创建一个具有别的类的属性和行为的新类的能力 B) C 语言支持面向对象的程序设计
C) 空白符是一种可以被编译的符号 D) 标示符不宜过长，一般设定为 16 个字符
25. 下列关于 C++ 语言的发展说法错误的是 C。
- A) C++ 语言起源于 C 语言 B) C++ 语言最初被称为“带类的 C” C) 在 1980 年 C++ 被命名 D) 在 1983 年 C++ 被命名
26. 下面说法错误的选项是 C。
- A) 引用表达式是一个左值表达式，因此它可以出现在形参和实参的任何一方
B) 若一个函数返回了引用，那么该函数的调用也可以被赋值
C) 所有函数都可以返回引用 D) 引用返回值时，不生成值的副本
27. 下列表示引用的方法中， A 是正确的。
- 已知：int k=1000;
- A) int &x=k; B) char &y; C) int &z=1000; D) float &t=&k;
28. 下列程序的执行结果为 A。
- ```
#include<iostream.h>
void main()
```

```

{
 int n=10;
 int* pn=&n;
 int* &rn=pn;
 (*pn)++;
 cout<<"n="<<n<<endl;
 (*rn)++;
 cout<<" n=" <<n<<endl;
}

```

A) n=11

n=12

B) n=11

n=11

C) n=10

n=11

D) n=11

n=10

29. 下列引用的定义中， B 是错误的。

A) int i; int &j=i; B) int i; int &j; j=i;

C) float i; float &j=i; C) char d; char &k=d;

30. B 是给对象取了一个别名，他引入了对象的同义词。

A) 指针 B) 引用 C) 枚举 D) 结构

31. 下列表示引用的方法中， A 是正确的。

已知： int m=10;

A) int &x=m;

B) int &y=10;

C) int

&z;

D) float &t=&m;

32. 下面程序的运行结果是 A 。

```

#include<iostream.h>

```

```

void fun(int &a, int &b)

```

```

{

```

```

 int p;

```

```

 p=a;

```

```

 a=b;

```

```

 b=p;

```

```

}

```

```

void exchange(int &a, int &b, int &c)

```

```

{

```

```

 if(a<b) fun(a, b);

```

```

 if(a<c) fun(a, c);

```

```

 if(b<c) fun(b, c);

```

```

}

```

```

void main()

```

```

{

```

```

 int a, b, c;

```

```

 a=12;

```

```

 b=639;

```

```

 c=78;

```

```

exchange(a, b, c);
cout<<"a="<<a<<"", b="<<b<<"", c="<<c<<endl;
}

```

- A) a=639, b=78, c=12                      B) a=12, b=639, c=78                      C) a=639, b=12, c=78                      D) a=78, b=12, c=639

33. 下面的程序输出结果为 C 。

```

#include<iostream.h>
void main()
{
 int *ir;
 int i;
 ir=&i;
 i=17;
 cout<<"int i="<<i<<endl;
 cout<<"int ir="<<*ir<<endl;
}

```

- A) int i=17                      int ir=0                      B) int i=0                      int ir=17  
C) int i=17                      int ir=17                      D) 程序语法错误，没有输出结果

34. 对定义重载函数的下列要求中， D 是错误的。

- A) 要求参数的个数不同                      B) 要求参数中至少有一个类型不同  
C) 要求参数个数相同时，参数类型不同                      D) 要求函数的返回值不同

35. 下面函数调用语句中实参的个数是 A 。

```
xxx((v1, v2), (v3, v4, v5), v6);
```

- A) 3                      B) 4                      C) 5                      D) 6

36. 系统在调用重载函数时往往根据一些条件确定哪个重载函数被调用，在下列选项中，不能作为依据的是 D 。

- A) 参数个数                      B) 参数的类型                      C) 函数名  
称                      D) 函数的类型

37. 若要定义一个只允许本源文件中所有函数使用的全局变量，则该变量需要使用的存储类别是 D 。

- A) extern                      B) register                      C) auto                      D) static

38. 在 C++ 中，关于下列设置参数默认值的描述中，正确的是 C 。

- A) 不允许设置设置参数的默认值                      B) 设置参数默认值只能在定义函数时设置  
C) 设置参数默认值时，应该实现设置右边的再设置左边的                      D) 设置参数默认值时，应该全部参数都设置

39. 若有下面函数调用语句：

```
fun(a+b, (x, y), fun(n+k, d, (a+b)));
```

在此函数调用语句中实参的个数是 A 。

- A) 3                      B) 4                      C) 5                      D) 6

40. 下列对重载函数的描述中， A 是错误的。

- A) 重载函数中不允许使用默认参数                      B) 重载函数中编译时根据参数表进行选择  
C) 不要使用重载函数来描述毫无相干的函数                      D) 构造函数重载将会给初始化带来多种方式

41. 在函数声明时，下列 D 项是不必要的。

A) 函数的类型      B) 函数参数类型      C) 函数的名字      D) 返回值表达式

42. 在函数的返回值类型与返回值表达式的类型的描述中，错误的是 B 。

A) 函数返回值的类型是在定义函数时确定，在函数调用时是不能改变的      B) 函数返回值的类型就是返回值表达式的类型

C) 函数返回值表达式类型与返回值类型不同时，函数表达式类型应转换成返回值类型

D) 函数返回值类型确定了返回值表达式的类型

43. 下面变量和函数说明：

```
#include<iostream.h>
char ch='*';
void sub(int x,int y,char ch,double *z)
{
 switch(ch)
 {
 case '+':*z=x+y;break;
 case '-':*z=x-y;break;
 case '*':*z=x*y;break;
 case '/':*z=x/y;break;
 }
}
```

下面合法的函数调用语句

是 B 。

A) sub(10, 20, ch, y);      B) sub(1. 2+3. 2, '+', &z);

C) sub(sub(1, 2, '+', &y), sub(3, 4, '+', &x), '-', &y);      D) sub(a, b, &x, ch);

44. 下列程序的输出结果是 B 。

```
#include<iostream.h>
void sub1(char a,char b)
{
 char c;
 c=a;
 a=b;
 b=c;
}
void sub2(char *a,char b)
{
 char c;
 c=*a;
 *a=b;
 b=c;
}
void sub3(char *a,char *b)
{
```

```

char c;
c=*a;
*a=*b;
*b=c;
}
void main()
{
 char a, b;
 a='A';
 b='B';
 sub1(a, b);
 putchar(a);
 putchar(b);
 a='A';
 b='B';
 sub2(&a, b);
 putchar(a);
 putchar(b);
 a='A';
 b='B';
 sub3(&a, &b);
 putchar(a);
 putchar(b);
 putchar('\n');
}

```

A) ABABAB

B) ABBBBA

C) ABBABA

D) ABABBA

45. 下面程序的正确执行结果为

D

。

```

#include<iostream.h>
int f(int);
void main()
{
 int a=2, i;
 for(i=0; i<3; i++)
 cout<<f(a)<<" ";
 cout<<endl;
}
int f(int a)
{
 int b=0;
 static int c=3;
 b++;
}

```

```

 c++;
 return (a+b+c);
}

```

A) 7 7 7                      B) 7 10 13                      C) 7 9 11                      D) 7 8 9

46. 下面程序的正确执行结果为                      C                      。

```

#include<iostream.h>
void fun(char * *m)
{
 m++;
 cout<<*m<<endl;
}
void main()
{
 static char *a[]={ "MORNING", "AFTERNOON", "EVENING" };
 char * *n;
 n=a;
 fun(n);
}

```

A) 为空                      B) MORNING                      C) AFTERNOON                      D) EVENING

47. 下面程序的输出结果                      B                      。

```

#include<iostream.h>
int fun(char *s)
{
 char *p=s;
 while(*p!='\0')
 p++;
 return (p-s);
}
void main()
{
 cout<<fun("ABCDEF")<<endl;
}

```

A) 3                      B) 6                      C) 8                      D) 0

48. 当一个函数无返回值时，函数的类型应定义为                      A                      。

A) void                      B) 任意                      C) int                      D) 无

49. 下面叙述中不正确的是                      D                      。

A) 在函数中，通过 return 语句传回函数值                      B) 在函数中，可以有多条 return 语句  
 C) 主函数名 main() 后面的一对圆括号中也可以带有形参                      D) 调用函数必须在一条独立的语句中完成

50. 下列                      D                      的调用方式是引用调用。

A) 形参是指针，实参是地址值                      B) 形参和实参都是变量                      C) 形参是数组，实参是变量                      D) 形参是引用，实参是变量

51. 在 C++ 中，函数原型不能标识 D。
- A) 函数的返回类型      B) 函数参数的个数      C) 函数参数类型      D) 函数的功能
52. 下列标识符中，C 是文件级作用域的。
- A) 函数形参      B) 语句标号      C) 外部静态类标识符      D) 自动类标识符
53. 在传值调用中，要求 C。
- A) 形参和实参类型任意，个数相等      B) 实参和形参类型都完全一致，个数相等
- C) 实参和形参对应的类型一致，个数相等      D) 实参和形参对应的类型一致个数任意
54. 有一个 int 型变量，在程序中频繁使用，最好定义它为 A。
- A) register      B) auto      C) extern      D) static
55. 考虑下面的函数原型：void testDefaultParam(int a, int b=7, char z='\*');  
下面函数调用中不合法的 C。
- A) testDefaultParam(5);      B) testDefaultParam(5, 8);
- C) testDefaultParam(5, '#');      D) testDefaultParam(0, 0, '\*');
56. 采用函数重载的目的在于 D。
- A) 实现共享      B) 减少空间      C) 提高速度      D) 使用方便，提高可靠性
57. 要求通过函数来实现一种不太复杂的功能，并且要加快执行速度，选用 A。
- A) 内联函数      B) 重载函数      C) 递归函数      D) 嵌套函数
58. 下面程序的输出结果是 D。

```
#include<iostream.h>
void fun(int i);
int main()
{
 int n=2;
 fun(n);
 return 0;
}
void fun(int i)
{
 static int x=1;
 int y=1;
 if(i>0)
 {
 ++x;
 ++y;
 cout<<x<<' '<<y<<' ';
 fun(i-1);
 }
}
```

- A) 语法错误，不能输出正确结果      B) 2 2 2 2      C) 2 2 3 3      D) 2 2 3 2
59. 下面程序的输出结果为 B。
- ```
#include<iostream.h>
```



```

void prt(int *x, int *y, int *z)
{
    cout<< + *x<< " , "<< + *y<< " , "<< *(z++)<<endl;
}
int a=10;
int c=20;
int b=40;
void main()
{
    prt (&a, &b, &c);
    prt (&a, &b, &c);
}

```

A) 11, 42, 31 12, 21, 41 B) 11, 41, 20 12, 42, 20
 C) 11, 21, 40 11, 21, 41 D) 11, 41, 21 12, 42, 22

60. 下面程序的正确运行结果为 C 。

```

#include<iostream.h>
fun(int a,int b)
{
    static int m=1,i=2;
    i+=m+1;
    m=i+a+b;
    return(m);
}

```

```

void main()
{
    int k=5,m=2,p;
    p=fun(k,m);
    cout<<p<<" , ";
    p=fun(k,m);
    cout<<p<<endl;
}

```

A) 11, 22 B) 12, 23 C) 11, 23 D) 10, 23

61. 下面程序的输出结果是 D 。

```

#include<iostream.h>
void fun(int *s, int *y)
{
    static int t=3;
    *y=s[t];
    t--;
}
void main()

```

```

{
    int a[]={1, 2, 3, 4};
    int i, x=10;
    for(i=0; i<4; i++)
    {
        fun(a, &x);
        cout<<x<<" ";
    }
    cout<<endl;
}

```

A) 1, 2, 3, 4 B) 0, 0, 0, 0 C) 4, 4, 4, 4 D) 4, 3, 2, 1

62. 下面程序的输出结果是

A 。

```

#include<iostream.h>
int fun(char *s)
{
    char *p=s;
    while(*p!='\0')
        p++;
    return (p-s);
}
void main()
{
    cout<<fun("ABCDEFGHJKLMNIPQRST")<<endl;
}

```

A) 20 B) 6 C) 19 D) 0

63. 下面程序的输出结果是

C 。

```

#include<iostream.h>
int x=10;
void fun(int c);
void main()
{
    fun(x);
    cout<<"x="<<x<<endl;
}
void fun(int x)
{
    x=1;
}

```

A) x=0 B) x=1 C) x=10 D) 语法错误，不能输出正确结果

64. 下面程序的执行结果是

B 。

```

#include<iostream.h>

```

```

void main()
{
    void swap();
    extern int a,b;
    a=3;
    b=10;
    swap();
    cout<<"a="<<a<<",b="<<endl;
}

```

```
int a,b;
```

```
void swap()
```

```

{
    int temp;
    temp=a;
    a=b;
    b=temp;
}

```

A) a=3, b=10 B) a=10, b=3 C) a=temp, b=temp D) 语法错误，不能输出正确结果

65. 下列程序的输出结果是

A

。

```
#include<iostream.h>
```

```
void f()
```

```

{
    static int i=15;
    i++;
    cout<<"i="<<i<<endl;
}

```

```
void main()
```

```

{
    for(int k=0;k<2;k++)
        f();
}

```

A) i=16 i=17 B) i=16 i=15 C) i=15 i=16 D) i=15 i=17

66. 在横线处填上适当的语句 C

，使下面程序执行结果是 13, 18, 23。

```
#include<iostream.h>
```

```
int add(int x,int y=8);
```

```
void main()
```

```

{
    _____;

    cout<<add(a)<<" ";
    cout<<add(a,add(a))<<" ";
    cout<<add(a,add(a,add(a)))<<endl;
}

```

```

}
int add(int a,int b)
{
    int s=a+b;
    return s;
}

```

A)int a=3 B)int b=3 C)int a=5 D)int b=5

67. 在横线处填上适当的语句 A ， 使下面的程序的执行结果是 19 24 30 37 45。

```
#include<iostream.h>
```

```
int fun(int a)
{
    static int x=10;
    int y=1;
    x+=a;
    a++;
    y++;
    return(x+y+a);
}

```

```
void main()
```

```
{
    int i=3;
    while(i<8)

```

```
;
```

```
    cout<<endl;
}

```

A)cout<<fun(i++)<<" " B)cout<<fun(i)<<"
" C)cout<<fun(i++)<<" ," D)cout<<fun(i)<<" ,"

68. 在 C++语言中，局部变量的隐含存储类型是 D 。

A)auto B)static C)extern D)无存储类型

69. 下面程序的输出结果为 。

```
#include<iostream.h>
```

```
int fun(char *s);
```

```
int main()
```

```
{
    cout<<fun("hello");
    return 0;
}

```

```
int fun(char *s)
```

```
{
    char *t=s;
    while(*t!='\0')
```

```

        t++;
    return(t-s);
}

```

A) 语法错，不能输出正确结果 B) 0 C) 6 D) 5

第三章一、选择题

1. 有关类的说法不正确的是____ D ____。

A) 类是一种用户定义的数据类型。

B) 只有类中的成员函数才能存取类中的私有成员。

C) 在类中，如果不做特别说明，所指的数据均为私有类型。

D) 在类中，如果不做特别说明，所指的成员函数均为公有类型。

2. 类 A 是类 B 的友元，类 B 是类 C 的友元，则____ D ____是正确的。

A) 类 B 是类 A 的友元

B) 类 C 是类 A 的友元

C) 类 A 是类 C 的友元

D) 以上都不对

3. 有下面类的说明，有错误的语句是____ A ____。

```

class X
{

```

A) int a=2;

B) X();

public:

C) X(int val):

D) ~X();

```

};

```

4. 有关析构函数的说法不正确的是____ B ____。

A) 析构函数有且仅有一个

B) 析构函数和构造函数一样可以有形参

C) 析构函数的功能是用来释放一个对象

D) 析构函数无任何函数类型

5. 已知一个类 X，____ C ____是定义指向类 X 成员函数的指针，假设类有 3 个公有成员：void f1(int)，void f2(int) 和 int a。

A) X*P

B) int X::*pc=&X::a

C) void(X::*pa)()

D) X*p[10]

6. 下列的各类函数中，____ C ____不是类的成员函数

A) 构造函数

B) 析构函数

C) 友元函数

D) 拷贝构造函数

7. 下列定义中, X 是一个类, _____ B _____ 是定义指向对象数组的指针 p.

A) x*p[4]

B) x(*p)[4]

C) (x*)p[4]

D) x*p[]

8. 已知 p 是一个指向类 X 数据成员 m 的指针, s 是类 X 的一个对象。如果要给 m 赋值为 3, __ C ____ 是正确的.

A) s.p=3

B) s ->p=3

C) s.*p=3

D) *s.p=3

9. 已知 f1(int) 是类 X 的公有成员函数, p 是指向成员 f1() 的指针, 采用它赋值, _____ B _____ 是正确的.

A) p=f1

B) p=X::f1

C) p=x::f1

D) p=f1()

10. 已知类 X 中的一个成员函数说明如下:

```
Void Set(x &a);
```

其中, X&a 的含义是_____ C ____.

A) 指向类 X 的指针为 a

B) 将 a 的地址赋给变量 Sst

C) a 是类 X 的对象引用, 用来做为 Set() 的形参

D) 变量 X 是 a 按位相与作为函数 Set() 的参数

11. 一个类的友元函数能够访问该类的_____ D ____.

A) 私有成员

B) 保护成员

C) 公有成员

D) 所有成员

12. 有关类和对象的说法不正确的是_____ B ____.

A) 对是类的一个实例

B) 一各类只能有一个对象

C) 任何一个对象只能属于一个具体的类

D) 类与对象的关系和数据类型和变量的关系相似

13. 下列静态数据成员的特性中, _____ D _____ 是错误的.

A) 说明静态数据成员时前边要加关键字 static 来修饰

B) 静态数据成员在类体外进行初始化

C) 引用静态数据成员时, 要在静态数据成员名前加<类名>和作用域运算符

D) 静态数据成员不是所有对象所共有的

14. 类模板的使用实际上是类模板实例化成一个具体的_____ A ____.

A) 类

B) 对象

C) 函数

D) 模板类

15. 通常拷贝构造函数的参数表是_____ C _____.

A) 某个对象名

B) 某个对象的成员名

C) 某个对象的引用名

D) 某个对象的指针名

16. 关于常数据成员的说法不正确的是_____ D _____.

A) 常数据成员的定义形式与一般常量的定义形式相同, 只不过常数据成员的定义必须出现在类体中.

B) 常数据成员必须进行初始化, 并且不能被更新

C) 常数据成员通过构造函数的成员初始化列表进行初始化

D) 常数据成员可以在定义时直接初始化

17. 已知 X 类. 则当程序执行到语句

```
X array[3];
```

时, 调用了_____ D _____次构造函数.

A) 0

B) 1

C) 2

D) 3

18. 类模板的模板参数_____ D _____.

A) 只可作为数据成员的类型

B) 只可作为成员函数的返回类型

C) 只可作为成员函数的参数类型

D) 以上三者皆可

19. 下列_____ D _____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) 构造函数可以设置默认参数

D) 构造函数必须指定类型说明

20. 下列程序中说明的公有成员是_____ D _____.

```
class Location{  
    int X;  
public:  
    void init(int initX,int initY);  
private:  
    int Y;  
public:  
    int GetX();  
    int GetY();  
}
```

A) X

B) init(int initX,int initY)

C)GetX(),GetY()

D)B)和C)都是

21. 类定义的内容允许被其他对象无限制地存取是_____ C _____.

A)private 部分

B)protected 部分

C)public 部分

D)以上都不对

22. 关于下面程序段说法正确的是_____ A _____.

```
class X
```

```
{
```

```
private:
```

```
    int n;
```

```
public:
```

```
    X(X&);
```

```
}
```

```
X::X(X&x)
```

```
{n=x.n}
```

```
X obj1,obj2;
```

```
obj2(obj1);
```

A) 语句 obj2 (obj1) ; 的功能是用对象 obj1 初始化具有相同类类型的对象 obj2

B) 语句 obj2 (obj1) ; 的功能是用对象 obj2 初始化具有相同类类型的对象 obj1

C) X(X&x)也可以不使用引用参数而仍然是拷贝构造函数

D) X(X&x)不能访问对象的私有成员

23. this 指针是 C++实现_____ B _____的一种机制.

A)抽象

B)封装

C)继承

D)重载

24. 下面程序段中是内联函数的是_____ C _____。

```
class location
```

```
{
```

```
private:
```

```
int x , y;
```

```
public:
```

```
void init(int initx,int inity)
```

```
{
```

```
    x=initx;
```

```
y=inity;
```

```
}
```

```
int getx(){return x;}
```

```
int gety();
```

```
};
```



```
inline int location::gety() {return y;}
```

- A) init (int initx,int inity)
- B) getx(),gety()
- C) A)和B) 都是
- D) A) 和 getx()

25. 关于对象成员的构造函数的调用顺序, 说法正确的是_____ C _____。

- A) 与他们在成员初始化列表中给出的顺序相同
- B) 与析构函数的调用顺序相同
- C) 与他们在类中说明顺序相同
- D) 以上说法都不对

26. 类模板 `template<class T>class X{.....}`, 其中, 友元函数 `f()` 成为该类模板实例化的每个模板类的友元, 则其说明应为_____ A _____。

- A) `friend void f()`
- B) `friend void f(X<T>&);`
- C) `friend void A::f();`
- D) `friend void C<T>::f(X<T>&);`

27. 运算符 `—>*` 的功能是 (A) 。

- A) 用来表示指向对象指针指向类成员指针的操作。
- B) 用来表示对象类成员的操作。
- C) 用来表示指向对象指针对类成员的操作。
- D) 用来表示对象指向类成员指针的操作。

28. 下列说明中

```
const char * ptr;
```

`ptr` 应该是 C 。

- A) 指向字符的指针
- B) 指向字符的常量指针
- C) 指向字符串常量的指针
- D) 指向字符串的常量指针

29. 下面关于友元的描述中, 错误的是 (D) 。

- A) 友元函数可以访问该类的私有数据成员。
- B) 一个类的友元类中的成员函数都是这个类的友元函数。
- C) 友元可以提高程序的运行效率。
- D) 类与类之间的友元关系可以继承。

30. 一个 (C) 允许拥护为类定义一种模式, 使得类中的某些数据成员、某些成员函数底参数和返回值可以取任意数据类型。

- A) 函数模板
- B) 模板函数
- C) 类模板
- D) 模板类

31. 有关构造函数说法不正确的是 (D)

- A) 构造函数的名字和类的名字一样。

- B) 在创建对象时，系统自动调用构造函数
C) 构造函数无任何函数类型
D) 构造函数有且只有一个
32. 下面对模板的声明，正确的是（ C ）
A) `template<T>`
B) `template<class T1,T2>`
C) `template<classT1,classT2>`
D) `template<classT1;classT2>`
33. （ D ）是析构函数的特征。
A) 析构函数可以有一个或多个参数
B) 析构函数定义只能在类体内
C) 析构函数名与类名不同
D) 一个类中只能定义一个析构函数
34. 类模板 `template<class T>class X{...};`，其中友元函数 `f` 对特定类型 `T`（如 `int`），使函数 `f(X<int>&)` 成为 `X<int>` 模板类的友元，则其说明应为（ B ）
A) `friend void f();`
B) `friend void f(X<T>&);`
C) `friend void A::f();`
D) `friend void C<T>::f(X<T>&);`
35. 若函数是在运行激活的，则该函数称为（ B ）
A) 内联函数
B) 非内联函数
C) 成员函数
D) 静态成员函数
36. 在 C++ 程序中，对象之间的象话通信通过（ B ）
A) 继承实现
B) 调用成员函数实现
C) 封装实现
D) 函数重载实现
37. 在类的定义形式中，数据成员、成员函数和（ A ）组成了类定义体。
A) 成员的访问控制信息
B) 公有消息
C) 私有消息
D) 保护消息
38. 对于任意一个类，析构函数的个数最多为（ B ）
A) 0
B) 1
C) 2
D) 3
39. （ C ）的功能是对对象进行初始化。
A) 析构函数

B) 数据成员

C) 构造函数

D) 静态数据成员

40. 友元关系不能 (A)

A) 继承

B) 是类与类之间是关系

C) 是一个类的成员函数与另一个类的关系

D) 提高程序的运行效率

41. (A) 只能访问静态成员变量，静态成员函数和类以外的函数和数据不能访问类中的非静态成员变量。

A) 静态函数

B) 虚函数

C) 构造函数

D) 析构函数

42. 下面说法正确的是 (B)

A) 内联函数在运行时是将该函数的目标代码插入每个调用该函数的地方

B) 内联函数在编译时是将该函数的目标代码插入每个调用该函数的地方

C) 类的内联函数必须在类体内定义

D) 类的内联函数必须在类体外关键字 inline 定义

43

下列关于成员访问权限的描述中，不正确的是 (B)

A) 公有数据成员和公有成员函数都可以被类对象直接处理

B) 类的私有数据成员只能被公有成员函数以及该类的任何友元类或友元函数访问

C) 保护成员在派生类中可以被访问，而私有成员不可以

D) 只有类或派生类的成员函数和友元类或友元函数可以访问保护成员

44. 局部变量可以隐藏全局变量，那么在有同名全局变量和局部变量的情形时，可以用 (A) 提供对全局变量的访问

A) 域运算符

B) 类运算符

C) 重载

D) 引用

45. 缺省析构函数的函数体是 (C)

A) 不存在

B) 随机产生的

C) 空的

D) 无法确定的、

46. 在下列关键字中，用以说明类中友元成员的是 (A)

A) public

B) private

C) protected

D) friend

47. 下面说法中正确的是 (B)

- A) 一个类只能定义一个构造函数, 但可以定义多个析构函数
- B) 一个类只能定义一个析构函数, 但可以定义多个构造函数
- C) 构造函数与析构函数同名, 知识名字前加了一个求反符号 (~)
- D) 构造啊函数可以指定返回类型, 而析构函数不能指定任何返回类型, 即使是 void 类型也不可以

48. 关于 delete 运算符的下列描述中, (C) 是错误的。

- A) 它必须用语 new 返回的指针
- B) 它也适用于空指针
- C) 对一个指针可以使用多次该运算符
- D) 指针名前只用一队方括号符, 不管所删除数组的维数

49. 关于成员函数的特征的下述描述中, 错误的是 (A)

- A) 成员函数一定是内联函数
- B) 成员函数可以重载
- C) 成员函数可以设置参数的默认值
- D) 成员函数可以是静态的

50. 假定 x 为一个类, 执行 Xa[3], *p[2]; 语句时会只动调用该类的构造函数 (B) 次

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

51. 由于数据隐藏的需要, 静态, 数据成员通常被说明为 (A)

- A) 私有的
- B) 公有的
- C) 保护的
- D) 不可访问的

52. 假定 X 为一个类, 则该类的拷贝构造函数的声明语句为 (C)

- A) My Class(My Class x)
- B) My Class&(My Class x)
- C) My Class(My Class &x)
- D) My Class(My Class *x)

53. Main() 函数在执行之前不可以调用其构造函数的 (C)

- A) 全局对象
- B) 静态对象
- C) 在 main() 函数体内说明的对象
- D) 全局对象数组

54. 对于常成员函数, 下列描述正确的是 (C)

- A) 常成员函数只能修改常数据成员
- B) 常成员函数只能修改一般数据成员
- C) 常成员函数不能修改任何数据成员
- D) _成员函数只能通过常对象调用

55. 友元的作用是 (A)

- A) 提高程序是运行效率
- B) 加强类的封装性
- C) 实现数据的隐藏性
- D) 增加成员函数的种类

56. 一个类的友元函数或友元类能够通过成员操作符访问该类的 (D)

- A) 私有成员
- B) 公有成员
- C) 保护成员
- D) 公有成员、保护成员和私有成员

57. 下面说法中正确的是 (D)

- A) 当声明的内容不同时, 声明同一个名字的两个类是允许的
- B) 不完全生命的类也可以实例化
- C) 声明同一个名字的两个类时, 要求声明的内容也相同
- D) 生命同一个名字的两个类是错误的

58. 下面对静态数据成员的描述中, 正确的是 (A)

- A) 静态数据成员是类的所有对象共享的数据
- B) 类的每个对象都有自己的静态数据成员
- C) 类的不同对象有不同的静态数据成员值
- D) 静态数据成员不能通过类的对象调用

59. 下列关于对象数组的描述中, 错误的是 (D)

- A) 对象数组的下标是从 0 开始的
- B) 对象数组的数组名是一个常量指针
- C) 对象数组的每个元素是同一个类的对象
- D) 对象数组只能赋初值, 而不能在定义后赋值

60. 对于下面定义类 A 在函数 f() 中将对象成员 n 的值修改为 60 的语句应该是 (C)

```
class A
{
public:
    A(int x) {n=x;}
    void SetNum(int n1) {n=n1;}
private
    int n;
}

int f()
{
    a*ptr=new A(60);
}
```

- A) A(60)
- B) SetNum(60)
- C) Ptr->SetNum(60)
- D) Ptr->n=60

61. 关于使用 const 关键字修饰的对象的说法正确的是 (A)

- A) 不能用一般的成员函数来访问
- B) 可以用一般的成员函数来访问
- C) 可以访问 const 成员函数及 volatile 成员函数
- D) 不能访问 const 成员函数

62. 下面是类 MyClass 的定义, 对定义 中各语句描述正确的是 (D)

```
class MyClass
{
private:
    int X, y, Z;
public;
coid MyClass(int a) {X=a;} file://①
int f(int a, int b) file://②
{
X=a;
Y=b;}
Int f(int a, int b, int c=0) file://③
{a;
Y=b;
Z=c;}
Static coid g() {X=10;} file://④
```

- A) 语句①是类 MyClass 的构造函数定义
- B) 语句②和语句③实现类成员函数的重载
- C) 语句④实现对类成员变量 X 的更新操作
- D) 语句①②③和④都不正确

63. 友元访问类的对象的成员时使用 (D)

- A) 类的成员名
- B) this 指针指向成员名
- C) “类名: 成员名”的形式
- D) “对象名. 成员名”的形式

64. 下列程序输出的结果正确的是 (B)

```
#include<iostream.h>
coid fun()
{static int n=10;
n++;
cout <<” n=” <<n<<endl;
}
void main()
{
for (int I=1;I<3;I++)
cun();
```

}

A) n=11 n=11

B) n=11 n=12

C) n=10 n=11

D) n=10 n=10

65. 下面类模板的定义中，不正确的是（ C ）

A) template<class T>

class A{T x,y};

B) template<class T>

class A{private:t x};

C) template<class T>

class A{public:T x};

D) template<class T>

class A{protected:t x};

66. 静态成员遵循类的其他成员所遵循的访问限制，除了（ B ）

A) 静态成员函数

B) 静态数据成员初始化

C) 私有静态数据成员

D) 公有静态成员函数

67. 空类的声明（ B ）是正确

A) class 类名;

B) class 类名{};

C) class 类名{};

D) class 类名

68. 可以用“px->*p”来访问类A的对象a的数据成员，则下面正确的是（ B ）

A) px是指向类数据成员的指针

B) p是指向类数据成员的指针

C) p是指向类的对象的指针

D) 无法确定指针类型

69. 关于类中数据成员的生存期的说法正确的是（ D ）

A) 与对象的生存期无关

B) 比对象的生存期长

C) 比对象的生存期短

D) 有对象的生存期决定

70. 类example的析构函数为（ D ）

A) example();

B) example()

C) ~example();

D) ~example()

71. 以知AA是一个类，则执行语句Aaa;后（ A ）

A) 自动调用无参构造函数

B) 自动调用有参构造函数

C) 自动调用拷贝构造函数

D) 产生一个对象 a, 但不调用任何函数

72. 对于类模板 tanytemp, 执行语句 tanytemp<int>iobject(23, 32); 后

(C)

A) 产生的模板类为 tanytemp

B) 产生的模板类为 iobject

C) 产生的模板类为 tanytemp<int>

D) 类模板实例化对象

73. 下列说法正确的是(C)

A) 类定义中只能说明函数成员的函数头, 不能定义函数体

B) 类中的函数成员只能可以在类体内定义, 不可能在类体外定义

C) 类中的函数成员在类体外定义时必须要与类声明在同一文件中

D) 在类体外定义的函数成员不能操作该类的私有数据成员

74. 类的构造函数被自动调用执行的情况是在定义该类的(C)

A) 成员函数时

B) 数据成员时

C) 对象时

D) 友元函数时

75. 说明友元函数使用关键字(A)

A) friend

B) static

C) const

D) colatile

76. 有关类的作用域, 下了说法中不正确的是(B)

A) 说明类时所使用的一对花括号形成所谓的类作用域

B) 类作用域不 C) 包含类汇总成员函数的作用域

D) 类作用域中说明的标 E) 识符只在类中可见

F) 在可能出现两义性的情况下, 必须使用作用域限定符” :: ”

77. 类的静态成员(C)

A) 是指 B) 静态数据成员

C) 是指 D) 静态函数成员

E) 为该类的所有对象共享

F) 遵循类的其他成员做遵循的所有访问权限

78. 允许访问类的所有对象的私有成员, 公有成员和保护成员的是(A)

A) 友元函数

B) 成员函数

C) 内联函数

D) 重载函数

79. 形如 X::X(X&) 的构造函数称为(D)

A) 拷贝函数

B) 复 C) 制函数

D) 初始化函数

E) 拷贝初始化构造函数

80. 以知 example 是已定义的类, 则程序结束前, 共存在(B)个对象.

```
#include<example.h>
```

```
main()
```

```
{example x(5);      example arr[2]={1, 2};
```

```
x=10;
```

```
x.print();
```

```
x=example(15);
```

```
x.print();
```

```
}
```

A) 3

B) 4

C) 7

D) 8

81. 所有在函数中定义的变量, 连同形式参数, 都属于(B)

A) 全局变量

B) 局部变量

C) 静态变量

D) 寄存器变量

82. 在 C++ 中, 封装是借助于(B)达到的.

A) 结构

B) 类

C) 数组

D) 函数

83. 下面不属于类的存取权限的是(B)

A) public

B) static

C) protected

D) private

84. 指向类 X 中类型为 type 的数据成员的指针的说明形式为(A)

A) type X::*pointer

B) x::type*pointer

C) type *X::pointer

D) X::*tyoe pointer

85. 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, (A)是正确的

A) void print() const

B) const void print()

C) void const print()

D) void print(const)

86. 关于 new 运算符的下列描述中, (D)是错误的.

- A) 它可以用来动态创建对象和对象数组
- B) 使用它创建对象或对象数组, 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) 使用它调用对象数组时不 E) 许指 F) 定初始值

87. 下列关于构造函数的描述中, 错误的是(C)

- A) 构造函数可以设置默认参数
- B) 构造函数在说明类变量时自动执行
- C) 构造函数可以对静态数据成员进行初始化
- D) 够找函数可以重载

88. 对于类 A, 语句 VOID(A::*P) (VOID); 表明(A)

- A) P 是一个指 B) 向类成员函数的指 C) 针
- D) P 是类 A 的一个成员
- E) P 是类 A 的一个对象
- F) P 是一个指 G) 向类对象的指 H) 针

89. 运算符 delete 删除一个动态对象时(C)

- A) 首先为该动态对象调用构造函数, 在释放其占用的内存
- B) 首先释放该动态对象占用的内存, 在为其调用构造函数
- C) 首先为该动态对象调用析构函数, 在释放其占用的内存
- D) 首先释放动态对象占用的内存, 在为其调用析构函数

第四、五章习题一选择题

1. 在 C++ 中的类有两种方法: 一种是类的实例化, 2. 即生成类的对象, 3. 并参与系统的运行; 另一种是通过_____B_____, 4. 派生出新的类。

- A) 复 B) 用
- C) 继承
- D) 单继承
- E) 多继承

5. 下面关于纯虚函数和抽象类的描述中, 6. _____B_____是错误的。

- A) 纯虚函数是一种特殊的虚函数, B) 它没有具体意义
- C) 一个基类中说明有纯虚函数, D) 该基类的派生类一定不 E) 再是抽象类
- F) 抽象类只能作为基类来使用, G) 其纯虚函数的定义由派生类给出
- H) 抽象类是指 I) 具有纯虚函数的类

7. 继承具有_____B_____, 8. 即当基类本身也是某一个类的派生类时, 9. 底层的派生类也会自动继承间接基类的成员。

- A) 规律性
- B) 传递性
- C) 重复 D) 性
- E) 多样性

10. 对虚函数的调用_____D_____。

- A) 一定使用动态联编
- B) 一定使用静态联编

- C) 必须使用动态联编
D) 不 E) 一定使用动态联编
11. 若类 A 和类 B 的定义如下:

```
class A
{
    int i, j;
    public:
void get();
file:///...
};

class B:A//默认为私有派生
{
    int k;
    public;
void make();
file:///...
};

void B:: make()
{
    k=i*j;
}
```

则上述定义中, ____D____是非法的表达式。

- A) void get();
B) int k;
D) void make();
D) k=i*j;

6. 下列描述中, ____D____是抽象类的特性。

- A) 可以说明虚函数
B) 可以定义友元函数
C) 可以进行构造函数重载
D) 不能说明其对象

7. 在多继承中, 公有私生和私有派生对于基类成员的派生类中的可访问性与单继承的规则____A____。

- A) 完全相同
B) 完全不同
C) 部分相同, 部分不同
D) 以上都不对

8. ____A____是一个在基类中说明虚函数, 它在该基类中没有定义, 但要求任何派生类中的可访问性的规定是相同的。

- A) 纯虚函数
B) 虚析构造函数
C) 虚构造函数

D) 静态成员函数

9. 下列对派生类的描述中, _____D_____是错误的。

A) 一个派生类可以作另一个派生类的基类

B) 派生类至少有一个基类

C) 派生类的成员除了它自己的成员外, 包含了它的基类的成员

D) 派生类中继承的基类成员的访问权限到派生类保持不变

10. 实现运行时的多态性要使用_____D_____。

A) 构造函数

B) 析构函数

C) 重载函数

D) 虚函数

11. 派生类的对象对它的基类成员中 _____A_____是可以访问的。

A) 公有继承的公有成员

B) 公有继承的私有成员

C) 公有继承的保护成员

D) 私有继承的公有成员

12. 关于函数的描述中, _____C_____是正确的。

A) 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

B) 基类中说明了虚函数后, 派生类中其对应的函数一定要说明为虚函数

C) 虚函数是一个成员函数

D) 虚函数是一个 static 类型的成员函数、

13. 对基类和派生类的关系描述中, _____B_____是错误的。

A) 派生类是基类的具体化

B) 派生类是基类的子集

C) 派生类是基类定义的延续

D) 派生类是基类的组合

14. 关于动态联编的下列叙述中, _____D_____是错误的。

A) 动态联编是以虚函数为基础的

B) 动态联编调用虚函数操作是指向对象的指针或引用

C) 动态联编是在运行时确定所调用的函数代码的

D) 动态联编是在编译时确定操作函数的

15. 派生类的构造函数的成员初始化列表中, 不能包含_____C_____。

A) 基类的构造函数

B) 派生类中子对象的初始化

C) 基类的子对象初始化

D) 派生类中一般数据成员的初始化

16. 如果一个类至少有一个纯虚函数, 那么就称该类为_____A_____。

A) 抽象类

B) 派生类

C) 纯基类

D) 以上都不对

17. 设置虚基类的目的是____B____。

- 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

18. 在下面程序中，编译时出现的错误是____C_____。

```
class A file:/// (1)
{
    public; file:/// (2)
    A()
    {
        func(); file:/// (3)
    }
    virtual void func()=0; file:/// (4)
};
```

- A) (1)
- B) (2)
- C) (3)
- D) (4)

19. 关于多继承二义性的描述中，错误的是____C_____。

- A) 一个派生类的基类中都有某个同名成员，在派生类中对这个成员的访问可能出现二义性
- B) 解决二义性的最常用的方法是对成员名的限定法
- C) 基类和派生类同时出现的同名函数，也存在二义性问题
- F) 一个派生类是从两个基类派生出来的，G) 而 H) 这两个基类又有一个共同 I) 的基类，J) 对该基类成员进行访问时，K) 可能出现二义性

20. 用于类中虚成员函数说明的关键字（ A ）

- A. virtual B. public C. protected D. private

21. 下列对派生类的描述中，错误的是（ D ）

- A. 派生类的缺省继承方式是 PRIVATE
- B. 派生类至少有一个基类
- C. 一个派生类可以作为另一个派生类的基类
- D. 派生类只继承了基类的公有成员和保密成员

22. 编译时的多态性通常使用（ C ）获得。

- A. 继承 B. 虚函数 C. 重载函数 D. 析构函数

23. 下列说法中错误的是（ D ）

- A. 公有继承时基类中的 public 成员在派生类中是 public 的
- B. 公有继承时基类中的 private 成员在派生类中是 private 的
- C. 私有继承时基类中的 public 成员在派生类中是 private 的
- D. 私有继承时基类中的 public 成员在派生类中是 protected 的

24. 下面哪个基类中的成员函数表示纯虚函数（ C ）

A. virtual void vf(int) B. void vf(int)=0

C. virtual void vf()=0 D. virtual void vf(int)||

25. 下面叙述不正确的是 (C)

A. 基类的公有成员在派生类中仍然是公有的

B. 派生类一般都用公有派生

C. 对基类成员访问必须是无二义性的

D. 赋值兼容规则也使用于多重继承的组合

26. 下列虚基类的声明中正确的是 (D)

A. class virtual B:public A B. virtual class B:public

B. class B:public A virtual D. class B: virtual public A

27. 下面叙述不正确的是 (A)

A. 基类的保护成员在派生类中仍然是保护的

B. 基类的保护成员在公有派生类中仍然是保护的

C. 基类的保护成员在私有派生类中仍然是私有的

D. 对基类成员的访问必须是无两义性

28. 带有虚基类的多层派生类构造函数的成员初始化列表中都要列出虚基类构造函数, 这样将对虚基类的子对象初始化 (D)

A. 与虚基类下面的派生类个数有关 B. 多次 C. 二次 D. 一次

29. 关于保护成员的说法正确的是 (B)

A. 在派生类中仍然是保护的

B. 具有私有成员和公有成员的双重角色

C. 在派生类中是私有的

D. 在派生类中是公有的

30. 下列关于子类型的描述中, 错误的是 (A)

A. 子类型关系是可逆的

B. 公有派生类的对象可以初始化基类的引用

C. 只有在公有继承下, D. 派生类是基类的子类型

E. 子类型关系是可传递的

31. 若要强制 C++ 对虚函数的调用使用静态联编, 则在调用中对该函数使用 (A)

A. 成员名限定 B. 指针 C. 引用 D. VIRTUAL 关键字

32. 模板和继承的关系是 (A)

A. 模板反映的是不同 C. 类型的相同 D. 操作; 而 E. 同 F. 一类系的各类处理的数据很多情况下是同 G. 一类型的

H. 继承类较模板类有更多的动态特征

C. 模板类系的成员较继承类系的成员稳定。

D. 模板和继承的方法都可以派生出一个关系, 因而本质上它们是相同的。

33. 下面 (B) 的叙述不符合赋值兼容规则。

A. 派生类的对象可以赋值给基类的对象

B. 基类的对象可以赋值给派生类的对象

C. 派生类的对象可以初始化基类的对象

D. 派生类的对象的地址可以赋值给指向基类的指针

34. C++有 (B) 种联编。

- A. 1
- B. 2
- C. 3
- D. 4

35. 对于下列程序段，没有二义性的表达式是 (C)

```
class A{public:int f();};  
class B{public:int g();int f();};  
class C:public A,public B{ public : int g( ) ; h( ) ; } ;  
C obj;
```

- A. obj.f()
- B. obj.A :: g()
- C. obj.B :: f()
- D. obj.B :: h()

36. 下面叙述错误的是 (C)

- A. 派生类可以使用 private 派生
- B. 对基类成员的访问必须是无二义性的
- C. 基类成员的访问能力在派生类中维持不变
- D. 赋值兼容规则也适用于多继承的组合

37. 派生类的对象与基类的对象的关系是 (D)

- A. 属于前者而不属于后者
- B. 属于前者，不一定属于后者
- C. 不属于前者则一定属于后者
- D. 属于前者则一定属于后者

38. 对于下面程序段，表达式正确的是 (A)

```
class Base {public:int b;};  
class Base1:public base{ };  
class Base2:public base{ };  
class Derived:public base1,public base2{public:int fun ( );};  
Derived d;
```

- A. d.Base1::b
- B. d.Base ::b
- C. d.b
- D. d.Base :: fun()

39. 下列程序中横线处正确的语句应该是 (C)

```
#include<iostream>  
using namespace std;  
class Base  
{  
    public;  
    void fun( ){cout<< "Base :: fun" << endl;}
```

```
};
class Derived : public Base
{
public:
void fun( )
{
_____//显示调用基类的函数 fun( )
cout << "Derived :: fun" << endl;
}
};
```

A.fun().

B.Base.fun()

C.Base :: fun()

D.Base ->fun()

40. 虚函数必须是类的 (A)

A. 成员函数

B. 友元函数

C. 构造函数

D. 析构造函数

41. 不能说明为虚函数的是 (B)

A. 析构造函数

B. 构造函数

C. 类的成员函数

D. 以上都不对

42. 不同的对象可以调用相同名称的函数，并可导致完全不同的行为的现象称为 (A)

A. 多态性

B. 抽象

C. 继承

D. 封装

43. 若析构造函数是虚函数，则 delete 对析构造函数的调用采用 (B)

A. 静态联编

B. 动态联编

C. 重载函数

D. 多重函数

44. 关于虚函数，下列表述正确的是 (B)

A. 如果在重定义虚函数时使用 virtual，则该重定义函数仍然是虚函数

B. 虚函数不得声明为静态函数

C. 虚函数不得声明为另一个类的友元函数

D. 派生类必须重新定义基类的虚函数

45. 下列程序段中具有相同值的是 (A)


```

class Base {int b;};
class Base1:virtual public base{int b1;};
class Base2:virtual public base{int b2;};
class derived:public base 1,public base 2{int b3;};
derived d;

```

A. d.b 与 d.base1::b

B. d.base1::b 与 d.base1::b1

C. d.b 与 d. b3

D. d.base2::b 与 d.base2::b2

46. 对于公有派生，它的“访问控制”是（ B ）

A. private

B. public

C. protected

D. inaccessible

47. 关于纯虚函数，下列表述正确的是（ B ）

A. 纯虚函数是给出实现版本（即无函数体定义）的虚函数

B. 纯虚函数的声明总是“=0”结束

C. 派生类必须实现基类的纯虚函数

D. 含有纯虚函数的类一定是派生类

48. 若派生类中没有对基类的虚函数进行重定义，则它继承基类中的（ C ）

A. 访问权限

B. 保护成员

C. 虚函数

D. 成员函数

49. 基类和派生类可以分别叫做（ B ）

A. “大类”和“小类”

B. “父类”和“子类”

C. “小类”和“大类”

D. “子类”和“父类”

50. 多态调用的指（ C ）

A. 以任何方式调用一个虚函数

B. 以任何方式调用一个纯虚函数

C. 借助于指向对象的基类指针或引用调用一个虚函数

D. 借助于指向对象的基类指针或引用调用一个纯虚函数

51. 若派类的成员函数不能直接访问基类中继承来的某个成员，则该成员一定是基类中的（ A ）

A. 私有成员

B. 公有成员

C. 保护成员

D. 保护成员或私有成员

52. 对基类和派生类的描述中错误的是（ B ）

A. 派生类是基类的具体化

- B. 基类继承了派生类的属性
- C. 派生类是基类定义的延续
- D. 派生类是基类的特殊化

53. 关于抽象类，下列表述正确的是 (D)

- A. 抽象类的成员函数中至少有一个没有实现的函数（即无函数体定义的函数）
- B. 派生类必须实现作为基类的抽象类中的纯虚函数
- C. 派生类不可能成为抽象类
- D. 抽象类不能用于定义对象

54. 类 B 是类 A 的公有派生类，类 A 和类 B 中都定义了虚函数 func()，p 是一个指向类 A 对象的指针，则 p->A::func() 将 (A)

- A. 调用类 A 中是函数 func()
- B. 调用类 B 中是函数 func()
- C. 根据 P 所指的对象类型而确定调用 A 中或类 B 中的函数 func()
- D. 既调用类 A 中函数，也调用类 B 中的函数

55. 对于类定义

```
class A{
public:
    virtual void func1() {}
    void func2() {}
};

class B:public A{
PUBLIC:
    Void func1() {cout<<" class Bfunc1" <<endl;}
    Cirtual void func2() {cout<<" class B func2" <<endl;}
};
```

下面正确的叙述是 (C)

- A. A: : func2() 和 B: : func1() 都是虚函数
- B. A: : func2() 和 B: : func1() 都不是虚函数
- C. B: : func1() 是虚函数，A: : func2() 不是虚函数
- D. B: : func1() 不是虚函数，A: : func2() 是虚函数

56. 对于多重继承，有 (D)

- A. 一个派生类只能有一个基类
- B. 一个基类只能产生一个派生类
- C. 一个基类必须产生多个派生类
- D. 一个派生类可有多个基类

57. 下列程序段中，错误的语句是 (C)

```
class A{
public:virtual void f()=0;           file://\(1\)
    void g() {f();}                file://\(2\)
    A() {f();}                     file://\(3\)
};
```

A. (1)

B. (2)

C. (3)

D. (1) 和 (2)

58. 下面程序段中, 说法正确的是 (D)

```
class location{
public :
int GetX();
};

class rectangle:private location {
public:
int GetW();
};
```

A. 类 rectangle 私有派生类 location

B. 类 rectangle 中的 public 成员在类 location 中是私有的

C. 类 location 中的 public 成员在类 rectangle 中是不可访问的

D. void f() {

Rectangle r;

int X=r .GetX();

}

其中 int X = r .GetX();是非法语句

59. 在公有派生情况下, 有关派生类对象和基类对象的关系的关系, 不正确的叙述是 (C)

A. 派生类的对象可以赋给基类对象

B. 派生类的对象可以初始化基类的引用

C. 派生类的对象可以直接访问基类的成员

D. 派生类的对象的地址可以赋给指向基类的指针

60. 可以用 p.a 的形式访问派生类对象 p 的基类成员 a 其中 a 是 (D)

A. 私有继承的公有成员

B. 公有继承的私有成员

C. 公有继承的保护成员

D. 公有继承的公有成员

61. 建造新的派生类是 (C) 的实质。

A. 派生

B. 抽象

C. 继承

D. 封装

62. 动态联编所支持的多态性称为 (D)

A. 虚函数

B. 继承

C. 编译时的多态性

D. 运行时的多态性

第七章习题一、选择题

1. 下列运算符中, A_____运算符在 C++中不能重载。

- A) ? :
- B) +
- C) -
- D) <=

2. 在 C++中, 打开一个文件就是将这个文件与一个 A_____建立关联; 关闭一个文件就是取消这种关联。

- A
- A) 流
- B) 类
- C) 结构
- D) 对象

3. 下列关于运算符重载的描述中, D_____是正确的。

- A) 运算符重载可以改变操作数的个数
- B) 运算符重载可以改变优先级
- C) 运算符重载可以改变结合性
- D) 运算符重载不可以改变语法结构

4. 若定义 `cin>>str;`

当输入

Microsoft Visual Studio 6.0!

所得的结果是 `str=_B_____`。

- A) Microsoft Visual Studio 6.0!
- B) Microsoft
- C) Microsoft Visual
- D) Microsoft Visual Studio 6.0

5. 有元运算符 `obj1>obj2` 被 C++编译器解释为_A_____。

- A) `operator>(obj1,obj2)`
- B) `>(obj1,obj2)`
- C) `obj2.operator>(obj1)`
- D) `obj1.operator(obj2)`

6. 进行文件操作时需要包含_B_____头文件。

- A) `iostream`
- B) `fstream`
- C) `stdio`
- D) `stdlib`

7. 当使用 `ifstream` 流类定义一个流对象并打开一个磁盘文件时, 文件的隐含打开方式为_A_____。

- A) `ios::in`
- B) `ios::out`
- C) `ios::trunc`
- D) `ios::binary`

8. 使用如 `setw()` 的操作符对数据进行格式输出时, 应包含_____D_____文件。

- A) iostream
- B) fstream
- C) stdio
- D) ioman