

## 面向对象程序设计模拟试卷二

一. 单项选择题 (从下列各题四个备选答案选出一个正确答案, 并将其代号填在题干前的括号内。答案选错或未作选择者, 该题不得分。每小题1分, 共10分)

- ( ) 1. 以下有关析构函数的叙述, 选择正确的填入括号内。
- A. 可以进行重载可以定义为虚函数
  - B. 不能进行重载可以定义为虚函数
  - C. 可以进行重载不能定义为虚函数
  - D. 不能进行重载不能定义为虚函数
- ( ) 2. 以下有关函数缺省参数的出现位置, 选择正确的叙述填入括号内。
- A. 必须全部出现在函数参数表的左部
  - B. 必须全部出现在函数参数表的右部
  - C. 必须全部出现在函数参数表的中间
  - D. 都不对
- ( ) 3. 关于两个同名函数重载的叙述, 选择最为正确的填入括号内。
- A. 两个函数的参数个数不同
  - B. 两个函数对应的参数类型不同
  - C. 两个函数的参数个数不同或对应的参数类型不同
  - D. 都不对
- ( ) 4. 关于inline保留字的用途, 选择正确的叙述填入括号内。
- A. 只能用于定义成员函数
  - B. 只能用于定义非成员函数
  - C. 可以定义成员函数及非成员函数
  - D. 都不对
- ( ) 5. 关于类的构造函数的定义位置, 将最为正确的叙述填入括号内。
- A. 只能在private下定义
  - B. 只能在protected下定义
  - C. 只能在public下定义
  - D. 定义位置没有限制
- ( ) 6. 如下修饰类体中函数的返回类型, 将正确的用法填入括号内。
- A. 同时使用static和friend

- B. 同时使用static和virtual
  - C. 同时使用friend和virtual
  - D. 不同时使用上述三个保留字中的任意两个
- ( ) 7. 在如下关于继承的叙述中, 选择正确的填入括号内。
- A. 私有继承用于实现类之间的ISA关系
  - B. 保护继承用于实现类之间的ISA关系
  - C. 公有继承用于实现类之间的ISA关系
  - D. 都不对
- ( ) 8. 对于用 union 定义的类 A, 选择正确的叙述填入括号内。
- A. 类A可以作为某个类的基类
  - B. 类A可以作为某个类的派生类
  - C. 类A既不能作基类又不能作派生类
  - D. 类A既可以作基类又可以作派生类
- ( ) 9. 有关在类中声明友元函数的叙述, 将正确的答案填入括号内。
- A. 只能在private下声明
  - B. 只能在protected下声明
  - C. 只能在public下声明
  - D. 声明位置没有限制
- ( ) 10. 在构造派生类 A 的对象时, 选择正确的叙述填入括号内。
- A. 最先构造派生类A的基类
  - B. 最先构造派生类A的虚基类
  - C. 最先构造派生类A的数据成员对象
  - D. 都不对

二. 多项选择题 (从下列各题四个备选答案选出二至四个正确答案, 并将其代号填在题干前的括号内。答案选错或未选全者, 该题不得分。每小题2分, 共10分)

- ( ) 11. 将符号为分隔符的编号填入括号内。
- A. m              B. +              C. {              D. =
- ( ) 12. 将用于定义类的保留字的编号填入括号内。
- A. class          B. struct          C. int              D. union
- ( ) 13. 将用于释放p=new int[8]的内存的编号填入括号内。
- A. free (p)      B. delete p      C. delete []p      D. sizeof p

( ) 14. 将派生类成员函数可访问的基类成员编号填入括号内。

A. private    B. protected    C. public    D. 所有成员

( ) 15. 将不能重载的运算符函数的编号填入括号内。

A. ? :    B. ::    C. %    D. <<

三. 填空题 (阅读以下程序并填空。每题 1 分, 共 10 分)

```
class A{
    int a;
protected:
    int b;
public:
    int c;
}a;
class B: protected A{
    int d;
protected:
    int e;
public:
    A::c;
    int f;
}b;
class C: A{
    int g;
protected:
    int h;
public:
    int c;
}c;
struct D: B, C{
    int j;
protected:
    B::b;
}d;
void main(void){
    int x;
}
```

16. 对于 main, 对象 a 的私有成员为\_\_\_\_\_。

17. 对于 main, 对象 a 的保护成员为\_\_\_\_\_。

18. 对于 main, 对象 a 的公有成员为\_\_\_\_\_。

19. 对于 main, 对象 b 的私有成员为\_\_\_\_\_。

20. 对于 main, 对象 b 的保护成员为\_\_\_\_\_。

21. 对于 main, 对象 c 的保护成员为\_\_\_\_\_。

22. 对于 main, 对象 c 的公有成员为\_\_\_\_\_。

23. 对于 main, 对象 d 的保护成员为\_\_\_\_\_。

24. 对于 main, 对象 d 的公有成员为\_\_\_\_\_。

25. main 应使用语句 x=\_\_\_\_\_取对象 d 的基类 C 的成员 c 的值。

四. 名词解释 (每小题 3 分, 共 15 分)

26. 标识符:

27. 重载:

28. 异常:

29. 类型强制转换:

30. 抽象类:

五. 判断改错题 (阅读以下程序并判断各题是否正确, 对正确的就在其题号前打√; 错误的在其题号前打×, 并在题干后的括号内更正。每小题 3 分, 共 15 分)

```
#include <iostream.h>
struct A{A(){ cout<<'A';}};
struct B{B(){ cout<<'B';}};
struct C: A{C(){ cout<<'C';}};
struct D: virtual B, C{D(){ cout<<'D';}};
struct E: A {
    C c;
    E(): c(){ cout<<'E';}
};
struct F: virtual B, C, D, E{
    F(){ cout<<'F';}
};
void main( ){ /*.....*/ }
```

- ( ) 31. 如 main 定义 A a, 则输出为 A。
- ( ) 32. 如 main 定义 C c, 则输出为 BAC。
- ( ) 33. 如 main 定义 D d, 则输出为 BACD。
- ( ) 34. 如 main 定义 E e, 则输出为 AACE。
- ( ) 35. 如 main 定义 F f, 则输出为 BCDEF。

六. 计算题 (阅读如下程序并计算执行结果。共20分)

```
int x=2, y=x+30;
```

```

struct A {
    static int x;
    int y;
public:
    operator int() { return x+y; }
    A operator ++(int) { return A(x++, y++); }
    A(int x::x+2, int y::y+3) { A::x=x; A::y=y; }
    int &h(int &x);
};

int &A::h(int &x)
{
    for(int y=1; y!=1 || x<201; x+=11, y++)
        if(x>200) { x-=21; y-=2;}
    return x-=10;
}

int A::x=23;

void main() {
    A a(54, 3), b(65), c;
    int i, &z=i, A::*p=&A::y;
    i=b.x; //..... ①
    z=a.x; //..... ②
    i=c.*p; //..... ③
    i=a++; //..... ④
    i::x+c.y; //..... ⑤
    i=a+b; //..... ⑥
    b.h(i)=7; //..... ⑦
}

```

36. 语句①执行结束后 i=           。 (2 分)
37. 语句②执行结束后 i=           。 (3 分)
38. 语句③执行结束后 i=           。 (3 分)
39. 语句④执行结束后 i=           。 (3 分)
40. 语句⑤执行结束后 i=           。 (3 分)
41. 语句⑥执行结束后 i=           。 (3 分)
42. 语句⑦执行结束后 i=           。 (3 分)

七. 设计题 (20分)

43. 如下复数类说明了五个成员函数, 请详细定义这五个函数的函数体。(每个定义4分)

```
class COMPLEX{ //real, imag 分别为复数的实部和虚部,  
    double real, imag; //若  $x=3+5i$ , 则  $x.real=3$ ,  $x.imag=5$   
public:  
    COMPLEX(double r=0, double i=0);  
    COMPLEX& operator += (const COMPLEX &c);  
    COMPLEX& operator ++ ( ); //复数的实部和虚部加 1  
    COMPLEX operator - ( ) const; //将实部和虚部的符号求反  
    COMPLEX operator + (const COMPLEX &c) const;  
};
```



