## 第1题

1. */\**
2. 1:对于如下类型声明，分别给出类A、B、C可访问的成员及其访问权限\*/
3. class A {
4. int a1, a2;
5. protected:
6. int a3, a4;
7. public:
8. int a5, a6;
9. };
10. class B : A {
11. int b1, b2;
12. protected:
13. using A::a3;
14. int b3, b4;
15. public:
16. using A::a6;
17. int b5, b6;
18. ;
19. };
20. struct C : public B {
21. int c1, c2;
22. using B::a3;
23. protected:
24. int c3, c4;
25. public:
26. int c5, c6;
27. int b5, b6, a6;
28. };

## 答：

### 一、类 A

1.私有成员 ：a1, a2//仅类 A 的成员函数可以访问。

2.受保护成员 ：a3, a4//类 A 及其派生类可以访问。

3.公有成员 ：a5, a6//类 A、派生类以及外部代码都可以访问

### 二、类 B

类 B 继承自 A，默认是**私有继承**

#### 2.1 继承后 A 的成员在 B 中的访问权限：

1. A 的**私有成员** (a1, a2)：

不可见

2.A 的**保护成员** (a3, a4)：

在私有继承的情况下，本来应该变为私有成员，但在类 B 中**显式地 using A::a3** 将 a3 的访问权限提升为protect，所以：

a3 在类 B 中为protect。

a4 在类 B 中仍为private。

3.A 的 **公有成员** (a5, a6)：

在私有继承的情况下，变为 私有成员，但在类 B 中**显式地 using A::a6** 将 a6 的访问权限提升为public，所以：

a5 在类 B 中为private。

a6 在类 B 中为public。

#### 2.2 类 B 自己的成员：

1.私有成员 ：b1, b2

2.保护成员：b3, b4

3.公有成员 (public)：b5, b6

访问权限描述同类A，不再重述

### 三、类 C

类 C 使用 **public 继承** B，因此 B 的公有和受保护成员在 C 中保持原有访问权限。

#### 3.1 继承后 B 的成员在 C 中的访问权限：

1.B 的 私有成员 (b1, b2)：不可见

2.B 的 受保护成员 (a3, b3, b4)：在 C 中仍为保护成员。

3.B 的 公有成员 (a6, b5, b6)：在 C 中仍为 公有成员**(C中重新声明了它们，默认访问C的成员，如果需要访问继承来的需要加B::限定**

#### 3.2 类 C 自己的成员：

1.私有成员 (private)：无（**结构体的默认是 public**）。

2.受保护成员 (protected)：c3, c4

3.公有成员 (public)：c1, c2, c5, c6,b5,b6,a6

## 第2题

1. */\**
2. 2: 根据以下代码，回答问题或完成类的定义
3. \*/
4. class A {
5. int i;
6. public:
7. A(int i);
8. void print();
9. };
10. */\**
11. 2.1 请在class A的类体外实现构造函数A(int i)，要求必须在成员初始化列表初始化i成员
12. \*/
13. class B : public A {
14. int j;
15. public:
16. B(int i, int j);
17. };
18. */\**
19. 2.2 请在class B的类体外实现构造函数B(int i,int j)
20. 要求必须在成员初始化列表初始化j成员
21. \*/

答：

A::A(int i) : i(i) {}

B::B(int i, int j) : A(i), j(j) {}

## 第3题

1. */\**
2. 3: 根据以下代码，回答问题或完成类的定义
3. \*/
4. class A {
5. protected:
6. int i{ 0 };
7. public:
8. A(int i = 0) {}
9. };
10. class B : public A {
11. int i;
12. public:
13. */\**
14. 3.1 B类构造函数没有显式构造基类A对象，编译是否可以通过？请说明原因
15. \*/
16. B(int i) { this->i = i; }
17. int sum(int i) {
18. */\**
19. 3.2 请完成sum函数的具体实现，要求对局部变量i、A类继承的i、B类的i求和并返回
20. \*/
21. }
22. };

答：(1)可以通过，A中i有默认值0,即使没有定义构造函数也可以编译通过。

(2) int sum(int i) {

return i + A::i + this->i;

}

其中i为传给sum的i，this->i为B类的i

## 第4题

1. */\**
2. 4: 根据以下代码，回答问题
3. \*/
4. class A {
5. int i;
6. public:
7. A(int x) :i(x) { }
8. };
9. class B {
10. int j;
11. public:
12. B() :j(0) {}
13. B(int x) :j(x) { }
14. };
15. class C : public A {
16. const int k;
17. A a;
18. B b;
19. A& ra;
20. public:
21. */\**
22. 下面类C的构造函数中正确的有哪些？错误的有哪些？错误的请说明原因
23. \*/
24. C(int v) :A(v), k(v), a(v), ra(a) {}          *//1*
25. C(int v):A(v),a(v),ra(a){ k = v;}             *//2*
26. C(int v):A(v),k(v),a(v),b(v),ra(a){}         *//3*
27. C(int v):k(v),a(v),b(v),ra(a){}               *//4*
28. };

答：1，3正确。2，4错误

2错在const 成员必须通过初始化列表中初始化，不能在构造函数体内部进行赋值。

4错在C继承A，但A没有默认构造函数 ，因此必须在初始化列表中显式调用A的构造函数

## 第5题

1. */\**
2. 5:下面程序，写出指定语句的输出结果，并解释原因。
3. \*/
4. class A {
5. public:
6. virtual void f() { std::cout << "A::f()" << std::endl; }
7. virtual void f(double x) { std::cout << "A::f(double)" << std::endl; }
8. static void g() { std::cout << "A::g()" << std::endl; }
9. A() = default;
10. virtual ~A() = default;
11. };
12. class B : public A {
13. public:
14. virtual void f() { std::cout << "B::f()" << std::endl; }
15. virtual void f(double x) { std::cout << "B::f(double)" << std::endl; }
16. static void g() { std::cout << "B::g()" << std::endl; }
17. static void g(int) { std::cout << "B::g(int)" << std::endl; }
18. B() = default;
19. virtual ~B() = default;
20. };
21. */\**
22. 请说明当执行下面的test函数后，
23. 每一条语句的情况，如果可以运行请给出运行结果；如果编译出错请说明原因。
24. \*/
25. void test1(A& o) {
26. o.f();        *//语句1*
27. o.f(1.0);     *//语句2*
28. o.g();        *//语句3*
29. *// o.g(1);       //语句4*
30. *// ((B)o).f();   //语句5*
31. *// ((B)o).g();   //语句6*
32. }
33. void test2(A& o) {
34. o.f(1.0f);    *//语句7*
35. o.f(1.0);     *//语句8*
36. *//  o.g(1);       //语句9*
37. o.g();        *//语句10*
38. ((A)o).g();   *//语句11*
39. }
40. void test() {
41. B b;
42. test1(b);
43. test2(b);
44. }

答：

1. 输出B::f()

f是虚函数，根据传入的对象类型，调用了B中的f

1. 输出B::f(double)

原因同上，根据函数重载，传入double参数，调用另一个f

1. 输出A::g()

由于 g() 是静态成员函数，它的调用不依赖于对象的动态类型（运行时类型），而是依赖于调用点处的静态类型.在 test1 函数中，参数 o 的静态类型是 A&，即使它实际上引用的是一个 B 类型的对象。因此，当调用 o.g(); 时，编译器会查找类型 A 中的静态成员函数 g()

1. **报错**，原因同(3)，编译时g为A::g()，而A中的g()是无参数的。
2. **报错**，没有定义A到B的转换 no known conversion for argument 1 from 'A' to 'const B&'
3. **报错**，没有定义A到B的转换 no known conversion for argument 1 from 'A' to 'const B&'
4. B::f(double)

动态绑定，具体原因如上。1.0f自动转换为double类型

1. B::f(double)

动态绑定，具体原因如上

1. **报错**，具体原因如(4)
2. 输出A::g()

静态绑定，具体原因如(3)

1. 输出A::g()

静态绑定，具体原因如(3),强制转换无意义

## 第6题

1. */\**
2. 6: 下面是类A、B、C的定义
3. \*/
4. class A {
5. public:
6. virtual void fa() {}
7. virtual void fb() = 0;
8. virtual void fc() = 0;
9. A() = default;
10. virtual ~A() = default;
11. };
12. class B : public A {
13. public:
14. virtual void fb() {}
15. };
16. class C : public B {
17. public:
18. */\**
19. 6-1: 如果C想成为一个具体类，则C必须要实现的方法是什么
20. \*/
21. };
22. */\**
23. 6-2: 请指出下面代码有错误的地方，并说明原因:
24. \*/
25. void f(A& o) {}
26. A& f(A\* p) { return \*p; }
27. void f(A o) {}

答：

(1)**必须实现纯虚函数fc**

因为A中有两个纯虚函数fb,fc,子类B已经实现了一个，C继承B，必须实现剩下的

(2)

**第三行错误**，抽象类A不能创造实例，只能通过引用或指针的形式传参

## 第7题

1. */\**
2. 7: 请给出下面程序的输出结果
3. \*/
4. class A {
5. public:
6. virtual void draw() { std::cout << "Draw A" << std::endl; }
7. virtual void display() { draw(); std::cout << "Display A" << std::endl; }
8. A() = default;
9. virtual ~A() = default;
10. };
11. class B : public A {
12. public:
13. virtual void draw() { std::cout << "Draw B" << std::endl; }
14. virtual void display() { A::display(); std::cout << "Display B" << std::endl; }
15. B() = default;
16. virtual ~B() = default;
17. };
18. void test() {
19. std::unique\_ptr<A> ptr = std::make\_unique<B>();
20. ptr->display();
21. }

答：输出

Draw B

Display A

Display B

首先调用 A::display()

其中又调用了draw()，根据实际类型进行了动态绑定

最后displayB