

第十四章测试

1. 单选题：

错误的是 ()。

- A. 派生类中继承的基类成员的访问权限到派生类保持不变。
- B. 一个派生类可以作为另一个派生类的基类。
- C. 派生类的成员除了其自身的成员外，还包含了其基类的成员。
- D. 派生类至少有一个基类。

答案： A

解释： 在派生类中，继承的基类成员的访问权限会根据继承类型发生变化。对于公有继承，基类的公有和保护成员可以在派生类中访问，但对于私有继承或保护继承，访问权限会有所不同。

2. 单选题：

在公有继承中，派生类成员函数不可以访问()。

- A. 派生类中的私有成员
- B. 基类中的保护成员
- C. 基类中的公有成员
- D. 基类中的私有成员

答案： D

解释： 公有继承中，派生类可以访问基类的公有成员和保护成员，但不能直接访问基类的私有成员。

3. 单选题：

派生类可以访问其基类的 ()。

- A. 私有派生
- B. 保护成员
- C. 公有成员
- D. 公有和保护成员

答案： D

解释： 派生类可以访问基类的公有成员和保护成员，但无法直接访问基类的私有成员。

4. 单选题：

```
1 #include "iostream"using namespace std;
```

```

2  class A
3  {
4  char c;
5  public:
6  A(char n)
7  {c=n;}
8  ~A()
9  {cout<<c;}
10 };
11 class B:public A
12 {
13 char c;
14 public:
15 B(char n):A(n+1)
16 {c=n;}
17 ~B()
18 {cout<<c;}
19 };
20 int main(){
21 B b('X');
22 b.~B();
23 system("pause");
24 return 0;
25 }

```

答案：YX

解释：在析构时，首先调用基类 A 的析构函数打印基类的字符，然后调用派生类 B 的析构函数打印派生类的字符。由于 B 是继承自 A，所以析构顺序是：基类先析构，派生类后析构。

- ☐ A. XY
- ☒ B. YX
- ☐ C. Y
- ☐ D. X

✗0.00/2.00

正确答案：A 你错选为B

提问

5. 单选题：

假设在以下派生类的 fun 函数中调用了基类的 fun 函数，则空白处应填写（）。

```
1 class AA
2 {public: void fun(){cout<<"aa fun called"<<endl;}}
3 };
4
5 class BB:public AA
6 { public:void fun()
7     {
8         -----;
9         cout<<"bb fun called"<<endl;
10    }
11 };
```

答案： A

解释： 基类的 `fun` 函数应通过 `AA::fun()` 来调用。

6. 单选题：

以下不是实现动态联编的必要条件的是（ ）。

- A. 虚函数操作的是基类对象
- B. 要有虚函数的支持
- C. 虚函数操作的是基类指针
- D. 虚函数操作的是基类引用

答案： A

解释： 动态联编（多态）需要虚函数支持，并且通过基类的指针或引用来调用。虚函数操作基类对象并不是实现动态联编的必要条件。

7. 判断题：

在程序中不能声明抽象类对象。

答案： A

解释： 抽象类不能直接实例化对象。抽象类必须被派生类继承并实现所有纯虚函数才能创建对象。

8. 判断题：

派生类继承基类中除构造函数和析构函数以外的所有成员。

答案： A

解释： 派生类会继承基类的成员，但构造函数和析构函数不会被继承。

9. 填空题：

一个派生类只有一个直接基类的情况称为（ ）继承。

答案： 单继承

解释： 单继承指的是派生类只有一个直接基类。

10. 填空题：

面向对象程序设计的（ ）机制提供了重复利用程序资源的一种途径。

答案： 继承

解释： 继承机制是面向对象程序设计中用于代码复用的方式。