

习 题

一、选择题

(注：涉及访问控制的题只考虑基于继承方式的控制)

1. 在任何继承方式下，派生类的自有成员函数都不能访问原基类的某些成员，这些成员的原访问属性必然是（ ）。

- A. 公有或保护
B. 保护或私有
C. 私有或继承的私有
D. 继承的私有或公有

2. 在任何继承方式下，派生类的自有成员函数都能够访问原基类的某些成员，这些成员的原访问属性必然是（ ）。

- A. 公有或保护
B. 保护或私有
C. 私有或继承的私有
D. 继承的私有或公有

3. 有如下程序：

```
#include <iostream>
using namespace std;
class Base {
public:
    void fun() { cout<<"Base::fun"<<endl; }
};
class Derived : public Base {
public:
    void fun()
    { _____ // 显式调用基类的成员函数 fun()
      cout<<"Derived::fun"<<endl;
    }
};
```

程序中划线处正确的语句是（ ）。

- A. fun(); B. Base.fun(); C. Base::fun(); D. Base->fun();

4. 派生类的保护成员比基类的保护成员多，但派生类并没有定义自有的保护成员，这说明所采用的继承方式为（ ）。

- A. 公有继承 B. 保护继承 C. 私有继承 D. 默认的继承

5. 基类中的某些成员在派生类中的访问属性与派生方式无关，这些成员必然是原基类中的（ ）。

- A. 公有成员 B. 私有成员 C. 保护成员 D. 继承的私有成员

6. 有如下程序：

```
#include <iostream>
using namespace std;
class Base {
private:
    void fun1() const { cout << "fun1"; }
protected:
    void fun2() const { cout << "fun2"; }
```

```

public:
    void fun3() const { cout << "fun3"; }
};
class Derived : protected Base {
public:
    void fun4() const { cout << "fun4"; }
};
int main() {
    Derived obj;
    obj.fun1();           // ①
    obj.fun2();           // ②
    obj.fun3();           // ③
    obj.fun4();           // ④
    return 0;
}

```

其中违反类成员访问控制权限的语句是 ()。

A. ①②③④

B. ①②③

C. ②③④

D. ①④

7. 关于类的继承, 下列表述中正确的是 ()。

A. 通过继承, 基类的所有成员都成为派生类的成员

B. 通过继承, 只有基类的公有成员和保护成员成为派生类的成员

C. 通过公有继承, 基类的所有成员都成为派生类的成员

D. 通过公有继承, 只有基类的公有成员和保护成员成为派生类的成员

8. 2005 年下半年全国计算机等级考试二级笔试试卷 C++ 语言程序设计。

有如下的类声明:

```

class MyBASE{
    int k;
public:
    void set(int n){ k=n;}
    int get()const{ return k; }
};
class MyDERIVED: protected MyBASE{
protected:
    int j;
public:
    void set(int m, int n){ MyBASE::set(m); j=m;}
    int get()const{ return MyBASE::get()+j; }
};

```

则类 MyDERIVED 中保护成员的个数是 ()。

A. 1

B. 2

C. 3

D. 4

9. 下面是两个类的声明:

```

class AA
{
    int x;
public:
    AA(int n):x(n){}
};
class BB
{
    static int a;
    const int b;
};

```

```

    AA c;
public:
    BB(int, int, int);
};

```

在定义构造函数 `BB::BB(int, int, int)` 时, 必须通过初始化列表中加以初始化的数据成员包括 ()。

- A. a 和 b B. a 和 c C. b 和 c D. a、b 和 c

10. 多继承是指 ()。

- A. 一个派生类同时继承多个基类
 B. 多个派生类同时继承一个基类
 C. 基类本身又是一个更高级基类的派生类
 D. 派生类本身又是更低一级派生类的基类

11. 有如下类声明:

```

class AA{
    void f1();
protected:
    void f2();
public
    void f3();
};
class BB: public AA{
public:
    void f4();
};

```

在 `f1`、`f2`、`f3` 这 3 个函数中, `f4` 可以调用的包括 ()。

- A. `f1` 和 `f2` B. `f2` 和 `f3` C. `f1` 和 `f3` D. `f1`、`f2` 和 `f3`

12. 有以下程序:

```

#include <iostream>
using namespace std;
class A {
public:
    A() { cout << 'A'; }
    ~A() { cout << 'C'; }
};
class B : public A {
public:
    B() { cout << 'G'; }
    ~B() { cout << 'T'; }
};
int main()
{
    B obj;
    return 0;
}

```

执行后的输出结果是 ()。

- A. GATC B. AGTC C. GACT D. AGCT

二、填空题

1. 在用 class 定义派生类时，默认的继承方式是_____。
2. 如果类 Alpha 继承了类 Beta，则类 Alpha 称为_____类，类 Beta 称为_____类。
3. 对基类数据成员的初始化必须在派生类构造函数中的_____处执行。
4. 有如下程序：

```
#include <iostream>
using namespace std;
class Base {
public:
    virtual void Show() { cout << "BB"; }
};
class Derived : public Base {
public:
    void Show() { cout << "DD"; }
};
void Fun(Base &obj) { obj.Show(); }
int main()
{
    Base objB;
    Derived objD;
    Fun(objB);
    Fun(objD);
    return 0;
}
```

这个程序的输出结果是_____。

5. 有如下程序：

```
#include<iostream>
using namespace std;

class Base{
    int m;
public:
    Base(int k):m(k){}
    int getValue()const{ return m;}
};

class Derived: public Base{
    int n;
public:
    Derived(int k):Base(k-1),n(k){}
    int getValue()const{ return _____; }
};

int main(){
    Derived d1(3), d2=4;
    cout<<d1.getValue()<<','<<d2.getValue();
    return 0;
}
```

程序应输出：6,12，程序有缺失的片段，请补充完整。