

C++面向对象程序设计模拟试题九

一、单项选择题（本大题共 10 小题，每小题 2 分，共 20 分）在每小题列出的四个备选项中，只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无分。

1. 说明虚函数的关键字是 ()。
A) inline B) virtual C) define D) static
2. C++中的模板包括 ()。
A) 对象模板和函数模板 B) 对象模板和类模板
C) 函数模板和类模板 D) 变量模板和对象模板
3. 有以下类声明：
class MyClass{ int num; };
则 MyClass 类的成员 num 是 ()。
A) 公有数据成员 B) 公有成员函数
C) 私有数据成员 D) 私有成员函数
4. 在 C++中，用于实现静态多态性的是()。
A) 友元函数 B) 重载函数
C) 静态成员函数 D) 虚函数
5. 派生类的成员函数不能访问基类的()。
A) 公有成员和保护成员 B) 公有成员
C) 私有成员 D) 保护成员
6. 在语句“cin >> data;”中，cin 是()。
A) C++的关键字 B) 类名 C) 对象名 D) 函数名
7. 下列关于输入流类成员函数 getline()的描述中，错误的是()。
A) 该函数可以用来读取键盘输入的字符串
B) 该函数读取的字符串长度不受任何限制
C) 该函数读取字符串时，遇到换行符便停止
D) 该函数读取字符串时，可以包含空格
8. 下列关于 C++函数的说明中，正确的是()。
A) 内联函数就是定义在另一个函数体内部的函数
B) 函数体的最后一条语句必须是 return 语句
C) 标准 C++要求在调用一个函数之前，如果没定义函数，则必须先声明其原型
D) 编译器会根据函数的返回值类型和参数表来区分函数的不同重载形式
9. 为取代 C 中带参数的宏，在 C++中使用了()。
A) 重载函数 B) 内联函数 C) 递归函数 D) 友元函数
10. 声明运算符重载的关键字的是()。
A) void B) int C) operator D) double

二、填空题（本大题共 5 小题，每小题 2 分，共 10 分）不写解答过程，将正确的答案写在每小空的空格内。错填或不填均无分。

1. 一个函数名为 GetArea，返回值类型为 double，没有参数的纯虚常成员函数可以声明为_____。
2. 在基本类型与类类型的转换中，类类型转换函数的作用是_____。
3. 派生类中的成员函数可以直接访问基类中的_____成员。
4. 重载函数运算符的函数名为_____。
5. 在 C++ 流类库中，根基类为_____。

三、程序分析题（本大题共 6 小题，每小题 5 分，共 30 分）给出下面各程序的输出结果。

1. 阅读下面程序，写出输出结果。

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

class Int
{
private:
    int n;

public:
    Int(int i = 9): n(i) { }
    int Get() { return n; }
    int Get() const { return n - 1; }
    operator int() const { return n; }
};

int main()
{
    Int a;
    const Int b = 8;
    cout << a.Get() << "," << b.Get() << "," << a << "," << b << "," << "end" << endl;

    return 0;
}
```

上面程序的输出结果为：

2. 阅读下面程序，写出输出结果。

```
#include <iostream>
```

```

using namespace std;

class A
{
public:
    A() { cout << "A()" << endl; }
    ~A() { cout << "~A()" << endl; }
};

int main()
{
    A *p = new A, a;
    delete p;
    cout << "end" << endl;

    return 0;
}

```

上面程序的输出结果为：

3. 阅读下面程序，写出输出结果。

```

#include <iostream>
using namespace std;

template <class Type>
Type f(const Type &x) {    return x + 1; }

int f(int x) { return x + 2.1; }

int f(double x) { return x + 3; }

int main()
{
    cout << f(96) << endl;
    cout << f(96.0) << endl;
    cout << f(float(96.0)) << endl;
    cout << f(char(96.0)) << endl;
    cout << f('a') << endl;

    return 0;
}

```

上面程序的输出结果为：

4. 阅读下面程序，写出输出结果。

```

#include <iostream>
using namespace std;

int x = 1;

namespace ns
{
    int x = 6;
    int f(int x) { return ::x + 1; }
    int f(double x) { return x + 1; }
}

int main()
{
    int x = 8;
    cout << x << " " << ::x << " " << ns::x << " " << ns::f(1) << " " << ns::f(1.0) << endl;

    return 0;
}

```

上面程序的输出结果为：

5. 阅读下面程序，写出输出结果。

```

#include <iostream>
using namespace std;

class A
{
public:
    virtual void f() const { cout << "A::f()" << endl; }
};

class B: public A
{
public:
    void f() const { cout << "B::f()" << endl; }
};

void Refer(const A &a) { a.f(); }

int main()
{
    A a, *p;
    B b;
}

```

```

        Refer(a);
        Refer(b);
        p = &a;
        p->f();
        p = &b;
        p->f();
        p->A::f();

        return 0;
}

```

上面程序的输出结果为:

6. 阅读下面程序, 写出输出结果。

```

#include <iostream>
using namespace std;

template <class Type, int size>
class Array
{
private:
    Type elem[size];

public:
    Array(Type a[]) { for (int i = 0; i < size; i++) elem[i] = a[i]; }
    int Length() const { return size; }
    Type &operator[](int i){ return elem[i - 1]; }
};

int main()
{
    int a[] = {1, 2, 3, 4, 5,};
    const int n = 5;
    Array<int, n> obj(a);

    for (int i = 1; i <= obj.Length(); i++) cout << obj[i] << "  ";
    cout << endl;

    return 0;
}

```

上面程序的输出结果为:

四、程序填空题（本大题共 4 小题，每小题 3 分，共 12 分）

1. 将如下程序补充完整。

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

class Double
{
private:
    double num;

public:
    Double(double num) { _____ = num; } //初始化数据成员 num 为形参 num
    void Show() const { cout << num << endl; }
};

int main()
{

    Double x(1.8);
    x.Show();

    return 0;
}
```

2. 将如下程序补充完整。

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

class Int
{
private:
    int n;

public:
    Int(int m): n(m) { }
    Int _____ { ++n; return *this; } // 重载前缀++运算符
    void Show() const { cout << n << endl; }
};

int main()
```

```

{

    Int i(8);
    (i++).Show();
    i.Show();

    return 0;
}

```

3. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

class A
{
protected:
    int m;

public:
    A(int a): _____{ }           // 用 a 初始化 m
    void Show() const { cout << m << endl; }
};

int main()
{
    A obj(1);
    obj.Show();

    return 0;
}

```

4. 将如下程序补充完整。

```

#include <iostream>
using namespace std;

class Point
{
private:
    int x, y;
    static int count;

public:

```

```

    Point(int m = 0, int n = 0): x(m), y(n) { count++; }
    ~Point() { count--; }
    int GetX() const { return x; }
    int GetY() const { return y; }
    static void ShowCount() { cout << count << endl; }
};

_____; // 静态数据成员的初始化

int main()
{
    Point::ShowCount(); // 输出 0
    Point a(6, 8);
    Point::ShowCount(); // 输出 1

    return 0;
}

```

五、编程题（本大题共 2 小题，第 1 小题 12 分，第 2 小题 16 分，共 28 分）

1. 试使用函数模板实现求一个数组各元素的最大值，要求编写测试程序。
函数模板声明如下：

2. 编写程序实现如下功能：

(1) 从键盘上输入一系列学生数据(学号、姓名、成绩)，并将这些学生数据写入到文件 stud.dat 中。

(2) 显示文件 stu.dat 中的学生数据和相应的成绩等级(≥ 90 为优，80~89 为良，70~79 中，60~69 为及格， ≤ 59 为不及格)。

C++面向对象程序设计模拟试题九参考答案

一、单项选择题（本大题共 10 小题，每小题 2 分，共 20 分）在每小题列出的四个备选项中，只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无分。

- | | | | | |
|------|------|------|------|-------|
| 1. B | 2. C | 3. C | 4. B | 5. C |
| 6. C | 7. B | 8. C | 9. B | 10. C |

二、填空题（本大题共 5 小题，每小题 2 分，共 10 分）不写解答过程，将正确的答案写在每小题的空格内。错填或不填均无分。

1. 参考答案: `virtual double GetArea() const = 0;`
2. 参考答案: 将类类型转换为基本类型
3. 参考答案: 公有与保护
4. 参考答案: `operator()`
5. 参考答案: `ios`

三、程序分析题（本大题共 6 小题，每小题 5 分，共 30 分）给出下面各程序的输出结果。

1. 参考答案:
9,7,9,8,end

2. 参考答案:
A()
A()
~A()
end
~A()

3. 参考答案:
98
99
97
a
b

4. 参考答案:
8 1 6 2 2

5. 参考答案:

A::f()

B::f()

A::f()

B::f()

A::f()

6. 参考答案:

1 2 3 4 5

四、程序填空题 (本大题共 4 小题, 每小题 3 分, 共 12 分)

1. 参考答案: this->num 或 Int:: num

2. 参考答案: operator++()

3. 参考答案: m(a)

4. 参考答案: int Point::count = 0

五、编程题 (本大题共 2 小题, 第 1 小题 12 分, 第 2 小题 16 分, 共 28 分)

1. 参考程序:

```
#include <iostream>                // 编译预处理命令
#include <cmath>                    // 编译预处理命令
using namespace std;              // 使用命名空间 std

template <class Type>
Type Max(Type a[], int n)          // 求数组 a 各元素的最大值
{
    Type m = a[0];                // 定义变量 m, 用于表示当前所求的最大值
    for (int i = 1; i < n; i++) m = m < a[i] ? a[i]: m;    // 依次比较求最大值
    return m;                     // 返回最大值
}

int main()                        // 主函数 main()
{
    double a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};    // 定义数组
    cout << Max(a, 9) << endl;    // 输出各元素的最大值

    return 0;                    // 返回值, 返回操作系统
```

```
}
```

2. 参考程序:

```
#include <iostream>           // 预处理命令
#include <fstream>             // 预处理命令
#include <iomanip>              // 流格式控制预处理命令
using namespace std;          // 使用标准命名空间 std

// 学生类
class Student
{
private:
// 数据成员:
    int no;                    // 学号
    char name[16];             // 姓名
    int score;                 // 成绩

public:
// 公有函数:
    void Input()                // 输入信息
    { cin >> no >> name >> score; }
    void Show() const           // 显示信息
    {
        char level[6];          // 等级
        if (score >= 90) strcpy(level, "优");    // >=90 为优
        else if (score >= 80) strcpy(level, "良"); // 80~89 为良
        else if (score >= 70) strcpy(level, "中"); // 70~79 为中
        else if (score >= 60) strcpy(level, "及格"); // 60~69 为及格
        else strcpy(level, "不及格");           // <60 为不及格
        cout << setw(16) << no << setw(18) << name << setw(16) << score << setw(16)
        << level << endl;
    }
};

// 学生管理类
class StudentManage
{
private:
// 辅助函数:
    static void Write();        // 写入学生信息
    static void Show();         // 显示学生信息

public:
// 公有函数:
```

```

        static void Run();                // 运行学生管理
    };

// 学生管理类的实现部分
void StudentManage::Write()              // 写入学生信息
{
    Student objStu;                      // 学生对象
    ofstream outFile("stud.dat", ios::binary | ios::app);    // 定义文件对象
    if (!outFile)
    {
        // 打开文件失败
        cout << "打开文件失败!" << endl;
        exit(1);
    }

    char flag = 'Y';                     // 判断是否继续录入的标志, 初始化为'Y'
    cout << "输入学生信息数据(学号 姓名 成绩):" << endl;
    while (toupper(flag) == 'Y')
    {
        objStu.Input();                  // 输入学生信息
        outFile.write((char *)&objStu, sizeof(objStu));    // 写入文件
        cout << "是否继续录入信息?(Y/N)";
        cin >> flag;
    }

    outFile.close();                     // 关闭文件
}

void StudentManage::Show()               // 显示学生信息
{
    Student objStu;                      // 学生对象
    ifstream inFile("stud.dat", ios::binary); // 定义文件对象
    if (!inFile)
    {
        // 打开文件失败
        cout << "打开文件失败!" << endl;
        exit(2);
    }

    cout << "显示学生信息" << endl;
    cout << setw(16) << "学号" << setw(18) << "姓名" << setw(16) << "成绩" <<
    setw(16) << "等级" << endl;
    inFile.read((char *)&objStu, sizeof(objStu));    // 从文件读数据
    while (!inFile.eof())
    {
        // 循环显示学生信息
        objStu.Show();                  // 显示学生信息
    }
}

```

```

        inFile.read((char *)&objStu, sizeof(objStu)); // 从文件读数据
    }

    inFile.close(); // 关闭文件
}

void StudentManage::Run() // 运行学生管理
{
    int select; // 临时变量

    do
    {
        cout << "选择(1.写入学生信息 2.显示学生信息 3.退出):";
        cin >> select; // 输入选择
        switch (select)
        {
            case 1:
                Write(); // 写入学生信息
                break;
            case 2:
                Show(); // 显示学生信息
                break;
            case 3:
                break; // 退出
            default:
                // 其它情况, 表示选择有误
                cout << "选择有误!" << endl;
                break;
        }
    } while (select != 3);
}

int main(void) // 主函数 main(void)
{
    StudentManage::Run(); // 运行学生管理

    system("PAUSE"); // 调用库函数 system( )
    return 0; // 返回值 0, 返回操作系统
}

```