C++面向对象程序设计模拟试题五

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. 下列对类的构造函数和析构函数描述正确的是（　　）。
   A）构造函数可以重载，析构函数不能重载
   B）构造函数不能重载，析构函数可以重载
   C）构造函数可以重载，析构函数可以重载
   D）构造函数不能重载，析构函数不能重载
2. 在函数定义前加上关键字“inline”，表示该函数被定义为（　　）。
   A）重载函数　　B）内联函数
   C）成员函数　　D）普通函数
3. 下面有关重载函数的说明中，（　　）是正确的。
   A）重载函数必须具有不同的返回值类型
   B）重载函数形参个数必须不同
   C）重载函数一般具有不同的形参列表
   D）重载函数名可以不同
4. 下列有关类与对象的说法中，(　　) 是不正确的。
   A）对象是类的一个实列
   B）任何一个对象只能属于一个具体的类
   C）一个类只能有一个对象
   D）类和对象的关系和数据类型与变量的关系类似
5. 已知: Print( )函数是一个类的常成员函数，它无返回值，下列表示中，正确的是(　　)。
   A）void Print( ) const;　　B）const void Print( );
   C）void const Print( );　　D）void Print(const)
6. 假定 Myclass 为一个类,那么下列的函数说明中(　　)为该类的析构函数。
   A）void ~Myclass( );　　B）~Myclass(int n);
   C）Myclass( );　　D）~Myclass( )
7. 下面类的定义中有(　　) 处错误。

```
class myclass
{
    int i;
public:
    void myclass( );
    ~myclass(value);
};
```

   A）1　　B）2　　C）3　　D）4
8. 说明虚函数的关键字是（　　）。
   A）inline　　B）virtual　　C）define　　D）static
9. cout 是某个类的标准对象的引用，该类是（　　）。
   A）ostream　　B）istream　　C）stdout　　D）stdin

10．如果 class 类中的所有成员在定义时都没有使用关键字 public、private 或 protected，则所有成员缺省定义为（　　）。

A）public　　B）protected　　C）private　　D）static

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

1．重载运算符“-” 的函数名为(　　)。

2．C++中类的用途有两种，一种是类的实例化，即生成类的对象，另一种是通过(　　)，派生出新的类。

3．派生类中的成员不能直接访问基类中的（　　）成员。

4．编译时多态性可以用（　　）函数实现。

5．使用 new 建立的动态对象在不用时应该用（　　）删除，以便释放所占用空间。

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．若有以下程序:

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

class A
{
    int a;

public:
    A(int aa = 0 ){ a = aa; }
    ~A( ) {cout << "Destructor A! " << a << endl; }
};

class B: public A
{
    int b;

public:
    B(int aa = 0, int bb = 0): A(aa) { b = bb; }
    ~B() { cout << "Destructor B! " << b << endl; }
};

int main()
{
    B x(5), y(6,7);

    return 0;
}
```

上面程序的输出结果为:

2．若有以下程序:

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

class Point
{
    int x, y;
public:
    Point() { x = 0; y = 0; }
    void SetPoint(int x1, int y1) { x = x1; y = y1; }
    void DisPoint() { cout << "x=" << x <<"," << "y=" << y << endl; }
};

int main()
{
    Point *p = new Point;
    p->SetPoint(5, 12);
    p->DisPoint();
    delete p;

    return 0;
}
```

上面程序的输出结果为：

3．若有以下程序:

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

class Sample
{
    int n;

public:
    Sample (int i) { n =i; }
    void Add() { s += n; }
    static int s;
    void    Dis() { cout << s << endl; }
};

int Sample::s = 0;

int main()
```

```
{
    Sample a(2), b(5), c(8);
    a.Add( );
    b.Add( );
    c.Dis( );

    return 0;
}
```
上面程序的输出结果为：

4. 若有以下程序:
```
#include <iostream>
using namespace std;

class Base
{
public:
    void Fun() { cout << "1" << endl;}
};

class Derived:public Base
{
public:
    void Fun() { cout << "2" << endl; }
};

int main()
{
    Derived a;
    Base *p;
    p=&a;
    p->Fun();
    a.Fun();

    return 0;
}
```
上面程序的输出结果为：

5. 若有以下程序:
```
#include <iostream>
using namespace std;

template<class T1, class T2>
void Fun(T1 &x, T2 &y)
```

```
{
    if (sizeof(T1) > sizeof(T2) ) x = (T1)y;
    else y = (T2)x;
}

int main()
{
    double d;
    int i ;
    d = 99.99;
    i = 88;
    Fun(d,i);
    cout << "d=" << d << "i=" << i <<endl;

    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;

class Point
{
public:
    Point (int a = 0, int b = 0): x(a), y(b) { }
    int GetX() const { return x; }
    int GetY() const { return y; }
    void SetX(int a) { x = a; }
    void SetY(int a) { y = a; }

private:
    int x;
    int y;
};

int main()
{
    Point p1;
    const Point p2(3, 4);
    cout << p1.GetX() << endl;
    p1.SetX(1);
    cout << p1.GetX() << endl;
    p1.SetY(2);
```

```
    cout << p1.GetY() << endl;
    cout << p2.GetX() << endl;
    cout << p2.GetY() << endl;

    return 0;
}
```

上面程序的输出结果为：

四、完成程序填题（本大题共 4 个小题，每小题 3 分，共 12 分）下面程序都留有空白，请将程序补充完整。

1．将如下程序补充完整。

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

class Test
{
private:
    int n;

public:
    Test(int n) { ___[1]___; }              // 实始化数据成员 n 为形能 n
    void Show() const { cout << n << endl; }
};

int main()
{

    Test i(108);
    i.Show();

    return 0;
}
```

参考答案：this->n 或 Test::n

2．将如下程序补充完整。

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

class A
{
private:
    int a;
```

```
public:
    A(int x) { a = x; }
    void Show() const
    { cout << "a:" << a << endl; }
};

class B: public A
{
protected:
    int b;

public:
    B(int x, int y):____[2]____{ b = y; }          // 初始化 a 为 x，y 为 b
    void Show() const
    {
        A::Show();
        cout << "b:" << b << endl;
    }
};

int main()
{
    B obj(5, 18);
    obj.Show();

    return 0;
}
```

3．将如下程序补充完整。

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

class Test
{
private:
    static int num;

public:
    Test() { num++; }
    ~Test() { num--; }
    static void ShowObjectNum() { cout << num << endl; }
};
```

_____[3]_____ // 静态数据成员 num 初始化为 0

```
int main()
{
    Test::ShowObjectNum();
    Test obj;
    Test::ShowObjectNum();

    return 0;
}
```

4．将如下程序补充完整。

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

class Complex
{
private:
    double realPart;
    double imagePart;

public:
    Complex(double real = 0, double image = 0): realPart(real), imagePart(image){ }
    Complex _____[4]_____(const Complex &a) const        // 重载加法运算符+
    {   return Complex(realPart + a.realPart, imagePart + a.imagePart); }
    void Show() const { cout << realPart << "+" << imagePart << "i" << endl;}
};

int main()
{
    Complex a(1, 2), b(2, 6), c;

    c = a + b;
    cout << "a="; a.Show();
    cout << "b="; b.Show();
    cout << "c="; c.Show();

    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．设计一个类 DateInfo，要求其满足下述要求：

（1）要求有一个无参的构造函数，其初始的年、月、日分别为：2010，6，8。

（2）要求有一个带参数的构造函数，其参数分别对应年、月、日。

（3）要求用一个成员函数实现日期的设置。

（4）要求用一个成员函数实现输出日期。

2．定义 Staff（员工）类，由 Staff 分别派生出 Saleman（销售员）类和 Manager（经理）类，要求：

（1）在 Staff 类中包含的数据成员有编号(num)、姓名(name)、出勤率(rateOfAttend)、基本工资(basicSal)和奖金(prize)。在 Saleman 类中还包含数据成员销售员提成比例(deductRate)和个人销售额(personAmount)，在 Manager 类中还包含数据成员经理提成比例(totalDeductRate)和总销售额(totalAmount)

（2）各类人员的实发工资公式如下：

员工实发工资 = 基本工资 + 奖金 * 出勤率

销售员实发工资 = 基本工资 + 奖金 * 出勤率 + 个人销售额 * 销售员提成比例

经理实发工资 = 基本工资 + 奖金 * 出勤率 + 总销售额 * 经理提成比例

（3）每个类都有构造函数、输出基本信息函数(Output)和输出实发工资函数(OutputWage)。

C++面向对象程序设计模拟试题五参考答案

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．A)	2．B)	3．C)	4．C)	5．A)
6．D)	7．B)	8．B)	9．A)	10．C)

二、填空题（本大题共 5 小题，每小题 2 分，共 10 分）不写解题过程，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：oprator-
2．参考答案：继承
3．参考答案：私有
4．参考答案：重载
5．参考答案：delete

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：
Destructor B! 7
Destructor A! 6
Destructor B! 0
Destructor A! 5
2．参考答案：x=5, y=12
3．参考答案：7
4．参考答案：
1
2
5．参考答案：d=88i=88
6．参考答案：
0
1
2
3
4

四、完成程序填题（本大题共 4 个小题，每小题 3 分，共 12 分）下面程序都留有空白，请将程序补充完整。

1．参考答案：[1] this->n = n 或 Test::n = n
2．参考答案：[2] A(x)
3．参考答案：[3] int Test::num = 0;
4．参考答案：[4] operator+

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>
using    namespace    std;

class DateInfo
{
private:
    int year, month, day;

public:
    DateInfo(): year(2010), month(6), day(8){ }
    DateInfo(int y, int m, int d): year(y), month(m), day(d){ }
    void Set(int y, int m, int d)
    {
        year = y;
        month = m;
        day = d;
    }
    void Show() { cout << year << "年" << month << "月" << day << "日" << endl; }
};

int main()
{
    DateInfo d1, d2(1988, 8, 18);
    d1.Show();
    d2.Show();
    d2.Set(1999, 9, 19);
    d2.Show();
    return 0;
}
```

2．参考程序：

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

class Staff
{
protected:
    int num;
    char name[18];
    double rateOfAttend;
    double basicSal ;
    double prize ;
    static int count;
```

```
public:
    Staff(){ }
    void Input()
    {
        num = ++count;
        cout << "请输入编号为" << num <<"号员工的信息" << endl;
        cout << "姓名:";
        cin >> name;
        cout << "基本工资:";
        cin >> basicSal;
        cout << "奖金:";
        cin >> prize;
        cout <<"出勤率(0~1):";
        cin >> rateOfAttend;
    }
    void Output() const
    {
        cout << "编号:" << num << endl;
        cout << "姓名:" << name << endl;
        cout << "基本工资:" << basicSal << "元" << endl;
        cout << "奖金:" << prize << "元" << endl;
        cout << "出勤率:" << rateOfAttend * 100 << "%" << endl;
    }
    void OutputWage() const
    {
        cout << "实发工资:"
            << basicSal + prize * rateOfAttend      << "元" << endl;
    }
};

int Staff::count = 1000;

class Saleman : public Staff
{
protected:
    float deductRate;
    float personAmount;

public:
    Saleman (){ };
    void Input()
    {
        Staff::Input();
        cout << "个人销售额:";
```

```
        cin >> personAmount;
        cout << "提成比例:";
        cin >> deductRate;
    }
    void Output() const
    {
        Staff::Output();
        cout << "个人销售额:" << personAmount << "元" << endl;
        cout << "提成比例:" << deductRate * 100 << "%" << endl;
    }
    void OutputWage() const
    {
        cout << "实发工资:"
             << basicSal + prize * rateOfAttend
             + personAmount * deductRate
             << "元" << endl;
    }
};

class Manager: public Staff
{
protected:
    double totalDeductRate;
    double totalAmount;

public:
    Manager(){ }
    void Input()
    {
        Staff::Input();
        cout << "公司总销售额:";
        cin >> totalAmount;
        cout << "提成比例:";
        cin >> totalDeductRate;
    }
    void Output() const
    {
        Staff::Output();
        cout << "公司总销售额:" << totalAmount << "元" << endl;
        cout << "提成比例:" << totalDeductRate * 100 << "%" << endl;
    }
    void OutputWage() const
    {
        cout << "实发工资:"
```

```
                << basicSal + prize * rateOfAttend
                + totalAmount * totalDeductRate
                << "元" << endl;
    }
};

int main()
{
    char flag = 'Y';

    while (toupper(flag) == 'Y')
    {
        cout << "请选择录入类别(1.员工  2.销售员  3.经理)";
        int n;
        cin >> n;

        if (n == 1)
        {    // 员工
            Staff objStaff;
            objStaff.Input();
            objStaff.Output();
            objStaff.OutputWage();
        }
        else if (n == 2)
        {
            Saleman objSaleman;
            objSaleman.Input();
            objSaleman.Output();
            objSaleman.OutputWage();
        }
        else if (n == 3)
        {
            Manager objManager;
            objManager.Input();
            objManager.Output();
            objManager.OutputWage();
        }
        else
        {
            cout << "选择有误!"<< endl;
        }

        cout << endl << "是否继续录入信息?(Y/N)";
        cin >> flag;
```

```
        }

        return 0;
}
```