

西北农林科技大学本科课程考试试卷 (A 卷)

2011-2012 学年第 1 学期《C++ 面向程序设计》课程闭卷

命题教师： 张 XX 审题教师： 李 XX 考试成绩： _____

注意事项：

1. 本试卷共 36 道试题，满分 100 分，考试时间 120 分钟。
2. 学生在答题前请先填写专业、学号、学院、姓名等基本信息。

题号	一	二	三	四	五	总分	审核人
得分							

得 分	
评阅人	

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_____。(B)
A. inline B. virtual C. define D. static
2. 在每个 C++ 程序中都必须包含有这样一个函数，该函数的函数名为 (A)
A. main B. MAIN C. name D. function
3. cout 是某个类的标准对象的引用，该类是_____。(A)
A. ostream B. istream C. stdout D. stdin
4. 如果在类外有函数调用 CPoint::func();则函数 func() 是类 CPoint 的 (C)
A. 私有静态成员函数 B. 公有非静态成员函数
C. 公有静态成员函数 D. 友元函数
5. 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_____。(C)
A. public B. protected C. private D. static
6. 一个类的所有对象共享的是_____。(D)
A. 私有数据成员 B. 公有数据成员
C. 保护数据成员 D. 静态数据成员

2. 函数模板紧随 template 之后尖括号内的类型参数要冠以保留字 class 。
3. 编译时多态性可以用 重载 函数实现。
4. 拷贝构造函数用它所在类的 对象 作为参数。
5. 用关键字 static 修饰的类的成员称为 静态 成员。
6. 重载运算符“+”的函数名为 operator+ 。
7. 设函数 max 是由函数模板实现的, 并且 max(3.5, 5) 和 max(3, 5) 都是正确的函数调用, 则此函数模板具有 2 个类型参数。
8. 在 C++ 中, 函数重载与虚函数帮助实现了类的 多态 性。
9. 由 static 修饰的数据成员为该类的所有对象 共享 。
10. 重载函数在参数类型或参数个数上不同, 但 函数名 相同。

得 分	
评阅人	

四. 程序分析题(本大题共 4 小题, 每小题 5 分, 共 20 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 CArray
{
public:
    CArray(int iArray[], int iSize):m_pArray(iArray), m_iSize(iSize)
    {
    }
    int GetSize()
    {
        return m_iSize;
    }
    int &operator[](int iIndex)
    {
        return m_pArray[iIndex - 1];
    }
private:
    int *m_pArray; // 指向一个数组空间
    int m_iSize; // 数组元素个数
};

int main()
```

```
{
    int s[]={3, 7, 2, 1, 5};
    CArray oArray(s, 5);
    oArray[1] = 9;
    for (int i = 1; i <= 5; i++)
    {
        cout << oArray[i] << " ";
    }
    cout << endl;
    return 0;
}
```

上面程序的输出结果为:

参考答案:

9 7 2 1 5

2. 阅读下面程序, 写出输出结果。

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

template <class Type>
void Print(Type a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << a[i] << " ";
    }
}

int main()
{
    int a[] = { 5, 6, 8};
    double b[] = {6.8, 9.6};

    Print(a, sizeof(a) / sizeof(int));
    Print(b, 2);
    cout << endl;

    return 0;
}
```

```
}
```

上面程序的输出结果为:

参考答案:

5 6 8 6.8 9.6

3. 阅读下面程序, 写出输出结果。

```
#include <iostream>
using namespace std;
class CTest
{
public:
    CTest(int iVar):m_iVar(iVar)
    {
        m_iCount++;
    }
    ~CTest()
    {
    }
    void Print() const;
    static int GetCount()
    {
        return m_iCount;
    }
private:
    int m_iVar;
    static int m_iCount;
};
int CTest::m_iCount = 0;
void CTest::Print() const
{
    cout << this->m_iVar << " " << this->m_iCount << " ";
}
int main()
{
    CTest oTest1(6);
    oTest1.Print();
    CTest oTest2(8);
```

```
oTest2.Print();
cout << CTest::GetCount();
cout << endl;
return 0;
}
```

上面程序的输出结果为：

<p>参考答案：</p> <p>6 1 8 2 2</p>

4. 阅读下面程序，写出输出结果。

```
#include <iostream>
using namespace std;
class CTest
{
public:
    CTest(int iX = 0, int iY = 0, int iZ = 0):m_iZ(iZ)
    {
        m_iX = iX;
        m_iY = iY;
    }
    void Print()
    {
        cout << m_iX << endl;
        cout << m_iY << endl;
    }
    void Print() const
    {
        cout << m_iZ << endl;
    }
private:
    int m_iX, m_iY;
    const int m_iZ;
};
int main()
{
    CTest oTest1;
    oTest1.Print();
```

```
CTest oTest2(1, 6, 8);
oTest2.Print();
const CTest oTest3(6, 0, 18);
oTest3.Print();
cout << endl;
return 0;
}
```

上面程序的输出结果为:

参考答案:

0 0 1 6 18

得 分	
评阅人	

五. 编程题(本大题共 2 个小题, 每小题 10 分, 共 20 分)

1. 编写一个函数模板, 用于求参数的绝对值, 并编写测试程序进行测试。函数模板声明如下:

```
template <class Type>
Type Abs(Type tVar)
```

参考答案:

```
#include <iostream>
using namespace std;
template <class Type>
Type Abs(Type tVar)
{
    if (tVar >= 0) return tVar;
    else return -tVar;
}
int main()
{cout << Abs(5) << endl;
cout << Abs(-5) << endl;
cout << Abs(2.5) << endl;
cout << Abs(-2.5) << endl;
```

```
return 0;}
```

2. 定义一个复数类 CComplex, 定义带有 2 个参数(其中一个为缺省参数)的构造函数, 显示复数值的函数 Show(), 重载“+”运算符(用成员函数实现), 并编写测试程序进行测试。

参考答案:

```
#include <iostream>
using namespace std;
class CComplex
{
public:
    CComplex(double dReal, double dImage = 0)
    {
        m_dReal = dReal;
        m_dImage = dImage;
    }
    void Show()
    {
        cout << m_dReal;
        if (m_dImage > 0)
            {cout << "+" << m_dImage << "i" << endl;}
        else if(m_dImage < 0)
            {cout << "-" << -m_dImage << "i" << endl;}
    }
    CComplex operator+(const CComplex &obj)
    {
        CComplex objTemp(m_dReal + obj.m_dReal, m_dImage + obj.m_dImage);
        return objTemp;
    }
private:
    double m_dReal, m_dImage;
};

int main()
{
    CComplex obj1(2, 6), obj2(3, 8), obj3(0);
    obj1.Show();
    obj2.Show();
}
```



```
obj3.Show();  
obj3 = obj1 + obj2;  
obj3.Show();  
return 0;  
}
```