重庆大学 面向对象程序设计与C++ 课程试卷

A卷　B卷

2008～2009 学年　第 1 学期

开课学院：计算机　课程号：18008035　考试日期：2009/01

考试方式：开卷　闭卷　其他　　考试时间：120 分钟

题 号	一	二	三	四	五	六	七	八	九	十	总分
得 分											

（注：本卷来源于网络，是重大 C++期末试卷，非考研复试的）

一、简答（25 分,5 分/小题）

1. 解释引用和指针的异同。
2. 什么叫晚绑定，如何实现的？
3. 在何种情况下，copy constructor 会被调用？默认的拷贝构造函数如何执行对象拷贝？
4. 解释 public，private，protected 的意义和用途。
5. 什么叫名字装饰，名字装饰在 C++中起什么作用？

二、程序分析（35 分）

1. 阅读下述程序，写出执行结果（6 分）

```
#include <iostream>
using namespace std;
int b=6;
int func(int &x){
   cout<<b<<endl;
   b=b+2;
   x=x+3;
   cout<<x<<endl;
   return b;
}
int main(){
   int a=2,b=2;
   b+=func(a);
   cout<<"b="<<b<<"\n";
   cout<<"a="<<a<<endl;
   cout<<::b;
   return 0;
}
```

2. 阅读下述程序，写出执行结果并对执行过程作出解释（8 分）

```
#include <fstream>
#include <string>
using namespace std;
ofstream out("HowMany.out");

class HowMany {
    static int objectCount;
public:
    HowMany() { objectCount++; }
    static void print(const string& msg = "") {
        if(msg.size() != 0) out << msg << ": ";
        out << "objectCount = "
        << objectCount << endl;
    }
    ~HowMany() {
        objectCount--;
        print("~HowMany()");
    }
};

int HowMany::objectCount = 0;
// Pass and return BY VALUE:
HowMany f(HowMany x) {
    x.print("x argument inside f()");
    return x;
```

```
    }

    int main() {
        HowMany h;
        HowMany::print("after construction of h");
        HowMany h2 = f(h);
        HowMany::print("after call to f()");
    }
```

3. 阅读程序，写出执行结果（8 分）

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

class Pet {
   int i;
public:
  virtual void eat() const {
        cout << "Pet::eat" << endl;
  }
  void speak() const{
        cout << "Pet::speak" << endl;
  }
  virtual void sleep() const{
        cout << "Pet::sleep" << endl;
  }
};

class Goldfish : public Pet {
public:
  void eat() const {
        cout << "Goldfish::eat" << endl;
  }
  virtual void speak() const{
        cout << "Goldfish::speak" << endl;
  }
};

int main(int argc, char* argv[])
{
        cout << “sizeof Pet=” << sizeof(Pet) << endl;
        Goldfish bob;
        cout<<”sizeof bob=”<<sizeof(bob)<<endl;
        bob.eat();
        bob.speak();
        bob.sleep();

        Pet* p = &bob;
        p->eat();
        p->speak();
        p->sleep();
        return 0;
}
```

4. 找出下述程序的错误，并说明原因（5 分）

```
    class X{
        int a;
    public:
      int func(void){
            return a++;
      }
    };

    class Y : public X{
      public:
        void set (int c){
            this->a = c;
        }
        int describe() const{
            return func();
          }
    };
```

5. 将模板类 Array 的定义补充完整，使得程序可以正确运行。（8 分）

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

template<class T>
class Array {
    enum { size = 100 };
    T A[size];
public:
};

void main()
{
    Array<string> as;
    as[0] = "0";
    for (int i=1; i < as.size(); i++) {
        as[i] += as[i-1];
    }
}
```

三、程序设计（40 分）

1. 创建一个名为Monitor的类，它可以记录其**incident( )** 成员函数被执行的次数。为Monitor类添加一个print()成员函数以显示incident()函数被执行的次数。（10分）

2. 创建一个简单的Shape继承结构。Shape被定义为基类，Circle,Square,Triangle定义为Shape的子类；为Circle,Square,Triangle定义合适的数据成员用于存储坐标、半径等信息；为三个子类定义合适的构造函数以初始化图形（构造函数参数中给出坐标，半径等信息）；为三个子类定义公开函数draw()用于描绘图形；定义一个全局函数drawShape(Shape& s),该函数以一个Shape的引用为参数，但能正确地描绘出不同子类型的图形。（20分）
说明：draw函数内部不用写出具体代码，以下述形式代码代替：

```
cout  << "Circle::draw()" << endl;
```

3. 创建一个Message类，其构造函数使用一个string型默认值为"Unnamed"的参数。在构造函数里，用参数初始化其私有string型数据成员sValue；为Monitor类创建两个重载的print函数，其中一个零参数，它简单输出sValue的值；另外一个接受一个string型参数，它先输出参数值作为标题，再输出sValue作为内容。（10分）

答题纸