maoli

厦门大学《C++程序设计》课程试卷

信息 学院 2019 级 软件工程类 专业

学年学期 19-20-2 主考教师：_____ （A）卷

☛请将答案按序写在学校统一印制的专用答题卷上，写在本卷或自备纸上者一律不得分。

一．单选题 (含 15 个小题，每小题 2 分，计 30 分)

1. 以下哪个选项不全是 C++中的关键字的（ A ）
   A. integer, float, double
   B. auto, inline, switch
   C. class, union, struct
   D. virtual, static, namespace

   不能用于声明变量

2. 下面叙述不正确的是（ D ）
   A. 派生类一般都用公有派生
   B. 对基类成员的访问必须是无二义性的
   C. 赋值兼容规则也适用于多重继承的组合
   D. 基类的公有成员在派生类中仍然是公有的 看继承方式

3. 下列关于 new 运算符的描述中，哪个是错误的（ D）
   A. 它可以用来动态创建对象和对象数组
   B. 使用 new 创建的 int 型数组 p[10]，可以用“delete []p;”来释放空间
   C. 使用它创建对象时要调用构造函数
   D. new 创建的动态变量的空间是在栈区中分配的 堆区

4. 下面代码编译时**不会报错**的语句是（ C ）

```
const int *p;
int *q;
const int x=0;
int y;
(1) p = &x;   *p = 1;   改 const
(2) q = &x;   类型不同
(3) p = &y;
(4) p = &x; q=p;   类型不同
```

   A. (1)　　B. (2)　　C. (3)　　D. (4)

5. 下列有关重载函数的说法中正确的是（ C ）

A. 重载函数必须具有不同的返回值类型 可以相同

B. 重载函数参数个数必须相同 可不同

C. 重载函数必须有不同的形参列表

D. 重载函数名可以不同 ⟶ 名字不一样就不是重载了

6. 对于下面的类 A、B、C 和 D

```
class A{
   int x;
public:
   A(int i){x=i;}
};
class B: virtual public A{
   int    y;
public:
   B(int    i): A(1) {y=i;}
};
class C: virtual public A{
   int z;
public:
   C(int i): A(2) {z=i;}
};
class D: public B, publicC {
   int m;
public:
   D(int i, int j, int k): C(j), B(i), A(3) {m=k;}
};
……
D d(1,2,3)
```

创建 D 类对象 d 时，所调用的构造函数及它们的执行顺序是（ D ）

A．D（）、B（）、C（）、A（）

B. D（）、C（）、B（）、A（）

C. A（）、C（）、B（）、D（）

D. A（）、B（）、C（）、D（）

基类 → 对象成员 → 派生类

↓

继承声明顺序

7. 如果 myclass 类定义了拷贝构造函数和一个整型参数的构造函数，还重载了赋值运算符。那么语句 myclass obj=10;，会（ B ）

A. 调用拷贝构造函数

B. 调用整型参数的构造函数

C. 调用赋值运算符（易错）

D. 引起编译错误

根据类型调用

example: myclass obj2=obj1;

8. 关于友元，下列说法错误的是（ A ）
A. 如果类 A 是类 B 的友元，类 B 是类 C 的友元，那么类 A 也是类 C 的友元
B. 如果函数 fun()被说明为类 A 的友元，那么在 fun()中可以访问类 A 的私有成员
C. 友元关系不能被继承
D. 友元是数据保护和数据访问效率之间的一种折衷方案

9. 关于 this 指针使用说法正确的是（ A ）
A. 保证每个对象拥有自己的数据成员，但共享处理这些数据的代码
B. 保证基类私有成员在子类中可以被访问
C. 保证基类保护成员在子类中可以被访问
D. 保证基类公有成员在子类中可以被访问

10. 假定一个类的构造函数为A(int aa,int bb) {a=aa--;b=a*bb;},则执行A x(4,5)；语句后 ，x.a和x.b的值分别为（ C ）
A. 3和15
B. 5和4
C. 4和20
D. 20 和 5

11. 有如下函数模板定义

```
template<class T>
T func( T x, T y)
{ return x*x+y*y; }
```

在下列 func 的调用中不正确的是（ A ）
A. func (3.5, 5) ;
B. func (3, 5);
C. func<double> (3.5, 5);
D. func< int > (3.5, 5);

12. 下列有关输入/输出（I/O）的说法中正确的是（ C ）
A. 在 C++中，输入/输出是语言定义的成分
B. 在 C++中，输入/输出操作不是一种基于字节流的操作
C. 对自定义的类重载插入操作符“<<”和抽取操作符“>>”时，不能作为类的成员函数来重载
D. 文件输入操作是指把计算机内存中的数据写入到外存中的文件里

13. 假设 B 是 A 的 public 继承的派生类，f 是 A 类的 public 成员函数，g 是 B 类新定义的 public 成员函数。则以下哪些语句是没有问题的（ C ）
A a;
B b;
a.g(); // (1)

A *p = &b; // (2)
b = a; // (3)
void func1(A *p);
func1(&b); // (4)
A. （1）（2）
B. （2）（3）
C. （2）（4）
D. （3）（4）

14. 有如下类定义

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

则类 MyDERIVED 中保护成员个数是（ ）

A. 4

B. 3

C. 2

D. 1

15. 设有一个类为 A，现希望为 A 类重载=运算符，并希望能实现如下形式的连续赋值 A a, b, c; a=b=c;且希望有较高的效率，则=运算符的最佳原型应为（ ）
A．A A::operator=(A a);
B．A A::operator=(const A&);
C．A& A::operator=(A a);
D．A& A::operator=(const A& a);

二．程序分析题(含 6 个小题，计 26 分)

16. 填写程序，利用引用类型实现交换两个 int *型指针变量的值。（本题 4 分）

```
#include <iostream>
using namespace std;
void swap(___ (1)_ __) //交换两个 int *型指针变量的值
{    int *t;
     _____(2)_____

}
int main()
{    int a=0,b=1;
     int *p=&a,*q=&b;
     cout << *p << ',' << *q << endl;   //输出：0,1
     swap(p,q);
     cout << *p << ',' << *q << endl;   //输出：1,0
     return 0;
}
```

(1) int* &a, int* &b

(2) t=a; a=b; b=t

17. 填写程序，实现二维数组的求和操作。（本题 4 分）

```
int sum(____(1)_ __, int num) //2 分
{    int s=0;
     for (int i=0; i<num; i++) s += x[i];
     return s;
}
.....
int a[10][5],b[40][20];
.....
cout << sum(____(2)_a__,10*5) ；//1 分
cout << sum(____(3)_b__,40*20)；//1 分
```

int *x / int x[]

18. 阅读程序回答问题。（本题 4 分）

```
class A
{        int x,y;
         char *p;
     public:
          A(char *str)
          {    x = 0;    y = 0;
               p = new char[strlen(str)+1];
               strcpy(p,str);
          }
          ~A() { delete [] p; p=NULL; }
```

};
.....
A a1("abcd");
A a2(a1);

（1） 以上代码存在什么问题？（2 分）无拷贝构造函数 a2无新空间.
（2） 如何解决？（2 分）
为一个、（注意重点在新声明空间）

19. 阅读下列程序，写出程序具体调用函数。（本题 6 分）

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

class A{
public:
    A() { f(); }
    virtual ~A();
    virtual void f();
  void g();
    void h() { f(); g(); }
};

class B: public A{
public:
    ~B();
    void f();
    void g();
};

void main(){
    B b;            //调用 B::B(), A::A()和 A::f
    A *p;
    p=&b;
    p->f();         //调用 B::f
    p->A::f();      //调用 A::f
    p->g();         //调用 A::g
    p->h();         //调用_____(1)__A::h(), B::f(), A::g()
    p = new B; //调用_____(2)__调用 B::B(), A::A()和 A::f
    ...
    delete p;   //调用_____(3)___
}                          B::~B(), A::~A()
```

important

总结一下虚析构函数的作用：

（1）**如果父类的析构函数不加virtual关键字**

当父类的析构函数不声明成虚析构函数的时候，当子类继承父类，父类的指针指向子类时，delete掉父类的指针，只调动父类的析构函数，而不调动子类的析构函数。

（2）**如果父类的析构函数加virtual关键字**

当父类的析构函数声明成虚析构函数的时候，当子类继承父类，父类的指针指向子类时，delete掉父类的指针，先调动子类的析构函数，再调动父类的析构函数。

20　填写程序，完成文件输入。（本题 4 分）

```
#include __(1)__<fstream>
#include <iostream>
using namespace std;

struct Student
{ int no;
    char name[10];
    int scores[5];
} s1;

void main(){
    //以二进制方式输入数据      ios::binary ← 此打开方式为二进制,更严谨.
  ifstream in_file("d:\\students.dat",__(2)__);
    if(_____(3)____) {  (3) !in_file
        cerr<<"Fail to open file"<<endl;
        exit(-1);
    }                  s1, sizeof(s1)
    in_file.read(_____(4)____);
    in_file.close();
}
```

21.　完成如下程序。（本题 4 分）

```
#include <iostream>
using namespace std;
            T, int size
template <__(1)___ >
class Stack
{       T buffer[size];  0~size-1
        int top;
    public:
        Stack() { top = -1; }
        bool push(const T &x)
        {                    size-1
            if (top ==__(2)___)
            {   cout << "Stack is overflow.\n";
                return false;
            }
            else
            {   top++; buffer[top] = x ;
                return true;
            }
```

```
        }
        bool pop(T &x)
        {
            if (top ==-1)
            {   cout << "Stack is empty.\n";
                return false;
            }
            else
            {   __(3)___; top--;
                return true;
            }
    }

};
int main()
{
    double x;
    Stack< __(4)___> st1;// st1 为元素个数为 100 的 double 型栈
    st1.push(10.0);
    st1.pop(x);
    ......
}
```

三．简答题 (含 4 个小题，每小题 4 分，计 16 分)

22. 在面向对象程序设计中，如何理解数据的抽象与封装。

23 拷贝构造函数的作用是什么？何时会调用拷贝构造函数？

24 C++怎样实现消息的动态绑定，请简单说明下实现过程。

25. C++标准模板库（STL）中包含哪几类模板？它们的作用分别是什么？

四．设计题 (含 3 个小题，计 28 分)

26. 编写类 String 的构造函数、析构函数、赋值函数，以及测试的 main 函数。（本题 9 分）
已知类 String 的原型为：

```
#include  <iostream >
#include  <string.h>
class  String
{public:
String(const  char  *str=NULL);  //  普通构造函数
String(const  String  &other);  //  拷贝构造函数
~String();  //  析构函数
String  &  operator=(const  String  &other);  //  赋值函数
void  show()
{cout<<m_data<<endl;
}
private:
char  *m_data;  //  用于保存字符串
};
```

27. 定义一个抽象立体图形类(Geometry)，成员有立体图形名称(name)、求体积的抽象方法 getVolume()，将它作为基类派生出球体类、长方体类和圆柱体类，实现这些派生类的方法 getVolume()，并在 main 函数中求它们的体积之和。要求用基类指针数组，使它每一个元素指向一个派生类对象。（本题 9 分）

28. 编写一个学生成绩输入/输出程序。该程序从键盘输入学生基本信息及各门课的成绩，然后把它们保存在文件中。内容包括：学号、姓名、选课门数以及各门课的成绩。要求通过重载操作符">>"和"<<"来实现学生信息的输入/输出。（本题 10 分）

学生成绩类的定义如下：

```
const int MAX_NUM_OF_COURSES=30;
const int MAX_ID_LEN=10;
const int MAX_NAME_LEN=8;
class StudentScores
{    public:
          StudentScores() { initialized = false; }
          bool data_is_ok() const { return initialized; }
     private:
          int scores[MAX_NUM_OF_COURSES],num_of_courses;
          char id[MAX_ID_LEN+1],name[MAX_NAME_LEN+1];
          bool initialized;
     friend istream &operator >>(istream &in, StudentScores &x);
     friend ostream &operator <<(ostream &out, StudentScores &x);
};
```

6.

```
#include <iostream>
#include <string.h>
using namespace std;
class String {
    public:
        String(const char *str=NULL); // 普通构造函数
        String(const String &other); // 拷贝构造函数
        ~String(); // 析构函数
        String& operator=(const String &other); // 赋值函数
        void show() {
            cout<<m_data<<endl;

}       friend ostream& operator<<(ostream&out,String &ob);
//本题不要求 仅作测试！友元函数重载<<的友元声明
        friend istream& operator>>(istream&in,String &ob);
//本题不要求 仅作测试！友元函数重载>>的友元声明
    private:
        char *m_data; // 用于保存字符串
};
//under this is my answe
#define SIZE 50
String::String(const char *str){
    m_data = new char [SIZE];
}
String::String(const String &other){
    m_data = new char [strlen(other.m_data)+1];
    for(int i=0;i<=strlen(other.m_data);i++){
        m_data[i] = other.m_data[i];

}
String::~String(){
    delete [] m_data;
}
String& String::operator=(const String &other){ //利用this指针
    if(strlen(other.m_data)>=50){
        delete []m_data;
        m_data = new char [strlen(other.m_data)+1];

}   for(int i=0;i<=strlen(other.m_data);i++){
        m_data[i] = other.m_data[i];

}
ostream& operator<<(ostream&out,String &ob) {//本题不要求 仅作测试！
    out<<ob.m_data;
    return out;
}

istream &operator>>(istream &in, String &ob) {//本题不要求 仅作测试！
    //记得将原有数据擦除
    if(ob.m_data!=NULL) {
        delete ob.m_data;
        ob.m_data=NULL;

}   char buf[SIZE]="";//临时buf
    in.getline(buf,SIZE);
//先得到键盘输入的数据 然后根据buf的实际大小开辟空间，最多读1023最后一位补\0
    ob.m_data=new char [strlen(buf)+1];
    strcpy(ob.m_data,buf);
    return in;
}
int main(){
    String a;
    cin >> a;
    String b=a;
    String* d = new String;
    cout << a << endl;//本题不要求 仅作测试！
    cout << b << endl;//本题不要求 仅作测试！
    delete d;
}
// 也调试了好几个bug才改对...考试的时候写出大概框架即可 老师看不出来 （逃
```

2).

```cpp
#include <iostream>
#include <string>
using namespace std;
class Geometry{
    string name;
    public:
        virtual double getVolume()=0;
} ;
class Ball:public Geometry{
    double r;
    public:
        double getVolume(){
            return 4.0/3*3.14*r*r*r;

}       void setr(double R){
            r=R;

};
class Rectangle:public Geometry{
    double a,b,h;
    public:
        double getVolume(){
            return a*b*h;

}       void set(double A,double B,double H){
            a=A;b=B;h=H;

};
class YuanZhuZhu:public Geometry{
    double r,h;
    public:
        double getVolume(){
            return 3.14*r*r*h;

}       void set(double R,double H){
            r=R;h=H;

} ;
int main(){
    Ball hx;
    Rectangle cyb;
    YuanZhuZhu hyx;
    Geometry* tmp[3];//基类指针数组
    hx.setr(1.0);
    cyb.set(1,1,1);
    hyx.set(1,1);
    tmp[0]=&hx;tmp[1]=&cyb;tmp[2]=&hyx;
    for(int i=0;i<=2;i++){
        cout << tmp[i]->getVolume() << endl;
    }
}
```

28.

```cpp
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
const int MAX_NUM_OF_COURSE =30;
const int MAX_ID_LEN=10;
const int MAX_NAME_LEN=8;
class StudentScore  {
    public:
        StudentScore () {
        s   initialized = false;

}       bool data_is_ok() const {
            return initialized;

}   private:
        int scores[MAX_NUM_OF_COURSE ],num_of_courses;
        char id[MAXSID_LEN+1],name[MAX_NAME_LEN+1];
        bool initialized;
        friend istream &operator >>(istream &in, StudentScore  &x);
        friend ostream &operator <<(ostream &out,sStudentScore  &x);
};                                              s
istream &operator >>(istream &in, StudentScore  &x){
    //读入时会有bug 不影响使用 不改了s
    cout << "请输入学号：" ;
    in >> x.id;
    cout << "请输入姓名：" ;
    in >> x.name;
    cout << "请输入选课门数：" ;
    in >> x.num_of_courses;
    if(x.num_of_courses>MAX_NUM_OF_COURSE ){
        cout << "选课超过s最大数量，修改为30个！" ;
        x.num_of_courses = MAX_NUM_OF_COURSE ;
                           S
}   for(int i=1;i<=x.num_of_courses;i++){
        cout << "请输入第" << i << "门课的成绩：" ;
        in >> x.scores[i];

}   x.initialized = true;
    return in;
}
ostream &operator <<(ostream &out, StudentScore  &x){
//  版本1：适合考试的时候写的，给老师看比较清楚，但实际上是错的（推荐考试写这种）
  （逃
    out << endl << "学生姓名：" << x.name << endl << "学号：" << x.id << endl
 << "选课门数：" << x.num_of_courses << endl;
    for(int i=1;i<=num_of_course  i++){
        out << "第" << i << "门课：" << scores[i] << "points." << endl;

}   return out;
//  版本2：正确并能使用的版本：
    out << x.id << endl << x.name << endl << x.num_of_courses << endl;
    for(int i=1;i<=x.num_of_courses;i++){
        out << x.scores[i] <<  endl;

}   return out;
}
int main() {
    //实在懒 这题太麻烦了 就写存取一个学生的程序。。。
    StudentScore  tmp;
    ifstream fout("F:\\temp.ini");
    if(!fout){
        cout << "未读到文件存储的学生信息，将直接开始写入！"   << endl;
    } else{
        //命令行会有奇怪的输出 别在意就行
        fout >> tmp;
        cout << endl << tmp;

}   fout.close();
    ofstream readin("F://temp.ini");
    cin >> tmp;
    readin << tmp;
}
```