中山大学计算机科学系 2006 级

计算机科学与技术专业、网络工程专业、信息安全专业（ABCDE 班）

程序设计　B 卷

学号 _____　姓名 _____　成绩_____

(试卷共 6 页，答案全写在答题纸上，交卷时连试卷一同交回)

考试形式：闭卷　　任课老师：林瑛、肖菁、杨永红　　2007-6

警示 **《中山大学授予学士学位工作细则》第六条："考试作弊不授予学士学位。"**

一、单项选择 (每小题 1 分，共 15 分)

1. 派生类的对象可访问：
   A) 公有继承的基类公有成员　　B) 公有继承的基类私有成员
   C) 公有继承的基类保护成员　　D) 私有继承的基类公有成员

2. 下面对结构或类中成员的访问不正确的是：
   A). *p.salary (p 为指向类对象的指针)　　B) p->salary (p 为指向类对象的指针)
   C) (*p).salary (p 为指向类对象的指针)　　D) Worker.salary (Worker 是类类型的对象)

3. 在 C++语言中，下列哪个关键字不能提供封装：
   A) struct　　B) union
   C) extern　　D) class

4. 指针 ptr 声明为 double* ptr。假设 ptr 当前的值是 ADDR，则表达式 (*ptr + 1) 的值是：
   A) ADDR + 1　　B) ADDR + 4
   C) ADDR + 8　　D) 无法确定

5. 下列对模板的声明，正确的是：
   A) template<T>　　B) template<class T1,T2>
   C) template<class T1;class T2>　　D) template<class T1,class T2>

6. 在 C++语言中，以下哪个表达式采用了十六进制表示整型常量：
   A) k = 0123 ;　　B) k = '\123' ;　　C) k = 123 ;　　D) k = 0x123 ;

7. 下列将类 A 说明为类 B 的虚基类的语句中，正确的是：
   A) class B: private A virtual　　B) class B: private virtual A
   C) class B: virtual private A　　D) virtual class B: private A

8. 下面哪一种是对类的析构函数的定义：
   A) ~X::X(参数)　　B) X::~X(参数)　　C) int ~X::X()　　D) X::~X()

9. 友员运算符 @obj 被 C++编译器解释为 (@表示某种运算符)：
   A) operator@(obj)　　B) operator@(obj,0)
   C) obj.operator@()　　D) obj.operator@(0)

10. 关于虚函数的描述中，正确的是：
   A) 虚函数是一个 static 类型的成员函数　　B) 虚函数是一个非成员函数
   C) 基类中说明了虚函数后，派生类中对同名函数的重定义时可不必说明为 virtual，其虚特性保持不变
   D) 派生类重定义的虚函数与基类的同名虚函数具有不同的参数个数和类型

11. 运算结果类型相同的是：

A) 9/2.0 和 9/2　　B) 9.0/2.0 和 9.0/2

C) 9.0/2 和 9/2　　D) 9/2 和 9.0/2.0

12. while (!x) 中的 !x 与下面哪个条件等价：

A) x==1　　B) x!=1　　C) x!=0　　D) x==0

13. 设有如下声明的类 FOO：

```
class FOO {
                private:
                                float std;
                                static float max, min;
};
```

则表达式 sizeof(FOO)的值为：

A) 4　　B) 8　　C) 12　　D) 16

14. 不能重载的运算符是：

A) :: [] ?:　　B) . :: ->　　C) () ?: #　　D) :: ?: #

15. 关于析构函数不正确的说法是：

A) 析构函数在对象生存期结束时自动调用

B) 一个类可以有多个析构函数

C) 析构函数不得指定参数

D) 析构函数是类中的公有成员函数

二、程序改错：指出以下题目所示程序段的语法错误(请通过行号来指出错误位置)，说明其错误原因并改正之！(8 个错误，每个错 2.5 分：指出错误位置 0.5 分；错误原因 1 分；改正 1 分，共 20 分)

1. 下列程序包含了 4 个错误，请在**不修改主函数**（假设主函数完全正确）以及**不添加任何函数**的前提下改正之：

```
(1)  class MyClass{
(2)  public:
(3)        MyClass(int ini) { member = ini; }
(4)        int GetMember() const { return member; }
(5)  private:
(6)        auto int member;
(7)        void SetMember(int m) { member = m;}
(8)  }
(9)  void main()
(10) {     MyClass obj1;
(11)       MyClass obj2(3);
(12)       obj1.SetMember(10);
(13) }
```

2. 下列程序包含了 4 个错误：

```
(1)  #include <iostream.h>
(2)  #include <string.h>
(3)  class PERSON {
(4)        public:
(5)              PERSON(char* name)
(6)              { int len;
(7)                len = strlen(name);
(8)                PERSON::name = new char[len + 1];
(9)                strcpy(PERSON::name, name);
(10)             }
(11)             ~PERSON() { delete []name; }
(12)             char* get_name() { return name; }
(13)       protected:
```

```
(14)             char* name;
(15) };
(16) class STUDENT: PERSON {
(17)       public:
(18)             void STUDENT(char* st_name, int scr)
(19)                   : score(scr) ,PERSON(st_name)
(20)             { int len;
(21)               len = strlen(st_name);
(22)               name = new char[len + 1];
(23)               strcpy(name, st_name);
(24)             }
(25)             int get_score() { return score; }
(26)       protected:
(27)             int score = 100;
(28) };
(29) void main()
(30) {     STUDENT soft("Zhao", 80);
(31)       cout << "Student: " << soft.get_name();
(32)       cout << "Score: " << soft.score << endl;
(33) }
```

三、**程序输出(共 30 分)：**写出下列程序的输出结果

1. (3 分)

```
#include<iostream.h>
void main()
{  int a;
   int &b=a;
   b=10;
   cout<<"a="<<a<<endl;
}
```

2. (8 分)

```
#include <iostream.h>
class CLASSNAME {
public:  CLASSNAME()
         {   cout << "\nCall Construction of CLASSNAME.";
         }
         int print()
         {   cout << "\nThis is CLASS_NAME."; return 1;
         }
         ~CLASSNAME()
         {   cout << "\nCall Destruction of CLASSNAME.";
         }
};

CLASSNAME object_name;

int main()
{  cout << "\nProgram begin...";
   object_name.print();
   cout << "\nProgram end...";
   return 1;
}
```

3. (6 分)

```
#include <iostream.h>
```

```
template<class TYPE>
TYPE max(TYPE k, TYPE t)
{  cout<< ”Calling generic version of max(” << k << ”,” << t
       << "),the max is :" ;
   return (k>t?k:t);
}

int max(int k, int t)
{  cout<< ”Calling special version of max(” << k << ”,” << t
       << "),the max is :" ;
   return (k>t?k:t);
}
void main()
{  cout<<max(10, 20)<<"\n";
   cout<<max(10.5, 20.7)<<"\n";
   cout<<max('B', 'A')<<"\n";
}
```

4. (13 分)

```
#include<iostream.h>
#include <string.h>
class CARTOON{
   public :
      CARTOON(char *name = "NULL")
      {   strcpy(CARTOON::name,name);
          cout<<"Construct: Cartoon ["<<name<<"]\n";
      }
      CARTOON(const CARTOON& other)
      {   strcpy(name,other.name);
          cout<<"Copy Construct: Cartoon ["<<name<<"]\n";
      }
      ~CARTOON()
      {   cout<<"Destruct: Cartoon ["<<name<<"]\n"; }
      CARTOON operator= (const CARTOON& other)
      {   if (&other==this) return *this;
          cout << "Calling operator =, set [" << name;
          cout << "] equal to [" << other.name << "]" << endl;
          strcpy(name, other.name);
          return *this;
      }
   protected:
      char name[30];
};

class MOUSE: public CARTOON {
   public:
      MOUSE(char* name = "Mickey"): CARTOON(name)
      {   cout << "Construct: MOUSE [" << name << "]\n";}
      ~MOUSE()
      {   cout<<"Destruct: MOUSE ["<<name<<"]\n";}
      void set(CARTOON other)
      {   friends = other;
      }
   protected:
      CARTOON friends; // 老鼠的朋友
};

void main()
{  CARTOON   duck("Donald");
```

```
    MOUSE     mouse;
    mouse.set(duck);
}
```

四、程序填空（每空 2 分，共 16 分）：根据以下各小题的描述和要求在指定位置填入适当语句

1. 在类 FOO 中，用成员函数重载运算符==，用于判断两个 FOO 类的对象是否相等，请完成重载函数的定义：

```
class FOO{
    public:     int operator == ( ____①____ right ) const
                {      return ____②____ ; } // right 是形参的名字
    private:    int m1,m2;
};
```

2. 完成如下的程序，使得输出结果为：

```
base::10
base::12
derived::24
```

```
#include <iostream.h>
class base {
        int x;
public: base(int a) { x=a; }
        ____③____{ cout<<"base::"<< x << endl; }
};
class derived: public base {
        int y;
public: derived(int a,int b):base(a) { y=b; }
        void print() { ____④____; cout<<"derived::"<<y<<endl; }
};
void main()
{  base b(10), *p;
   derived d(12,24);
   b.print();
   ____⑤____;
   p->print();
}
```

3. 打印出 2 至 99 之间的所有素数（即不能被任何数整除的数）

```
#include<iostream.h>
#include<math.h>
void main( )
{  int i,n;
   for(n=2;____⑥____;n++){
      int temp=int(sqrt(n));   //求出 n 的平方根并取整
      for(i=2;____⑦____;i++)
      if(n%i==0)____⑧____;
      if(i>temp) cout<<n<<' ';
   }
   cout<<'\n';
}
```

五、程序设计（19 分）

1. (10 分)类属类 *SET* 描述一个集合。集合中的元素采用一个单向链表表示。*SET* 提供了判断元素从属的成员函数 *has*()、以友元形式重载了运算符*+*和***表示集合的并与交。*SET* 的用法如以下程序所示。要求给出 *SET* 的类界面。

   提示：注意在主程序中使用到该类的功能，尽可能完善该类属类的操作，不必提供 SET 的类实现。

```
#include <iostream.h>
#include "set.hpp"
int main()
{  SET<int> s1, s2, s3;
   SET<int>* s_ptr;
   ……
   s1 = s1 + s2 * s3;
   s_ptr = new SET<int>(s1);
   if (s1.has(0))  cout << *s_ptr * s2;
   delete s_ptr;
   return 0;
}
```

2. (9 分)设计一个日期类 DATE，根据以下的类界面：

```
class DATE{
   int year, month, date ;
public:
   DATE(int y = 1900,int m = 1, int d = 1){ year = y; month = m; date = d; }
   DATE(const DATE &other);
   DATE operator=(const DATE &other);
   DATE operator-();
   friend DATE operator+( DATE a, DATE b);
};
```

请写出类 DATE 声明的后 4 个成员函数的定义（函数的实现）。