

中山大学计算机科学系 2006 级

程序设计 B 卷 答案

一、单项选择 (每小题 1 分, 共 15 分)

A A C D D

D C D A C

B D A D B

二、程序改错 (8 个错误, 每个错 2.5 分, 共 20 分): 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!

(每个错 2.5 分——错误位置 0.5 分、错误原因 1 分、改正 1 分, 共 20 分)

题号	错误所在行	错误	改正
1	1) 第 (3) 行	因第 (10) 行: MyClass obj1; 要使用无参数的构造函数, 但类中并没有提供此构造函数,	因不可修改主函数和添加函数, 所以第 (3) 行构造函数的定义中必须为形参提供缺省值, 改为: MyClass(int ini=0) { member = ini; }
	2) 第 (6) 行	类的数据成员在声明时不可用 auto、register、extern 修饰	把 auto 去掉, 改为: int member;
	3) 第 (8) 行	类的定义少了分号	添加分号, 改为: };
	4) 第 (12) 行	SetMember 是 MyClasee 类的私有成员函数, 不可在类外调用	把第 (7) 行 SetMember 函数的声明移到类的公有段。
2	1) 第 (18) 行	STUDENT 的构造函数不可有返回类型,	STUDENT(char* st_name, int scr)
	2) 第 (27) 行	STUDENT 中不能对数据成员 score 进行初始化	int score;
	3) 第 (31) 行	由于私有派生, soft 的成员函数 get_name() 是私有的, 不可访问	改为公有派生: 16) 改为 class STUDENT: public PERSON{
	4) 第 (32) 行	main() 中 soft 的保护数据成员 score 不可访问	改为调用 get_score() 公有函数: 31) 改为 cout << "Score: " << soft.get_score() << endl;
【备注说明】 19) 行: STUDENT 构造函数中的 score(scr) 没有错误, 如学生指错则扣 1 分			

三、程序输出 (共 30 分):

1. a=10 (3 分)

2. 第 1、3、5 行每行 2 分, 第 2、4 行每行 1 分, 共 8 分

Call Construction of CLASSNAME.

Program begin...

This is CLASSNAME.

Program end...

Call Destruction of CLASSNAME

3. 输出结果: (每行 2 分: 输出调用函数提示 1 分; 最大值 1 分, 共 6 分)

Calling special version of max(10,20),the max is 20

Calling generic version of max(10.5,20.7),the max is 20.7

Calling generic version of max(B,A) ,the max is B

4. 每行 1 分, 共 13 分

```
Construct: Cartoon [Donald]
Construct: Cartoon [Mickey]
Construct: Cartoon [NULL]
Construct: MOUSE [Mickey]
Copy Construct: Cartoon [Donald]
Calling operator =, set [NULL] equal to [Donald]
Copy Construct: Cartoon [Donald]
Destruct: Cartoon [Donald]
Destruct: Cartoon [Donald]
Destruct: MOUSE [Mickey]
Destruct: Cartoon [Donald]
Destruct: Cartoon [Mickey]
Destruct: Cartoon [Donald]
```

四、程序填空 (共 16 分, 每空 2 分)

- ① const FOO&
- ② m1 == right.m1 && m2 == right.m2
- ③ virtual void print()
- ④ base::print()
- ⑤ p=&d
- ⑥ n<=99
- ⑦ i<=temp
- ⑧ break

五、程序设计 (19 分)

1. (10 分) 以下成员是为了满足给定主函数的应用, 该类必须提供的, 缺少的的话相应扣分

```
template <class ELEMENT>          // 1 分
class SET {
public:
    SET();                          // 1 分
    SET(const SET& other);          // 1 分
    ~SET();                        // 1 分
    int has(ELEMENT obj);          // 1 分
    SET operator =(const SET& other); // 1 分
    friend SET operator +(const SET& left, const SET& right); // 1 分
    friend SET operator *(const SET& left, const SET& right); // 1 分
protected:
    struct NODE {                  // 1 分
        ELEMENT element;
        NODE* link;
    };
    NODE* first;                  // 1 分
};
```

2. (9 分)

DATE:: DATE (const DATE &other) // 2 分

```
{    year = other.year;
    month = other.month;
    date = other.date;
}
```

DATE DATE::operator=(const DATE &other) // 3 分

```
{    year = other.year;
    month = other.month;
    date = other.date;
    return *this;
}
```

DATE DATE::operator-() // 2 分

```
{    COMPLEX t;
    t.year = - year;
    t.month = - month;
    t.date = - date;
    return t;
}
```

DATE operator+(DATE a, DATE b) // 2 分

```
{    COMPLEX t;
    t.year = a.year + b.year;
    t.month = a.month + b.month;
    t.date = a.date + b.date;
    return t;
}
```