

华中科技大学计算机科学与技术学院 2020~2021 第一学期

“C++程序设计”考试试卷 (A 卷)

考试方式 开卷 考试日期 2020-11-14 考试时长 150 分钟

专业班级 _____ 学 号 _____ 姓 名 _____

题号	一	二	三	四	五	六	总分	核对人
分值	15	20	20	15	15	15	100	
得分								

分 数	
评卷人	

一、单选题：请从 4 个选项选择一个最合适的选项作为答案（15 分：每小题 3 分）。

解答内容不得超过装订线

1. 关于定义 “struct A {int x; mutable int y; const a={1,3};” , 如下叙述哪个 B 正确:
A. a.x 可被赋值, a.y 不可被赋值 B. a.x 不可被赋值, a.y 可被赋值
C. a.x 和 a.y 均不可被赋值 D. a.x 和 a.y 均可被赋值
2. 关于 union 定义的类的叙述 A 正确:
A. 既不能是基类也不能是派生类 B. 不能是基类, 但可以是派生类
C. 可以是基类, 但不能是派生类 D. 既可以是基类, 也可以是派生类
3. 对于说明 “int &f(); int &&g();” 及其函数调用 f() 和 g(), 如下 4 个叙述 D 正确:
A. 调用 f() 和调用 g() 均不可被赋值 B. 调用 f() 不可被赋值, 调用 g() 可被赋值
C. 调用 f() 和调用 g() 均可被赋值 D. 调用 f() 可被赋值, 调用 g() 不可被赋值
4. 对于定义 “struct A { int f(); }”, 关于 int 前面是否可用 static 和 virtual, 如下的叙述 C 错误:
A. 可以只使用 static B. 可以只使用 virtual
C. 必须同时使用 static 和 virtual D. 都不对(错)
5. 对于定义 “char *const &f();”, 如下哪个语句是错误的 A :
A. f()=(char*) "abcd"; B. *f()='A';
C. char *p=f(); D. *f()="ABC"[1];

分 数	
评卷人	

二、在最多使用单重作用域例如 A::x 的前提下, 在空白处填写以下各类可被访问的成员及其访问权限 (20 分: 根据正确回答的成员个数按比例计算给分)。

```
class A {  
    int a;  
protected:  
    int b, c;  
public:
```

//类 A 的可访问成员: a, b, c, d, e
//私有成员: a
//保护成员: b, c
//公有成员: d, e

```

    int d, e;
};
class B: protected A { //类 B 的可访问成员:
    int a;             //私有成员: a
protected:           //私有成员:
    int b, f;          //保护成员: b, f, A::d, A::e, A::b, A::c
    using A::d;        //保护成员:
public:               //公有成员: e, g
    int e, g;          //公有成员:
};
struct C: A {         //类 C 的可访问成员:
    int a;             //私有成员: 无
protected:           //私有成员:
    int b, f;          //保护成员: b, f, A::b, A::c
public:               //保护成员:
    int e, g;          //公有成员: a, e, g, A::d, e
    using A::d;        //公有成员:
};
struct D: B, C {       //类 D 的可访问成员:
    int a;             //私有成员:
protected:           //私有成员: b, f; B::b, f; A::b, c, d, e; C::b, f,
    int b, f;          //保护成员: A::b, c
public:               //保护成员: a, e, g; B::e, g;
    int e, g;          //公有成员: A::d, e
};

```

分 数	
评卷人	

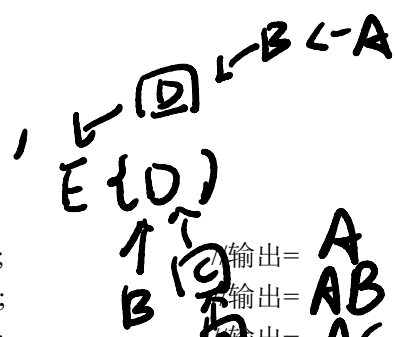
三、回答 main 中每行语句的输出结果 (20 分: 前四个语句的输出每个 3 分, 后两个语句的输出每个 4 分)。

```

#include <iostream>
using namespace std;
struct A { A() { cout << 'A'; } };
struct B { A a; B() { cout << 'B'; } };
struct C : virtual A { C() { cout << 'C'; } };
struct D : B, virtual C { D() { cout << 'D'; } };
struct E : virtual A, virtual D {
    D d;
    E() : A() { cout << 'E'; }
};
struct F : virtual C, B, virtual D, virtual E {
    D d; E e;
    F() { cout << 'F'; }
};

```





```
void main() {
```

```
  A a; cout << '\n';
```

```
  B b; cout << '\n';
```

```
  C c; cout << '\n';
```

```
  D d; cout << '\n';
```

```
  E e; cout << '\n';
```

```
  F f; cout << '\n';
```

输出=

输出=

输出=

输出=

输出=

输出=

A

AB

AC

ACABD

ACABDACA

E

ACABDACA

}

分 数	
评卷人	

四、综合分析并指出以下程序中下划线位置可能出现的语法错误及其原因 (共 15 分: 每错约 1 分)。

BIDEABACAABDACA

BDEF

```
class A {
```

```
  int a;
```

```
protected:
```

```
  virtual ~A() {}
```

```
public:
```

```
  int& b;
```

```
  int c;
```

```
  virtual A(*g)()=0;
```

```
  virtual A(int x) { a = x; };
```

```
  a = (4, 3);
```

```
class B: A {
```

```
  int d;
```

```
  using A::a;
```

```
public:
```

```
  friend int operator()(int) { return 2; };
```

```
  B(int x, int y, int z) { d = x + y + z; };
```

```
  b(5, 6, 7);
```

```
class C: B {
```

```
  int z;
```

```
public:
```

```
  ~C(int x) { z = x; };
```

```
  c;
```

```
void main() {
```

```
  int A::* p = &A::b;
```

```
  int i = a.a;
```

```
  int && y = i;
```

```
  i = b.b;
```

```
  i = i + c.d;
```

```
  i = b.*p;
```

指针: virtual 不定
义数据成员

//错误1:

//错误2: 构造函数返回

//错误4:

//错误6:

//错误7:

//错误9:

//错误10:

//错误11:

//错误12:

//错误13:

//错误14:

//错误15:

//错误16:

//错误17:

//错误18:

//错误19:

//错误20:

//错误21:

//错误22:

//错误23:

//错误24:

//错误25:

//错误26:

//错误27:

//错误28:

//错误29:

//错误30:

//错误31:

//错误32:

//错误33:

//错误34:

//错误35:

//错误36:

//错误37:

//错误38:

//错误39:

//错误40:

//错误41:

//错误42:

//错误43:

//错误44:

//错误45:

//错误46:

//错误47:

//错误48:

//错误49:

//错误50:

//错误51:

//错误52:

//错误53:

//错误54:

//错误55:

//错误56:

//错误57:

//错误58:

//错误59:

//错误60:

//错误61:

//错误62:

//错误63:

//错误64:

//错误65:

//错误66:

//错误67:

//错误68:

//错误69:

//错误70:

//错误71:

//错误72:

//错误73:

//错误74:

//错误75:

//错误76:

//错误77:

//错误78:

//错误79:

//错误80:

//错误81:

//错误82:

//错误83:

//错误84:

//错误85:

//错误86:

//错误87:

//错误88:

//错误89:

//错误90:

//错误91:

//错误92:

//错误93:

//错误94:

//错误95:

//错误96:

//错误97:

//错误98:

//错误99:

//错误100:

//错误101:

//错误102:

//错误103:

//错误104:

//错误105:

//错误106:

//错误107:

//错误108:

//错误109:

//错误110:

//错误111:

//错误112:

//错误113:

//错误114:

//错误115:

//错误116:

//错误117:

//错误118:

//错误119:

//错误120:

//错误121:

//错误122:

//错误123:

//错误124:

//错误125:

//错误126:

//错误127:

//错误128:

//错误129:

//错误130:

//错误131:

//错误132:

//错误133:

//错误134:

//错误135:

//错误136:

//错误137:

//错误138:

//错误139:

//错误140:

//错误141:

//错误142:

//错误143:

//错误144:

//错误145:

//错误146:

//错误147:

//错误148:

//错误149:

//错误150:

//错误151:

//错误152:

//错误153:

//错误154:

//错误155:

//错误156:

//错误157:

//错误158:

//错误159:

//错误160:

//错误161:

//错误162:

//错误163:

//错误164:

//错误165:

//错误166:

//错误167:

//错误168:

//错误169:

//错误170:

//错误171:

//错误172:

//错误173:

//错误174:

//错误175:

//错误176:

//错误177:

//错误178:

//错误179:

//错误180:

//错误181:

//错误182:

//错误183:

//错误184:

//错误185:

//错误186:

//错误187:

//错误188:

//错误189:

//错误190:

//错误191:

//错误192:

//错误193:

//错误194:

//错误195:

//错误196:

//错误197:

//错误198:

//错误199:

//错误200:

//错误201:

//错误202:

//错误203:

//错误204:

//错误205:

//错误206:

//错误207:

//错误208:

//错误209:

//错误210:

//错误211:

//错误212:

//错误213:

//错误214:

//错误215:

//错误216:

//错误217:

//错误218:

//错误219:

//错误220:

//错误221:

//错误222:

//错误223:

//错误224:

//错误225:

//错误226:

//错误227:

//错误228:

//错误229:

//错误230:

//错误231:

//错误232:

//错误233:

//错误234:

//错误235:

//错误236:

//错误237:

分 数	
评卷人	

五、请填入自己学号的最后一位十进制数字，计算 main 函数中变量 i 在每条赋值语句执行后的值 (共 15 分：每小题 2.5 分)。

```

int x = 填写自己学号最后一位十进制数 2, y = x + 3; 5
struct A {
    int x = ::x + 2; 4
    static int &y; 0
public:
    operator int( )const { return x + y; }
    int& v(int& x) {
        for (int y = 1; x < 301; x ^= y, y++)
            if (x > 300) { x -= 31; y -= 2; }
        return ++x;
    }
    A& operator++( ) { ++x; ++y; return *this; }
    A(int x, int y = ::y + 3) { A::y = y; }
};
int &A::y = ::y; (2, 10)
void main( ) {
    A a(2, 7), b(5); (5, 10)
    int i, &j = i, A::*p = &A::x;
    i = a.y;
    j = a.x;
    i = a.*p;
    i = ++a;
    i = b.y + ::y;
    (b.v(i) = 5) += 2;
}

```

45 11

10
4
16
22
7

分 数	
评卷人	

六、对于如下所有单词按升序排列的字典类 DCT，对其中的函数成员进行程序设计(共 15 分：每个函数 1.5 分)。

```

class DIC {
    char **const e; //用于存放字典的单词
    const int m; //能够存放的单词个数
    int r; //已经存放的单词个数
public:
    DIC(int m=1000); //创建最多存 m 个单词的字典，所有 e[i]置空指针
    DIC(const DIC &d); //根据已知字典 d 深拷贝构造新字典
    DIC(DIC &&d) noexcept; //根据已知字典 d 移动构造新字典
}

```

DIC& operator=(const DIC &d);	//深拷贝赋值运算符的重载
DIC& operator=(DIC &&d) noexcept;	//移动赋值运算符的重载
DIC& operator<<(const char *w);	//将单词插入字典并保持升序, 若有该单词则不插
operator int() const noexcept;	//获得字典实际存放的单词个数
const char* operator[] (int i) const noexcept;	//获得下标为 i 的单词
int operator() (const char *w) const noexcept;	//查找单词 w 在字典中的位置
~DIC() noexcept;	//析构字典

};

```

① DIC::DIC(int m):
    e(new char*[m]), m(e?m:0), r(0) {}
② DIC::DIC(const DIC &d):
    e(new char*[d.m]), m(e?d.m:0), r(d.r) {
        if(d.e) throw "memory"
        for(int h=0; h<r; h++)
            {e[h]=new char[strlen(d.e[h])+1];
             if(e[h]==nullptr) throw "memory";
             strcpy(e[h], d.e[h])
            }
    }

```

解答内容不得超过装订线