

# 中山大学

## 二〇一五年攻读硕士学位研究生入学考试试题

科目代码: 856

科目名称: 程序设计基础

考试时间: 12 月 28 日下午

考生须知

全部答案一律写在答题  
纸上, 答在试题纸上的不计  
分!

### 一. 单项选择题 (每题 2 分, 共 40 分)

1. 哪个关于 C 语言程序设计的说法是错误的?

- A) 一个 C 语言程序在运行前必须先编译
- B) 一个程序在运行前必须先被加载入主存储器
- C) 注释可以加快程序的运行速度
- D) 一个程序是一个指令的序列

2. 哪个关于 C 语言和 C++ 语言的关系是正确的?

- A) C 语言和 C++ 语言都是面向对象的语言
- B) C++ 语言比 C 语言更好
- C) C 语言和 C++ 语言能解决同样的问题
- D) C 语言和 C++ 语言是同一个语言

3. 哪个关于变量的说法是正确的?

- A) 函数参数表中的变量是静态变量
- B) 函数参数表中的变量是局部变量
- C) 类的成员变量是静态变量
- D) 类的成员变量是局部变量

4. 哪个是以下程序的正确输出?

```
int value = 0x3C;  
cout << value;
```

- A) 60
- B) 0x3C
- C) 42
- D) M

5. 哪行程序有编译错误?

- A) `char letter = 'A' + 'a';`
- B) `char letter = "A";`
- C) `char letter = 48;`
- D) `char letter = '\48';`

6. 以下循环的次数是?

```
for (int i = 3; i < 15; ++ i) {  
    cout << i << endl;  
}
```

- A) 11
- B) 12
- C) 13
- D) 14

7. 以下程序的错误是?

```
float squareRootOf2() {  
    float root = 1;  
    while (true) {  
        if (root * root == 2) return root;  
        root = (root + 2/root) / 2;  
    }  
}
```

- A) 程序有编译错误                      B) 程序可能出现死循环  
C) 除数为 0 错误                        D) 程序没有错误

8. 哪个关于循环的说法是正确的?

- A) 任何 do-while 循环可以用 while 循环代替  
B) for 循环比 while 循环更强  
C) while 循环的循环体肯定执行一次以上  
D) 每个程序必须有循环

9. 哪个不属于使用函数的好处?

- A) 提高程序的可读性                    B) 提高程序的速度  
C) 隐藏实现细节                        D) 代码重用

10. 哪个字符串常量不正确?

- A) "\\//HAT?"                            B) "\\//\\//hat?"  
C) "Is it \"correct\"?"                D) "I'm correct\n"

11. 哪个字符串初始化语句没有错误?

- A) char name = "L";                      B) char date[] = { "Mon" };  
C) char month[] = { 0 };                D) char year[4] = "2014";

12. 哪个情况下两个函数互相有歧义?

- A) 两个函数的名字相同  
B) 两个函数的签名相同  
C) 编译器不知道应该匹配哪个函数  
D) 编译器不知道应该调用哪个函数

13. 文件以读的方式打开时失败的可能原因是?

- A) 文件的长度为 0  
B) 文件的长度超过可用的内存大小  
C) 文件正在被别的程序读  
D) 文件正在被别的程序写

14. 哪个关于 C++ 语言构造函数的说法正确?

- A) 构造函数的名字必须与所在类相同  
B) 构造函数的返回值类型必须是对象  
C) 构造函数的返回值类型必须是 void  
D) 构造函数不能被重载(overload)

15. 哪个成员变量不能出现在类 A 中?

- A) A \* a;      B) A a[10];      C) A & a;      D) A \* & a;

16. 下面语句中类 A 的构造函数被调用的次数是

A a, b[2], \* c[3];

- A) 1      B) 3      C) 4      D) 6

17. p 是一个对象的地址, 调用对象函数 x() 的正确语法是

- A) (\*p).x()      B) x(p)      C) \*p->x()      D) p->(x)

18. 哪个不是类的成员?

- A) 构造函数      B) 析构函数      C) 友员函数      D) 虚函数

19. 哪个是正确的常量函数声明?

- A) const void print()  
B) void const print()  
C) void print(const)  
D) void print() const

20. 要实现多态(polymorphism), 哪对功能是必须的?

- A) 数组+函数调用  
B) 子类地址可赋值给父类地址+虚函数  
C) 虚继承+纯虚函数  
D) 多重继承+函数重载

## 二. 概念题(每题 4 分, 共 40 分)

分别解释并比较以下各对概念

1. 局部变量、全局变量
2. 对象成员变量、类的静态变量
3. 函数、宏
4. 数组、地址
5. 数组、链表
6. 继承、多态
7. 递归、迭代
8. 堆、栈
9. 哈希表、平衡二叉树
10. 插入排序、归并排序

### 三. 输出分析题 (每题 5 分, 共 20 分)

请严写出程序的输出, 空格和换行不能省略。

所给出的程序都省略了头文件的包含, 程序都没有错误。

1. 请给出以下循环程序的输出

```
int main() {
    int number = 1;
    while (number < 100) {
        number *= 10;
        number -= 8;
        cout << number << endl;
    }
}
```

2. 请给出以下递归程序的输出

```
int fun(int num) {
    if (num < 5) return num;
    int num1 = num / 2;
    int num2 = num - num1;
    return fun(num1) * fun(num2);
}

int main() {
    for (int i = 5; i <= 10; ++ i) {
        cout << fun(i) << endl;
    }
}
```

3. 请给出以下使用 string 和 vector 程序的输出

```
void print(vector<string> & vec) {
    for (int i = 0; i < vec.size(); ++ i) {
        if (i != 0) cout << ",";
        cout << vec[i];
    }
    cout << endl;
}

int main() {
    vector<string> vec;
    string sentence("hey");
    vec.push_back(sentence);
    vec.push_back("you");
    print(vec);

    vec.pop_back();
    sentence += " hey";
    vec.push_back(sentence);
    sentence = "bye";
    vec.push_back(sentence);
    print(vec);
}
```

4. 请给出以下类定义程序的输出

```
class A
{
public:
    virtual void fun1() { cout << 1; }
    virtual void fun2() { cout << 2; }
    void fun3() { cout << 3; }
    void fun4() { cout << 4; }
};

class B : public A
{
public:
    virtual void fun1() { cout << 5; }
    void fun3() { cout << 6; }
};

class C : public B
{
public:
    virtual void fun2() { cout << 7; }
    void fun4() { cout << 8; }
};

int main() {
    C c;
    c.fun1(); c.fun2(); c.fun3(); c.fun4();
    cout << endl;

    B & b = c;
    b.fun1(); b.fun2(); b.fun3(); b.fun4();
    cout << endl;

    B d = c;
    d.fun1(); d.fun2(); d.fun3(); d.fun4();
    cout << endl;
}
```

#### 四. 错误分析题 (每题 10 分, 共 20 分)

每小题有 20 个错误。请找出各个错误的位置, 并且必须说明各个错误的原因。

```
1. static int getAverage(double & array, int size) const {
    double sum;
    for (i = 0, i <= size, ++ i) {
        sum = array(i);
    }
    sum /= array.size;
}

include <iostream>
use namespace stdio;

int main() {
    double array[] = 1,2,3,4,5;
    count >> getAverage(array, 6);
}

2. class A<T>
{
    const T t;
public:
    A(const T * t2) { t = t2; }
    virtual static print() const;
    friend static int get() const;
}

class B public A
{
public:
    B(int i) { }
    void print() { cout << t; }
};

int main() {
    A a = B();
    B & b = a;
}
```

#### 五. 程序设计题 (每题 15 分, 共 30 分)

1. 写一段程序返回一个实数数组的第三大的元素, 假设数组的长度必定大于 3。要求函数具有下面的函数头。

```
double getTheThirdLargest(double array[], int arraySize)
```

2. 写一个类 Student, 它继承以下两个接口, 并且使得以下 main 函数能正确工作。假设每个学生的名字少于 100 个字符。

程序中用到的 LIST.TXT 如下:

```
Bill Gates,12340001
Steve Jobs,12340002
Larry Page,12340003
Sergey Brin,12340004
```

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

class StudentInterface
{
public:
    virtual const char * getName() const = 0;
    virtual void setName(const char name[]) = 0;
    virtual int getID() const = 0;
    virtual void setID(int id) = 0;
};

class Serializable
{
public:
    virtual void read(istream & in) = 0;
    virtual void write(ostream & out) const = 0;
};

/*-----
请假设你所写的程序会被放在这里
-----*/

int main() {
    // read a list of 10 students from LIST.TXT
    ifstream input("list.txt");
    Student students[4];
    for (int i = 0; i < 4; ++ i) {
        Serializable & s = students[i];
        s.read(input);
    }
    input.close();

    // check the validity if the IDs
    for (int i = 0; i < 4; ++ i) {
        StudentInterface & stu = students[i];
        Serializable & ser = students[i];
        if (stu.getID() <= 0) {
            cout << "Invalid ID" << endl;
        }
        ser.write(cout);
    }
}
```