

科软复试手册

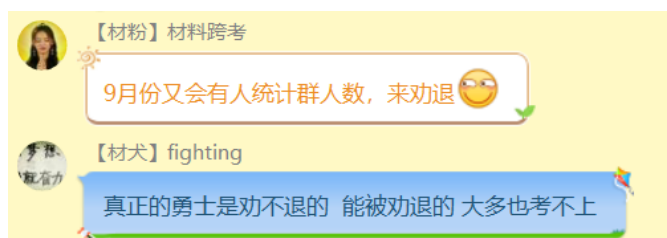
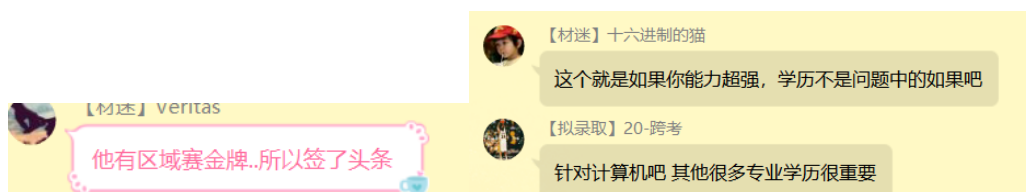
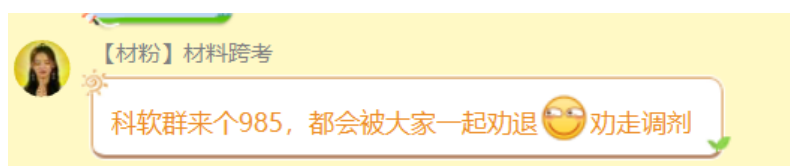
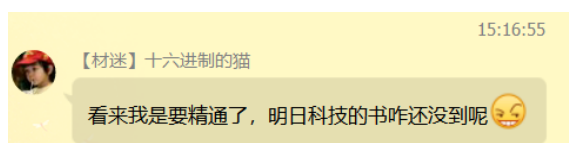
2019 第一版



【皮皮】
免费提供，不得销售

免费提供

2019 复试时间节点【预测】	3
2019 复试细则【招生简章】	5
专业基础测评（上机）	6
面试相关	66
2018 年硕士研究生招生（第一志愿）复试安排	93
2018 年调剂招生简章	98



2019 复试时间节点【预测】

初试成绩查询时间

18 年查询时间 2018 年 2 月 3 日

17 年查询时间 2017 年 2 月 16 日

根据目前各省份公布的信息

可以大致推测出今年初试成绩查询时间为 2019 年 2 月 15 日

国家线公布时间

2018 考研国家线—— 2018 年 3 月 16 日公布 (当年春节是 1 月 28 日)

2017 考研国家线—— 2017 年 3 月 15 日公布 (当年春节是 2 月 16 日)

2016 考研国家线—— 2016 年 3 月 11 日公布 (当年春节是 2 月 8 日)

可以大致推测出今年国家线公布时间为 2019 年 3 月 15 日

科软复试时间

2017 年 3 月 18 日中科大软院第一志愿复试

2017 年 3 月 20 日中科大软院调剂复试

2018 年 3 月 18 日中科大软院第一志愿复试

2018 年 3 月 24 日中科大软院调剂复试

可以大致推测 2019 年复试时间:

2019 年 3 月 18 日第一志愿复试

调剂系统开放时间

2018 年调剂系统: 3 月 23 日-4 月 30 日

2017 年调剂系统: 3 月 18 日-4 月 30 日

2019 年 4 月

在信息化平台填写入学方向

地址：<http://enroll.sse.ustc.edu.cn/ssers2014/stupages/regeist.aspx>

2019 年 7 月

发放录取通知书

1. 通知书上写的有关于报道及开学所有注意事项，请务必仔细看。
2. 通知书上写的有你是几等奖学金，没写就是没有

学业奖学金

国家助学金 6000 一年【均有】

另：

学院设立的优秀研究生奖学金，额度为金奖 4000 元、银奖 2000 元、铜奖 1000 元；

科大讯飞奖学金额度为特等奖 10000 元、一等奖 5000 元、二等奖 3000 元、优秀实习生奖 4000 元；

腾讯科技卓越奖学金设立特等奖 20000 元每人，优秀奖 10000 元每人；

江苏软件产业人才发展基金会设立江苏软件奖学金 8000 元每人。

上述各类奖项累计覆盖率约为 40%左右。

复试地点



2019 复试细则【招生简章】

一、**复试形式**：面试+上机。

二、**复试内容**：

1、专业基础测评（上机）：专业基础测评主要考察考生有关 C++ 和数据结构的基础知识及基本运用能力。本部分成绩为“通过/不通过”，只有“通过”考生方可录取。

2、英语综合面试（满分 50 分）：重点考核考生的英语听说能力。

3、专业综合面试（满分 100 分）：主要考察考生对本学科（专业）理论知识和应用技能的掌握程度，包括考生对本专业基础知识的掌握和理解程度，综合应用所学知识解决实际问题的能力；考察考生在本专业以外的学习、科研、社会实践或实际工作等方面的经历等。

三、**复试成绩**：满分 100 分，其中英语综合面试满分 50 分，专业综合面试满分 100 分，复试成绩=（英语综合面试成绩+专业综合面试成绩）÷1.5。

四、**最终成绩**：满分 100 分，初试成绩不计政治，复试成绩占比 50%，即最终成绩=（初试成绩【不计政治】÷4+复试成绩）÷2。

五、**录取**：按最终成绩由高到低排序，提出拟录取名单报批。为保证招生质量，报批人数可小于招生计划。

六、**调剂**：本专业在生源不足的情况下接受调剂。调剂信息将于复试阶段在中国科大研究生招生在线网站（<http://yz.ustc.edu.cn>）发布”

七、**学费标准**：共计 4 万元，第一学年初缴费 2 万元，第二学年初缴费 2 万元。

复试需要携带的物品

①身份证原件与复印件

②（应届生）学生证/（历届生）本科毕业证和学位证的原件与复印件

③本科学业成绩单原件

P.S 入学后才有体检安排

中科大所有学院。没有政审【政治还不算分】

专业基础测评（上机）

说是上机，其实是上机做选择题【往年的情况】

19 年考察内容为：**C++和数据结构的基础知识及基本运用能力**

仅有通过/不通过两个结果，不计入复试总成绩

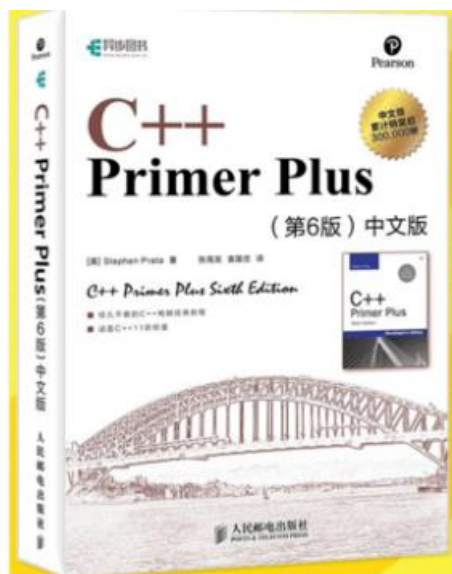
P.S 18 年没有人不通过

题目不会很难，皮皮建议将初试时的数据结构再过一遍

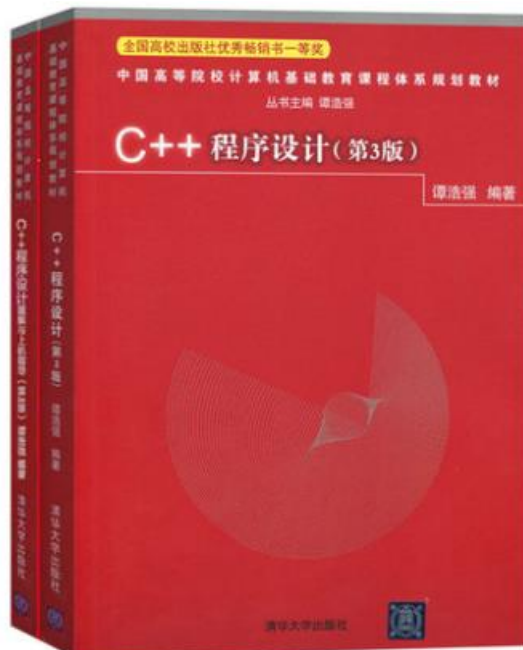
C++可以专门找一本书复习，之后通过题库刷题

C++推荐参考书籍为

大佬版：



菜鸡版：



C++选择题集

1. 下列的各类函数中，不是类的成员函数。(C)
A) 构造函数 B) 析构函数 C) 友元函数 D) 拷贝初始化构造函数
2. 作用域运算符“::”的功能是: (B)
A) 标识作用域的级别的 B) 指出作用域的范围的
C) 给定作用域的大小的 D) 标识成员是属于哪个类的
3. 下列说明中 `const char *ptr`; 其中 `ptr` 应该是: (C)
A) 指向字符常量的指针 B) 指向字符的常量指针
C) 指向字符串常量的指针 D) 指向字符串的常量指针
4. 已知: `print()` 函数是一个类的常成员函数，它无返回值，下列表示中，是正确的。(A)
A) `void print() const;` B) `const void print();`
C) `void const print();` D) `void print(const);`
5. 在类定义的外部，可以被访问的成员有 (C)
A) 所有类成员 B) `private` 的类成员
C) `public` 的类成员 D) `public` 或 `private` 的类成员
6. 要求打开文件“d:\file.dat”，可写入数据，正确的语句是: (B)
`ifstream infile("d:\file.dat", ios::in);`
`ifstream infile("d:\\file.dat", ios::in);`
`ofstream infile("d:\file.dat", ios::out);`
`fstream infile("d:\\file.dat", ios::in|ios::out);`
7. 关于类和对象不正确的说法是: (C)
A) 类是一种类型，它封装了数据和操作 B) 对象是类的实例
C) 一个类的对象只有一个 D) 一个对象必属于某个类
8. 在 C++ 中，类与类之间的继承关系具有 (C)
A) 自反性 B) 对称性 C) 传递性 D) 反对称性
9. 结构化程序设计所规定的三种基本控制结构是: (C)
A) 输入、处理、输出 B) 树形、网形、环形

C)顺序、选择、循环 D)主程序、子程序、函数

10. C++语言是从早期的 C 语言逐渐发展演变而来的. 与 C 语言相比, 它在求解问题方法上进行的最大改进是: (B)

A)面向过程 B)面向对象 C)安全性 D)复用性

11. 在 C++中实现封装是借助于(B)

A)枚举 B)类 C)数组 D)函数

12. C++语言是从早期的 C 语言逐渐发展演变而来的与 C 语言相比, 它在求解问题方法上进行的最大改进是: (B)

A)面向过程 B)面向对象 C)安全性 D)复用性

13. 在 C++中用 (D) 能够实现将参数值带回。

a) 数组 b) 指针 c) 引用 d) 上述 ABC 都可

14. 考虑函数原型 `void test(int a, int b=7, char ch="*")`, 下面的函数调用中, 属于不合法调用的是: (C)

A) `test(5)` B) `test(5, 8)` C) `test(6, "#")` D) `test(0, 0, "*")`;

15. 下列关于类和对象的叙述中, 错误的是: (A)

A) 一个类只能有一个对象; B) 对象是类的具体实例;

C) 类是对某一类对象的抽象; D) 类和对象的关系是一种数据类型与变量的关系。

16. 已知: `int m=10`; 下列表示引用的方法中, 正确的是: (C)

A) `int &Z;` B) `int &t=10;` C) `int &X=m;` D) `float &f=&m;`

17. 对于 `int *pa[5]`; 的描述中, 正确的是: (D)

A) `pa` 是一个指向数组的指针, 所指向的数组是 5 个 `int` 型元素

B) `pa` 是一个指向某数组中第 5 个元素的指针, 该元素是 `int` 型变量

C) `pa[5]` 表示数组的第 5 个元素的值, 是 `int` 型的值

D) `pa` 是一个具有 5 个元素的指针数组, 每个元素是一个 `int` 型指针

18. 定义析构函数时, 应该注意: (C)

A) 其名与类名完全相同 B) 返回类型是 void 类型

C) 无形参, 也不可重载 D) 函数体中必须有 delete 语句

19. 在公有继承的情况下, 基类的成员 (私有的除外) 在派生类中的访问权限
(B)

A) 受限制 B) 保持不变 C) 受保护 D) 不受保护

20. 下列不是函数重载所要求的条件是: (D)

A) 函数名相同 B) 参数个数不同

C) 参数类型不同 D) 函数返回值类型不同

21. 在 C++ 中执行以下 4 条语句后输出 rad 值为: (C)

```
static int hot=200;
```

```
int &rad=hot;
```

```
hot=hot+100;
```

```
cout<<rad<<endl;
```

A) 0 B) 100 C) 300 D) 200

22. 类 class CC1 的说明如下, 错误的语句是: (A)

```
class CC1{
```

```
    int a=2;      // (A)
```

```
    CC1();      // (B)
```

```
public:
```

```
    CC1(int val);      // (C)
```

```
    ~CC1();      // (D)
```

```
};
```

23. 下列各类函数中, 不是类的成员函数是: (C)

A) 构造函数 B) 析构函数

C) 友元函数 D) 拷贝构造函数

24. 类的指针成员的初始化是通过函数完成的, 这个函数通常是: (C)

A) 析构函数 B) 构造函数 C) 其它成员函数 D) 友元函数

25. 关于构造函数的说法, 不正确的是: (A)

A) 没有定义构造函数时, 系统将不会调用它 B) 其名与类名完全相同

C) 它在对象被创建时由系统自动调用 D) 没有返回值

26. 按解释中的要求在下列程序划线处填入的正确语句是: (C)

```
#include <iostream.h>

class Base{
public:
    void fun() {cout<<"Base::fun"<<endl; }
};

class Derived:public Base{
public:
    void fun()
    {_____ //在此空格处调用基类的函数 fun()
    cout<<"Derived::fun"<<endl; }
};
```

A) fun(); B) Base.fun(); C) Base::fun(); D) Base->fun();

27. 在 C++ 中编译器所生成的文件的扩展名为 (B)

A) .cpp B) .obj C) .lib D) .exe

28. 在保护继承的情况下, 基类的成员 (私有的除外) 在派生类中的访问权限

(C)

A) 受限制 B) 保持不变 C) 受保护 D) 不受保护

29. 有如下一段程序:

```
int f1(float);
int f2(char);
int f3(float);
int f4(float);
int (*pf) (float);
```

则以下不合法的是: (C)

A) int (*p) (float) =&f1; B) pf=&f4; C) pf=&f2; D) pf=&f3;

30. 通常拷贝构造函数的参数是: (C)

A) 对象名 B) 对象的成员名 C) 对象的引用名 D) 对象的指针名

31. 系统在调用重载函数时, 不能作为确定哪个重载函数被调用的依据是:

(D)

A) 参数个数 B) 参数类型 C) 函数名称 D) 函数的返回值类型

32. 下列描述中，正确的是：(D)

A) 所有的运算符都可以重载。

B) 基类类型的指针可以指向子类，子类类型的指针也可以指向基类。

C) 可以在类的构造函数中对静态数据成员进行初始化。

D) 动态联编要满足两个条件：被调用的成员函数是虚函数；用指针或引用调用虚函数。

33. 关于成员函数特征的描述中，错误的是：(A)

A) 成员函数一定是内联函数 B) 成员函数可以重载

C) 成员函数可以设置参数的默认值 D) 成员函数可以是静态的

34. 下列关于构造函数的描述中，错误的是：(D)

A) 构造函数可以设置默认参数； B) 构造函数在定义类对象时自动执行

C) 构造函数可以是内联函数； D) 构造函数不可以重载

35. 下面描述中，表达错误的是：(B)

A) 公有继承时，基类中的 public 成员在派生类中仍是 public 的

B) 公有继承时，基类中的 private 成员在派生类中仍是 private 的

C) 公有继承时，基类中的 protected 成员在派生类中仍是 protected 的

D) 私有继承时，基类中的 public 成员在派生类中是 private 的

36. C++语言是从早期的 C 语言逐渐发展演变而来的，与 C 语言相比，C++在求解问题方法上进行的最大改进是：(B)

A) 面向过程 B) 面向对象 C) 安全性 D) 复用性

37. 有如下类声明 “class

A{int x; }; ”，则

A 类的成员 x 是：(B)

A) 公有数据成员 B) 私有数据成员 C) 公有成员函数 D) 私有成员函数

38. 有如下程序：执行后的输出结果应该是：(A)

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

```
class cla{
```

```
    static int n;
```

```
public:
    cla() {n++; }
    ~cla() {n--; }
    static int get_n() {return n; }
};

int cla::n=0;

int main()
{
    cla * p=new cla;
    delete p;
    cout<<"n"<<cla::get_n()<<endl;
    return 0;
}
```

A)n=0 B)n=1 C)n=3 D)n=4

39. 有如下程序: 执行后的输出结果应该是: (A)

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

```
class A{
public:
    A() {cout<<"A"; }
};
```

```
class C:public A{
public:
    C() {cout<<"C"; }
};
```

```
void main() {C cobj; }
```

A) AC B) CA C) A D) C

40. 有如下程序: 执行后的输出结果应该是: (D)

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

```
class BASE{
```

```
public:
    ~BASE() {cout<<"BASE"; }
};
class DERIVED: public BASE{
public:
    ~DERIVED() {cout<<"DERIVED"; }
};
void main() {DERIVED x; }
```

A)BASE B)DERIVED C)BASEDERIVED D)DERIVEDBASE

41. 类 TM 和类 TN 是两个独立的类, 那么类 TM 中哪种形式的数据成员一定是不允许的: (C)

A)TM * a B)TN & a C)TM a D)TN a;

42. 通过下述哪一项调用虚函数时, 采用动态联编。(A)

A)对象指针 B)对象名 C)成员名限定 D)派生类名

43. 在下述哪种情况下适宜采用 inline 定义内联函数。(C)

A)函数体含有循环语句 B)函数体含有递归语句

C)函数代码少、频繁调用 D)函数体含有 switch 语句

44. 创建动态对象, 需要使用操作符 (A)

A) new B) delete C) . D) *

45. 删除动态对象, 需要使用操作符 (B)

A) new B) delete C) . D) *

46. 在 C++ 中不能被重载的一组运算符是: (A)

A):: ?: . sizeof B)?: . sizeof ++ -->

C):: ?: new delete D)++ -- ?: sizeof

47. 如果类 A 被说明成类 B 的友元, 则 (C)

A)类 A 的成员即是类 B 的成员

B)类 B 的成员即是类 A 的成员

C)类 A 的成员函数可以访问类 B 的所有成员

D)类 B 的成员函数可以访问类 A 的所有成员

48. 下面关于友元的描述中, 错误的是: (D)

- A) 友元函数可以直接访问该类的私有成员。
- B) 一个类的友元类中的所有成员函数都是这个类的友元函数。
- C) 利用友元可以提高程序的运行效率, 但却破坏了封装性。
- D) 友元关系不能被继承, 是双向可交换的。

49. 下列虚基类的声明中, 正确的是: (B)

- A) `class virtual B: public A` B) `class B: virtual public A`
- C) `class B: public A virtual` D) `virtual class B: public A`

50. C++设置虚基类的目的是 (A)

- A) 消除二义性 B) 简化程序 C) 提高运行效率 D) 减少目标代码

51. 有如下模板定义:

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

在下列对 fun 的调用中, 错误的是: (C)

- A) `fun(2, 8)` B) `fun(2.0, 8.2)` C) `fun(2.3, 8)` D) `fun<float>(2, 8.3)`

52. 通常情况下, 函数模板中的类型参数个数不能是 (A)

- A) 0 个 B) 1 个 C) 2 个 D) 3 个

53. 以下关于 this 指针的叙述中正确的是: (D)

- A) 任何与类相关的函数都有 this 指针; B) 类的成员函数都有 this 指针;
- C) 类的友元函数都有 this 指针; D) 类的非静态成员函数才有 this 指针;

54. 当一个类的某个函数被说明为 virtual 时, 在该类的所有派生类中的同原型函数 (A)

- A) 都是虚函数 B) 只有被重新说明时才是虚函数
- C) 都不是虚函数 D) 只有被重新说明为 virtual 时才是虚函数

55. 在派生类中重新定义虚函数时, 必须在下列哪方面与基类保持一致。

(A)

- A) 参数类型 B) 参数名字 C) 操作内容 D) 赋值

56. 关于 const 关键字说法错误的是 (D)

- A) const 关键字可以修饰对象和成员函数 B) const 对象不能被修改
- C) const 成员函数不能修改类数据成员 D) const 可以用于说明类

57. 执行如下程序后的输出结果是: (B)

```
#include<iostream.h>

class test{
    static int count;
public:
    test() {count++; }
    ~test() {count--; }

    static int
    getCount() {return
    count; }
};

int test::count=0;

int main()
{
    test * p=new test;
    test * q=new test;
    delete p;
    cout<<"count="<<test::
    getCount()<<endl;
    return 0;
}
```

A) count=0 B) count=1 C) count=2 D) count=3

58. 关于类模板的说法正确的是: (B)

A) 类模板的主要作用是生成抽象类

B) 类模板实例化时, 编译器将根据给出的模板实参生成一个类

C) 在类模板中的数据成员具有同样类型

D) 类模板中的成员函数没有返回值

59. 有如下说明语句 `int a[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; int *p=a;` 则数值为 9 的表达式是: (B)。

A)*p+9 B)*(p+8) C)*p+=9 D)p+7

60. 以下关键字不能用来声明类的访问权限的是: (C)

A)public B)private C)static D)protected

61. 实现运行时的多态性采用(D)

A)重载函数 B)构造函数 C)析构函数 D)虚函数

62. 下列说明语句中正确的是: (D)

A)int a,&ra=a,&&ref=ra; B)int &refa[10];

C)int a,&ra=a,&*refp=&ra; D)int *pi, *&pref=pi;

63. 进行文件操作时需要包含头文件 (C)

A)iostream.h B)stdio.h C)fstream.h D)stdlib.h

64. 使用如 setw() 的格式操纵符时需要包含头文件 (C)

A)iostream.h B)fstream.h C)iomanip.h D)stdlib.h

65. 对 C++ 语言和 C 语言的兼容性, 描述正确的是: (A)

A)C++ 兼容 C B)C++ 部分兼容 C C)C++ 不兼容 C D)C 兼容 C++

66. 在 C++ 中使用流进行输入输出, 其中用于屏幕输出的对象是: (C)

A)cerr B)cin C)cout D)cfile

67. 对使用关键字 new 所开辟的动态存储空间, 释放时必须使用 (C)

A)free B)create C)delete D)realse

68. 如没有使用 private 关键字定义类的数据成员, 则默认为 (A)

A)private B)public C)protected D)friend

69. 使用值传递方式将实参传给形参, 下列说法正确的是: (A)

A)形参是实参的备份 B)实参是形参的备份

C)形参和实参是同一对象 D)形参和实参无联系

70. 在函数调用时, 如某一默认参数要指明一个特定值, 则有 (A)

A)其之前所有参数都必须赋值 B)其之后所有参数都必须赋值

C)其前、后所有参数都必须赋值 D)其前、后所有参数都不必赋值

71. 设存在函数 int max(int, int) 返回两参数中较大值, 若求 22, 59, 70 三者中最大值, 下列表达式不正确的是: (C)

A)int m = max(22, max(59, 70)); B)int m = max(max(22, 59), 70);

C) `int m = max(22, 59, 70);` D) `int m = max(59, max(22, 70));`

72. 下列哪个类型函数不适合声明为内联函数 (A)

A) 函数体语句较多 B) 函数体语句较少 C) 函数执行时间较短 D) 函数执行时间过长

73. `int Func(int, int);` 不可与下列哪个函数构成重载 (B)

A) `int Func(int, int, int);` B) `double Func(int, int);`
C) `double Func(double, double);` D) `double Func(int, double);`

74. 类的私有成员可在何处访问 (D)

A) 通过子类的对象访问 B) 本类及子类的成员函数中
C) 通过该类对象访问 D) 本类的成员函数中

75. 如果没有为一个类定义任何构造函数的情况下, 下列描述正确的是:

(A)

A) 编译器总是自动创建一个不带参数的构造函数 B) 这个类没有构造函数
C) 这个类不需要构造函数 D) 该类不能通过编译

76. 一个类可包含析构函数的个数是: (B)

A) 0 个 B) 1 个 C) 0 个或 1 个 D) 0 个或多个

77. 一个类可包含构造函数的个数是: (D)

A) 0 个 B) 0 个或 1 个 C) 0 个或多个 D) 1 个或多个

78. `this` 指针存在的目的是: (B)

A) 保证基类公有成员在子类中可以被访问
B) 保证每个对象拥有自己的数据成员, 但共享处理这些数据成员的代码
C) 保证基类保护成员在子类中可以被访问
D) 保证基类私有成员在子类中可以被访问

79. 下列关于类的权限的描述错误的是: (A)

A) 类本身的成员函数只能访问自身的私有成员
B) 类的对象只能访问该类的公有成员
C) 普通函数不能直接访问类的公有成员, 必须通过对象访问
D) 一个类可以将另一个类的对象作为成员

80. 在编译指令中, 宏定义使用哪个指令 (B)

A)#include B)#define C)#if D)#else

81. 设类 A 将其它类对象作为成员, 则建立 A 类对象时, 下列描述正确的是:

(B)

A) A 类构造函数先执行 B) 成员构造函数先执行

C) 两者并行执行 D) 不能确定

82. 下列描述错误的是: (A)

A) 在创建对象前, 静态成员不存在 B) 静态成员是类的成员

C) 静态成员不能是虚函数 D) 静态成员函数不能直接访问非静态成员

83. 对于友元描述正确的是: (B)

A) 友元是本类的成员函数 B) 友元不是本类的成员函数

C) 友元不是函数 D) 友元不能访问本类私有成员

84. 在哪种派生方式中, 派生类可以访问基类中的 protected 成员 (B)

A) public 和 private B) public 和 protected

C) protected 和 private D) 仅 protected

85. 在每个 C++ 程序中都必须包含的一个函数是 (B)

A) main() B) MAIN() C) name() D) function()

86. 设 x 和 y 均为 bool 量, 则 x&&y 为真的条件是 (A)

A) 它们均为真 B) 其中一个为真 C) 它们均为假 D) 其中一个为假

87. 下面的哪个保留字不能作为函数的返回类型 (C)

A) void B) int C) new D) long

88. 假定 a 为一个整型数组名, 则元素 a[4] 的字节地址为 (C)

A) a+4 B) a+8 C) a+16 D) a+32

89. 假定有类 AB, 有相应的构造函数定义, 能正确执行 “AB a(4), b(5), c[3], *p[2]={&a, &b};” 语句, 请问执行完此语句后共调用该类构造函数的次数为

(C)

A) 3 B) 4 C) 5 D) 6

C++程序设计模拟试卷(一)

一、单项选择题

1. 编写C++程序一般需经过的几个步骤依次是 ()

- A. 编辑、调试、编译、连接
- B. 编辑、编译、连接、运行
- C. 编译、调试、编辑、连接
- D. 编译、编辑、连接、运行

答案：B

解析：(P21)经过编辑、编译、连接和运行四个步骤。编辑是将C++源程序输入计算机的过程，保存文件名为cpp。编译是使用系统提供的编译器将源程序cpp生成机器语言的过程，目标文件为obj，由于没有得到系统分配的绝对地址，还不能直接运行。连接是将目标文件obj转换为可执行程序的过程，结果为exe。运行是执行exe，在屏幕上显示结果的过程。

2. 决定C++语言中函数的返回值类型的是 ()

- A. return语句中的表达式类型
- B. 调用该函数时系统随机产生的类型
- C. 调用该函数时的主调用函数类型
- D. 在定义该函数时所指定的数据类型

答案：D

解析：(P51)函数的返回值类型由定义函数时的指定的数据类型决定的。A项的表达式值要转换成函数的定义时的返回类型。

3. 下面叙述不正确的是 ()

- 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

答案：D

解析：(P136)继承方式有三种：公有、私有和保护。多继承中，多个基类具有同名成员，在它们的子类中访问这些成员，就产生了二义性，但进行访问时，不能存在二义性。赋值兼容规则是指派生类对象可以当作基类对象使用，只要存在继承关系，所以单继承或多继承都适用。基类中的公有成员采用私有继承时，在派生类中变成了私有成员，所以D项错误。

4. 所谓数据封装就是将一组数据和与这组数据有关操作组装在一起，形成一个实体，这实体也就是 ()

- A. 类
- B. 对象
- C. 函数体
- D. 数据块

答案：A

解析：(P39)类即数据和操作的组合体，数据是类的静态特征，操作是类具有的动作。

5. 在公有派生类的成员函数不能直接访问基类中继承来的某个成员，则该成员一定是基类中

的 ()

- A. 私有成员
- B. 公有成员
- C. 保护成员
- D. 保护成员或私有成员

答案: A

解析: (P133)在派生类中基类的保护或者基类公有都可以直接访问, 基类的私有成员只能是基类的成员函数来访问。所以选择A项。

6. 对基类和派生类的关系描述中, 错误的是 ()

- A. 派生类是基类的具体化
- B. 基类继承了派生类的属性
- C. 派生类是基类定义的延续
- D. 派生类是基类的特殊化

答案: B

解析: (P129)派生类的成员一个是来自基类, 一个来自本身, 所以派生类是基类的扩展, 也是基类的具体化和特殊化, 派生类是对基类扩展。B项基类不能继承派生类成员, 所以错误。

7. 关于this指针使用说法正确的是 ()

- 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。

答案: A

解析: (P86)this指针是隐藏的, 可以使用该指针来访问调用对象中的数据。基类的成员在派生类中能否访问, 与继承方式有关, 与this没有关系。所以选择A项。

8. 所谓多态性是指 ()

- A. 不同的对象调用不同名称的函数
- B. 不同的对象调用相同名称的函数
- C. 一个对象调用不同名称的函数
- D. 一个对象调用不同名称的对象

答案: B

解析: (P167)多态性有两种静态多态性和动态多态性, 静态多态性是指调用同名函数, 由于参数的不同调用不同的同名函数; 动态多态性是指不同对象调用同名函数时, 由于对象不同调用不同的同名函数。多态性肯定具有相同的函数名, 所以选择B项。

9. 一个函数功能不太复杂, 但要求被频繁调用, 则应把它定义为 ()

- A. 内联函数
- B. 重载函数
- C. 递归函数
- D. 嵌套函数

答案: A

解析: (P59)内联函数特征代码少, 频繁调用, 执行效率高。重载函数解决统一接口的问题; 递归是子程序调用, 程序调用要耗费很多空间和时间, 循环/迭代

都比递归有效率得多, 递归只是从形式上, 逻辑比较简洁。嵌套函数即反复调用, 速度较慢。所以选择A项。

10. 下面函数模板定义中不正确的是 ()

- A. A
- B. B
- C. C
- D. D

答案: A

解析: (P147)A项中F是一个返回Q类型的值, 而return中用返回类型作为返回值错误。所以选择A项。

11. 假设ClassY:publicX, 即类Y是类X的派生类, 则说明一个Y类的对象时和删除Y类对象时

, 调用构造函数和析构函数的次序分别为 ()

- A. X, Y; Y, X
- B. X, Y; X, Y
- C. Y, X; X, Y
- D. Y, X; Y, X

答案: A

解析: (P130)派生类构造函数必须对这三类成员进行初始化, 其执行顺序: 调用基类构造函数; 调用子对象的构造函数; 派生类的构造函数体。析构函数在执行过程中也要对基类和成员对象

进行操作, 但它的执行过程与构造函数正好相反, 即对派生类新增普通成员进行清理; 调用成员对象析构函数, 对派生类新增的成员对象进行清理; 调用基类析构函数, 对基类进行清理, 所以选择A项。

12. 适宜采用inline定义函数情况是 ()

- A. 函数体含有循环语句
- B. 函数体含有递归语句
- C. 函数代码少、频繁调用
- D. 函数代码多、不常调用

答案: C

解析: (P59)内联函数具有程序代码少、频繁调用和执行效率高的特征, 所以选择C项。

13. 假定一个类的构造函数为A(int aa, int bb) {a=aa--;b=a*bb;}, 则执行Ax(4, 5); 语句后

, x.a和x.b的值分别为 ()

- A. 3和15
- B. 5和4
- C. 4和20
- D. 20和5

答案: C

解析: (P75)a=4, 因为后减, b的值与a、bb相关, b=4*5=20, 而与aa没有任何关系。

14. 在类中说明的成员可以使用关键字的是 ()

- A. public

- B. extern
- C. cpu
- D. register

答案: A

解析: extern用于声明外部变量的。register声明寄存器类型变量。无cpu类型。它们都不能声明类成员。public声明为公有访问权限, 所以选择A项。

15. 下列不能作为类的成员的是 ()

- A. 自身类对象的指针
- B. 自身类对象
- C. 自身类对象的引用
- D. 另一个类的对象

答案: B

解析: 类的定义, 如果有自身类对象, 使得循环定义, B项错误。在类中具有自身类的指针, 可以实现链表的操作, 当然也可以使用对象的引用。类中可以有另一个类的对象, 即成员对象。所以选择B选项。

16. 使用地址作为实参传给形参, 下列说法正确的是 ()

- A. 实参是形参的备份
- B. 实参与形参无联系
- C. 形参是实参的备份
- D. 实参与形参是同一对象

答案: D

解析: (P51)地址作为实参, 表示实参与形参代表同一个对象。如果实参是数值, 形参也是普通变量, 此时形参是实参的备份。所以选择D项。

17. 下列程序的输出结果是 ()

```
#include <iostream.h>
void main()
{int n [ ] [3] = {10, 20, 30, 40, 50, 60};
int (*p) [3];
p=n;
cout<<p [0] [0] <<"", "<<*(p [0] +1)<<", "<<(*p) [2] <<endl;}
```

- A. 10, 30, 50
- B. 10, 20, 30
- C. 20, 40, 60
- D. 10, 30, 60

答案: B

解析: 如果数组元素都是相同类型的指针, 则称这个数组为指针数组。指针数组一般用于处理二维数组。声明的格式为: <数据类型>(&(*变量名))< [元素个数] >。

p表示指向数组n的行指针。如果将指针的初始化(*p) [3] =b;地址的等价形式:

p+i p [i] *(p+i)都表示b数组第i+1行的第1个元素的首地址。

*(p+i)+j p [i] +j &p [i] [j] 都表示b数组第i+1行、第j+1列元素的地址。值的等价形式:

$*(p+i)+j$ $*(p[i]+j)$ $p[i][j]$ 都表示b数组第i+1、第j+1列元素的值。

所以题目分别访问 $p[0][0]$, $p[0][1]$, $p[0][2]$ 。

18. 在C++中, 使用流进行输入输出, 其中用于屏幕输入 ()

- A. cin
- B. cerr
- C. cout
- D. clog

答案: A

解析: (P193) (1)标准输入流cin: istream类的对象。(2)标准输出流cout: ostream类的对象。(3)非缓冲型标准出错流cerr: ostream类的对象。(4)缓冲型标准出错流clog: ostream类的对象

19. 假定AA为一个类, a() 为该类公有的函数成员, x为该类的一个对象, 则访问x对象中函数

成员a()的格式为 ()

- A. x.a
- B. x.a()
- C. x->a
- D. (*x).a()

答案: B

解析: (P41)对象访问成员的方式为: 对象名.成员。指针可以有两种: (*对象指针).成员或者对象指针->成员。A选项是访问数据成员, B项是访问成员函数。

20. 关于对象概念的描述中, 说法错误的是 ()

- A. 对象就是C语言中的结构变量
- B. 对象代表着正在创建的系统中的实体
- C. 对象是类的一个变量
- D. 对象之间的信息传递是通过消息进行的

答案: A

解析: (P37)A对象在C++中才有, 包括数据和操作两项, 而C中的变量只有数据, 没有操作。所以A项错误。

二、填空题

1. C++的流库预定义了4个流, 它们是cin、cout、clog和___。

答案: (P193)cerr

[解析] cin、cout、clog和cerr分别用于标准输入、输出、标准错误流(缓冲)和标准错误流(非缓冲)。

2. 每个对象都是所属类的一个___。

答案: (P69)实例

[解析] 类是对象的抽象, 对象是类的一个实例。

3. 在已经定义了整型指针ip后, 为了得到一个包括10个整数的数组并由ip所指向, 应使用语句___。

答案: (P78) `int *ip=new int [10];`

[解析] new用来动态开辟空间。常用来产生动态数组及对象构造函数。

4. 函数模板中紧随template之后尖括号内的类型参数都要冠以保留字___。

答案: (P145)class

[解析] 类模板的使用。template <class T>, 也可以引入多参数的如:

template <class

T1, class T2,..., class Tn>

5. 定义类的动态对象数组时, 系统只能够自动调用该类的___构造函数对其进行初始化。

答案: (P80)无参

[解析] 使用new创建对象数组, 调用无参构造函数。

6. 表达式cout<<endl 还可表示为___。

答案: '\n'

[解析] endl与字符常量'\n'等价。

7. 在C++中, 访问一个指针所指向的对象的成员所用的指向运算符是___。

答案: ->

[解析] 指针使用成员有两种方法: “->”指向运算符和“.”成员访问运算符。

8. 假如一个类的名称为MyClass, 使用这个类的一个对象初始化该类的另一个对象时, 可以调

用___构造函数来完成此功能。

答案: (P80)复制或拷贝

复制或拷贝构造函数就是用对象初始化新的对象。

9. 对赋值运算符进行重载时, 应声明为___函数。

答案: (P183)类成员

[解析] 运算符重载的方法有友元或者成员函数两种途径, 但是赋值运算符只能使用成员函数的方法来实现。

10. 如果要把A类成员函数f()且返回值为void声明为类B的友元函数, 则应在类B的定义中加入的语句___。

答案: (P109)friend void A::f();

[解析] 成员函数作为另一个类的友元函数, 格式为: friend 返回类型 类名::函数(形参)。

11. 下列程序段的输出结果是___。

```
for(i=0, j=10, k=0; i<=j; i++, j-=3, k=i+j); cout<<k;
```

答案: 4

[解析] for循环结构, 三个表达式的作用, 初始化、循环判断条件和循环变量变化。循环执行了三次, k的作用是计算i、j的和。

12. String 类的___方法返回查找到的字符串在主串的位置。

答案: (P40)find

[解析] string类对象方法的find, 查不到字符串, 则返回-1。

13. int n=0;

while (n=1) n++;

while循环执行次数是___。

答案: 无限次

[解析] =是赋值运算符, 不是关系运算符, 且不等0, 所以死循环。

14. 控制格式输入输出的操作中, 函数___是用来设置填充字符。要求给出函数

名和参数类型

答案: (P195) setfill(char)

[解析] 格式控制方法的使用, 如setw, setfill等等。

15. C++语言支持的两种多态性分别是编译时的多态性和____的多态性。

答案: (P167) 运行时

[解析] 多态性包括静态的(编译时)多态性和动态的(运行时)多态性。

16. 设函数sum是由函数模板实现的, 并且sum(3, 6)和sum(4. 6, 8)都是正确的函数调用, 则函数模板具有____个类型参数。

答案: (P61) 2

17. 执行下列代码

```
string str("HelloC++");
```

```
cout<<str.substr(5, 3);
```

程序的输出结果是____。

答案: (P42) C++

[解析] substr取子字符串, 第1个参数表示要截取子串在字符串中的位置, 第2个表示取多少个字符。

18. 在面向对象的程序设计中, 将一组对象的共同特性抽象出来形成____。

答案: (P38) 类

[解析] 类是相似特征的对象抽象, 对象是类的一个实例。

19. 定义类动态对象数组时, 元素只能靠自动调用该类的____来进行初始化。

答案: (P77) 无参构造函数

[解析] 使用new 创建动态对象数组, 不能有参数, 所以只能调用无参的构造函数, 初始化对象

20. 已知有20个元素int类型向量V1, 若用V1初始化为V2向量, 语句是____。

答案: (P151) vector <int>V2(V1);

[解析] 采用向量初始化另一个向量的形式: vector <type> name1(name);

一、单项选择题

1. 静态成员函数没有 ()

- A. 返回值
- B. this指针
- C. 指针参数
- D. 返回类型

答案: B

解析: (P107) 静态成员函数是普通的函数前加入static, 它具有函数的所有的特征: 返回类型、形参, 所以使用(P107)静态成员函数, 指针可以作为形参, 也具有返回值。静态成员是类具有的属性, 不是对象的特征, 而this表示的是隐藏的对象指针, 因此静态成员函数没有this 指针。静态成员函数当在类外定义时, 要注意不能使用static关键字作为前缀。由于静态成员函数在类中只有一个拷贝(副本), 因此它访问对象的成员时要受到一些限制: 静态成员函数可以直接

访问类中说明的静态成员, 但不能直接访问类中说明的非静态成员; 若要访问非静态成员时, 必须通过参数传递的方式得到相应的对象, 再通过对象来访问

问。

2. 假定AB为一个类, 则执行 “AB a(2), b [3] , *p [4] ;” 语句时调用该类构造函数的次数

为 ()

- A. 3
- B. 4
- C. 5
- D. 9

答案: B

解析: (P79) a(2) 调用1次带参数的构造函数, b [3] 调用3次无参数的构造函数, 指针没有给它分配空间, 没有调用构造函数。所以共调用构造函数的次数为4。

3. 有关多态性说法不正确的是 ()

- A. C++语言的多态性分为编译时的多态性和运行时的多态性
- B. 编译时的多态性可通过函数重载实现
- C. 运行时的多态性可通过模板和虚函数实现
- D. 实现运行时多态性的机制称为动态多态性

答案: C

解析: (P171) 多态性分为静态的和动态的。静态通过函数的重载来实现, 动态是通过基类指针或基类引用和虚函数来实现的。所以错误的是C项。

4. 假定一个类的构造函数为 “A(int i=4, int j=0) {a=i;b=j;}”, 则执行 “A x (1);” 语

句后, x. a和x. b的值分别为 ()

- A. 1和0
- B. 1和4
- C. 4和0
- D. 4和1

答案: A

解析: (P75) 带默认的构造函数, 对应实参没有值时就采用形参值。调用构造函数时, i=1, 不采用默认值, 而只有一个参数, j采用默认值0即j=0, 因此a=1, b=0, 选择A项。

5. 类MyA的拷贝初始化构造函数是 ()

- A. MyA()
- B. MyA(MyA*)
- C. MyA(MyA&)
- D. MyA(MyA)

答案: C

解析: (P80) 复制即拷贝构造函数使用对象的引用作形参, 防止临时产生一个对象, A无参构造函数, B是指针作为形参, D项是对象, 所以选择C项。

6. 在C++中, 函数原型不能标识 ()

- A. 函数的返回类型
- B. 函数参数的个数
- C. 函数参数类型
- D. 函数的功能

答案: D

解析: 函数的声明, 说明函数的参数、返回类型以及函数名, 函数体即实现部分决定功能。所以函数的原型不能决定函数的功能。

7. 友元关系不能 ()

- A. 提高程序的运行效率
- B. 是类与类的关系
- C. 是一个类的成员函数与另一个类的关系
- D. 继承

答案: D

解析: (P111)友元可以是函数与类的关系即友元函数, 也可以类与类的关系即友元类, 但友元不能继承, 是单向性, 且不具有传递性。友元可以访问类中所有成员, 提高了访问的方便性。因此选择D项。

8. 实现两个相同类型数加法的函数模板的声明是 ()

- A. `add(T x, T y)`
- B. `T add(x, y)`
- C. `T add(T x, y)`
- D. `T add(T x, T y)`

答案: D

解析: (P63)实现两个相同类型数加法结果应该和操作数具有相同类型。进行加法运算后结果也是和参数具有相同类型, 需要返回值。A无返回值时要用void, B形参无类型, C形参y没有类型, 所以选择D项。

9. 在`int a=3, int *p=&a;`中, `*p`的值是 ()

- A. 变量a的地址值
- B. 无意义
- C. 变量p的地址值
- D. 3

答案: D

解析: `*p`代表引用a变量的值, `p`代表a的地址值。所以选择D项。

10. 下列不是描述类的成员函数的是 ()

- A. 构造函数
- B. 析构函数
- C. 友元函数
- D. 拷贝构造函数

答案: C

解析: (P109)构造函数、析构函数、拷贝构造函数都是特殊的成员函数, 友元则不是成员函数。

所以选择C项。

11. 如果从原有类定义新类可以实现的是 ()

- A. 信息隐藏
- B. 数据封装
- C. 继承机制
- D. 数据抽象

答案: C

解析: (P129)继承指在原有类的基础上产生新类。数据封装即数据和操作组合

在一起, 形成类。信息的隐藏, 通过访问权限来实现。数据抽象, 将事物的特征抽象为数据成员或服务。因此选择C项。

12. 下面有关类说法不正确的是 ()

- A. 一个类可以有多个构造函数
- B. 一个类只有一个析构函数
- C. 析构函数需要指定参数
- D. 在一个类中可以说明具有类类型的数据成员

答案: C

解析: (P80)构造函数可以有参数、可以重载、因此可以有多个, A项正确。析构函数只有一个不能重载、不能继承, 没有返回值, B项正确, C项错误。

13. 在函数定义中的形参属于 ()

- A. 全局变量
- B. 局部变量
- C. 静态变量
- D. 寄存器变量

答案: B

解析: 形参或函数中定义的变量都是局部变量。在函数外定义的变量是全局变量。形参只能用局部变量, 频繁使用的变量可以声明为寄存器变量, 形参不能使用静态变量或寄存器变量。

14. 下列有关重载函数的说法中正确的是 ()

- A. 重载函数必须具有不同的返回值类型
- B. 重载函数参数个数必须相同
- 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

答案: C

解析: (P59)函数的重载必须函数名相同而形参类型或个数不同, 与返回值无关。

15. this指针存在的目的是 ()

- A. 保证基类私有成员在子类中可以被访问
- B. 保证基类保护成员在子类中可以被访问
- C. 保证每个对象拥有自己的数据成员, 但共享处理这些数据成员的代码
- D. 保证基类公有成员在子类中可以被访问

答案: C

解析: (P86)C++要求函数在被调用之前, 应当让编译器知道该函数的原型, 以便编译器利用函数原型提供的信息去检查调用的合法性, 强制参数转换成为适当类型, 保证参数的正确传递。对于标准库函数, 其声明在头文件中, 可以用#include宏命令包含这些原型文件; 对于用户自定义函数, 先定义、后调用的函数可以不用声明, 但后定义、先调用的函数必须声明。一般为增加程序的可理解性, 常将主函数放在程序开头, 这样需要在主函数前对其所调用的函数一一进行声明, 以消除函数所在位置的影响。所以选择C项。

16. 关于new运算符的下列描述中, 错误的是 ()

- A. 它可以用来动态创建对象和对象数组
- B. 使用它创建的对象或对象数组可以使用运算符delete删除
- C. 使用它创建对象时要调用构造函数

D. 使用它创建对象数组时必须指定初始值

答案: D

解析: (P78)new创建的对象数组不能指定初始值, 所以调用无参的构造函数, 选择D项。

17. 已知: p是一个指向类A数据成员m的指针, A1是类A的一个对象。如果要给m赋值为5, 正确的是 ()

A. A1.p=5;

B. A1->p=5;

C. A1.*p=5;

D. *A1.p=5;

答案: C

解析: (P118)A中p是指针即地址, 错误; B选项中A1不是指针不能使用指向运算符->, 错误

; “*”比“.”级别要高, 所以D选项*A1.p=5相当于(*A1).p=5; 错误。另外涉及到指向成员函数

时注意以下几点:

指向成员函数的指针必须于其赋值的函数类型匹配的三个方面: (1) 参数类型和个数; (2) 返回类型; (3) 它所属的类类型。

成员函数指针的声明: 指向short型的Screen类的成员的指针定义如下:

```
short Screen::* ps_Screen;
```

ps_Screen可以用_height的地址初始化如下: short

```
Screen::*ps_Screen=&Screen::_height;
```

类成员的指针必须总是通过特定的对象或指向改类型的对象的指针来访问。是通过使用两个指向成员操作符的指针(针对类对象和引用的.*, 以及针对指向类对象的指针的->*)。

18. 以下基类中的成员函数表示纯虚函数的是 ()

A. virtual void tt()=0

B. void tt(int)=0

C. virtual void tt(int)

D. virtual void tt(int) {}

答案: A

解析: (P173)当在基类中不能为虚函数给出一个有意义的实现时, 可以将其声明为纯虚函数, 实现由派生类完成。格式: virtual<函数返回类型说明符><函数名>(<参数表>)=0;。

19. C++类体系中, 不能被派生类继承的有 ()

A. 常成员函数

B. 构造函数

C. 虚函数

D. 静态成员函数

答案: B

解析: (P132)构造函数不能被继承。

20. 静态成员函数不能说明为 ()

- A. 整型函数
- B. 浮点函数
- C. 虚函数
- D. 字符型函数

答案: C

解析: (P108)使用关键字static声明的成员函数就是静态成员函数, 静态成员函数也属于整个类而不属于类中的某个对象, 它是该类的所有对象共享的成员函数。

静态成员函数可以在类体内定义, 也可以在类外定义。当在类外定义时, 要注意不能使用static关键字作为前缀。

由于静态成员函数在类中只有一个拷贝(副本), 因此它访问对象的成员时要受到一些限制: 静态成员函数可以直接访问类中说明的静态成员, 但不能直接访问类中说明的非静态成员; 若要访问非静态成员时, 必须通过参数传递的方式得到相应的对象, 再通过对象来访问。虚函数是非静态的、非内联的成员函数。静态成员函数不能被说明为虚函数。

二、填空题

1. 假设int a=1,b=2;则表达式(++a/b)*b--的值为___。

答案: 2

[解析] 前缀++或--表示先使变量值变化, 再使用, 这和后缀恰恰相反。但是编译cout<<(++a/b)*b--时, 先++a/b值为1, 后1*b--, 先取b=2, 结果为2, 再让b=1。

2. 抽象类中至少要有一个___函数。

答案: (P173)纯虚

[解析] 至少有一个纯虚函数的类就称为抽象类, 即不能实例化。

3. 一个抽象类的派生类可以实例化的必要条件是实现了所有的___。

答案: (P173)纯虚函数的定义

[解析] 抽象类只因有纯虚函数, 所以不能被实例化, 所以派生类要实例化必须对纯虚函数进行定义。

4. 下面程序的输出结果为___。

```
#include <iostream.h>
void main()
{int num=2,i=6;
do
{i--;
num++;
}while(--i);
cout<<num<<endl;
}
```

答案: 5

[解析] do-while循环, 前缀先使i减少1后判断是否为零, 不为零时再次执行循环, 为零退出循环。循环值执行3次就退出, 所以结果为5。

5. 静态成员函数、友元函数、构造函数和析构函数中, 不属于成员函数的是___。

答案: (P109)友元函数

〔解析〕友元函数不是类成员，但可以访问类成员。类的封装性保证了数据的安全，但引入友元，虽然访问类是方便了，但确实破坏类访问的安全性。

6. 在用C++进行程序设计时，最好用___代替malloc。

答案：(P10) new

〔解析〕new与delete是C++语言特有的运算符，用于动态分配和释放内存。new用于为各种数据类型分配内存，并把分配到的内存首地址赋给相应的指针。new的功能类似于malloc()函数。

使用new的格式为：

<指针变量>new<数据类型>;

其中，<数据类型>可以是基本数据类型，也可以是由基本类型派生出来的类型；<指针变量>取得分配到的内存首地址。new有3种使用形式。

(1) 给单个对象申请分配内存

```
int *ip; ip=new int; //ip指向1个未初始化的int型对象
```

(2) 给单个对象申请分配内存的同时初始化该对象

```
int *ip; ip=new int(68); //ip指向1个表示为68的int型对象
```

(3) 同时给多个对象申请分配内存

```
int *ip; ip=new int [5]; //ip指向5个未初始化的int型对象的首地址
```

```
for(int i=0; i<5; i++) ip[i] =5*i+1; //给ip指向的5个对象赋值
```

用new申请分配内存时，不一定能申请成功。若申请失败，则返回NULL，即空指针。因此，在程序中可以通过判断new的返回值是否为0来获知系统中是否有足够的空间供用户使用。

7. 由const修饰的对象称为___。

答案：(P113) 常对象

〔解析〕使用const关键字说明的成员函数称为常成员函数，使用const关键字说明的对象称为常对象。

常成员函数的说明格式如下：<返回类型说明符><成员函数名>(<参数表>)const;

常成员函数不更新对象的数据成员，也不能调用该类中没有用const修饰的成员函数。常对象只能调用它的常成员函数，而不能调用其他成员函数。const关键字可以用于参与重载函数的区分。

8. 在C++程序设计中，建立继承关系倒挂的树应使用___继承。

答案：(P138) 单

〔解析〕一个基类可以派生多个子类，一个子类可以再派生出多个子类，这样就形成了一个倒立的树。

9. 基类的公有成员在派生类中的访问权限由___决定。

答案：(P132) 访问控制方式或继承方式

10. 不同对象可以调用相同名称的函数，但执行完全不同行为的现象称为___。

答案：(P167) 多态性

〔解析〕多态性的概念。虚函数是实现多态的基础，运行过程中的多态需要同时满足3个条件：(1)类之间应满足子类型关系。(2)必须要有声明的虚函数。

(3)调用虚函数操作的是指向对象的指针或者对象引用；或者是由成员函数调用虚函数（如果是在构造函数或析构函数中调用虚函数，则采用静态联编）。

11. this指针始终指向调用成员函数的___。

答案：对象

this指针是隐藏的指针，它指向调用函数的对象。

12. 预处理命令以__符号开头。

答案: (P183)operator

[解析] 文件包含、预处理和编译都是以#开头。

13. 类模板用来表达具有__的模板类对象集。

答案: (P145)相同处理方法

[解析] 模板特点是不同的数据具有相同的处理方法的抽象。

14. C++程序的源文件扩展名为__。

答案: (P21)cpp

[解析] 源程序*.cpp, 目标文件为*.obj, 可执行程序*.exe。

15. 在#include命令中所包含的头文件，可以是系统定义的头文件，也可以是__的头文件。

答案: (P7)自定义

[解析] #include装入文件有两种方式<>和“”，一是系统的，一是自定义文件。

16. vector类中向向量尾部插入一个对象的方法是__。

答案: (P157)push_back

17. C++语言中如果调用函数时，需要改变实参或者返回多个值，应该采取__方式。

答案: (P51)传地址或引用

[解析] 传地址即指针，在函数中通过指针修改它指向的变量的值时，实参也就变化了。使用引用，直接修改变量的别名即引用的值，该变量也就随着变化。

18. 语句序列

```
ifstream infile;
```

```
infile.open(“data.dat”);
```

的功能可用一个语句实现，这个语句是__。

答案: (P199)ifstream infile(“data.dat”);

[解析] void ifstream::open(const char *fname, int mode=ios::in, int access=filebuf::openprot);

```
ifstream::ifstream(const char *fname, int mode=ios::in, int
access=filebuf::openprot);
```

其中，第一个参数是用来传递文件名的；第二个参数mode的值决定文件将如何被打开；第三个参数access的值决定文件的访问方式，一般取缺省值filebuf::openprot，表示是普通文件。

mode的取值如下：(1)ios::in：打开一个文件进行读操作，而且该文件必须已经存在；(2)ios::nocreate：不建立新的文件。当文件不存在时，导致open()失败

；(3)ios::noreplace：不修改原来已经存在的文件。若文件已经存在，导致open()失败；(4)ios::binary：文件以二进制方式打开，缺省时为文本文件。

19. 如果要类B的成员函数void fun()说明为类A的友元函数，则应在类A中加入语句__。

答案: (P111)friend void B::fun();

[解析] 声明成员函数作为另外一个类的友元函数时，使用类作用域运算

符: : 。

20. 在编译指令中, 宏定义使用___指令。

答案: (P6、97)#define

[解析] 静态成员是所有对象共享的特征, 也就是类的特征。

C++程序设计模拟试卷(三)

一、单项选择题

1. 设有定义int i;double j=5;; 则10+i+j值的数据类型是 ()

- A. int
- B. double
- C. float
- D. 不确定

答案: B

解析: 考察数据的转换, j是double类型, 运算只能作同类型的运算, 所以要转换, 而int能自动转换为double类型, 所以结果是double类型。

2. 要禁止修改指针p本身, 又要禁止修改p所指向的数据, 这样的指针应定义为 ()

- A. const char *p= "ABCD" ;
- B. char *const p= "ABCD" ;
- C. char const *p= "ABCD" ;
- D. const char * const p= "ABCD" ;

答案: D

解析: (P12)const char *p说明禁止通过p修改所指向的数据。char * const p则说明不能修改指针p的地址。因此const char * const p= "ABCD" ; 它禁止修改指针p本身, 又禁止修改p所指向的数据。

3. 类的构造函数被自动调用执行的情况是在定义该类的 ()

- A. 成员函数时
- B. 数据成员时
- C. 对象时
- D. 友元函数时

答案: C

解析: (P75)建立对象时, 自动构造函数的初始化对象, 是系统自动调用的。而成员函数、友元函数, 需要用户直接调用, 因此选择C项。

4. 已知类A是类B的友元, 类B是类C的友元, 则 ()

- A. 类A一定是类C的友元
- B. 类C一定是类A的友元
- C. 类C的成员函数可以访问类B的对象的所有成员
- D. 类A的成员函数可以访问类B的对象的所有成员

答案: C

解析: (P105)友元说明方法如下:

friend?<类名>;//友元类类名

使用友元可以访问所有成员:

(1)友元关系不能被继承。

(2)友元关系是单向的,不具有交换性。所以,B项和D项错误。

(3)友元关系不具有传递性。所以,A项错误。

5. 假定一个类的构造函数为“`A(int i=4, int j=0) {a=i;b=j;}`”,则执行“`A x (1);`”语

句后,x.a和x.b的值分别为 ()

A. 1和0

B. 1和4

C. 4和0

D. 4和1

答案: A

解析: (P75)带默认的构造函数,对应实参没有值时就采用形参值。调用构造函数时,i=1,不采用默认值,而只有一个参数,j采用默认值0即j=0,因此a=1,b=0,选择A项。

6. 关于this指针使用说法正确的是 ()

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。

答案: A

解析: (P86)this指针是隐藏的,可以使用该指针来访问调用对象中的数据。基类的成员在派生类中能否访问,与继承方式有关,与this没有关系。所以选择A项。

7. 所谓多态性是指 ()

A. 不同的对象调用不同名称的函数

B. 不同的对象调用相同名称的函数

C. 一个对象调用不同名称的函数

D. 一个对象调用不同名称的对象

答案: B

解析: (P167)多态性有两种静态多态性和动态多态性,静态多态性是指调用同名函数,由于参数的不同调用不同的同名函数;动态多态性是指不同对象调用同名函数时,由于对象不同调用不同的同名函数。多态性肯定具有相同的函数名,所以选择B项。

8. 友元关系不能 ()

A. 提高程序的运行效率

B. 是类与类的关系

C. 是一个类的成员函数与另一个类的关系

D. 继承

答案: D

解析: (P111)友元可以是函数与类的关系即友元函数,也可以类与类的关系即友元类,但友元不能继承,是单向性,且不具有传递性。友元可以访问类中所有成员,提高了访问的方便性。因此选择D项。

9. 语句`ofstream f(“TEMP.DAT”,ios::app | ios::binary)?`的功能是建立流对象f,试图打

开文件TEMP.DAT 并与之连接,并且 ()

- A. 若文件存在, 将文件写指针定位于文件尾; 若文件不存在, 建立一个新文件
- B. 若文件存在, 将其置为空文件; 若文件不存在, 打开失败
- C. 若文件存在, 将文件写指针定位于文件首; 若文件不存在, 建立一个新文件
- D. 若文件存在, 打开失败; 若文件不存在, 建立一个新文件

答案: A

解析: (P199) `ios::binary`, 采用二进制形式, `ios::app`定位到文件尾部。

10. 构造函数不具备的特征是 ()

- A. 构造函数的函数名与类名相同
- B. 构造函数可以重载
- C. 构造函数可以设置默认参数
- D. 构造函数必须指定类型说明

答案: D

解析: (P75) 构造函数无返回类型不能继承但可以重载, 所以选择D项。

11. 在公有继承的情况下, 基类的公有或保护成员在派生类中的访问权限 ()

- A. 受限制
- B. 保持不变
- C. 受保护
- D. 不受保护

答案: B

解析: (P132) 继承方式的不同派生类成员的权限也不同, 采用公有继承, 除了私有无法访问外, 公有、保护在派生类中保持不变, 所以选择B项。

12. 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(4, 5)`; 语句后, `x.a`和`x.b`的值分别为 ()

- A. 3和15
- B. 5和4
- C. 4和20
- D. 20和5

答案: C

解析: (P75) `a=4`, 因为后减, `b`的值与`a`、`bb`相关, `b=4*5=20`, 而与`aa`没有任何关系。

13. C++对C语言做了很多改进, 即从面向过程变成为面向对象的主要原因是 ()

- A. 增加了一些新的运算符
- B. 允许函数重载, 并允许设置缺省参数
- C. 规定函数说明符必须用原型
- D. 引进了类和对象的概念

答案: D

解析: (P29) C++是一面向对象的语言, 面向对象的特征, 抽象、多态、继承和封装。

14. 在类中说明的成员可以使用关键字的是 ()

- A. `public`
- B. `extern`
- C. `cpu`
- D. `register`

答案: A

解析: extern用于声明外部变量的。register声明寄存器类型变量。无cpu类型。它们都不能声明类成员。public声明为公有访问权限,所以选择A项。

15. C++语言中所有在函数中定义的变量,连同形式参数,都属于 ()

- A. 全局变量
- B. 局部变量
- C. 静态变量
- D. 函数

答案: B

解析: 变量存储类可分为两类: 全局变量和局部变量。

(1) 全局变量: 在函数外部定义的变量称为全局变量,其作用域为: 从定义变量的位置开始到源程序结束。使用全局变量降低了程序的可理解性,软件工程学提倡尽量避免使用全局变量。

(2) 局部变量: 在函数内部定义的变量称为局部变量,其作用域为: 从定义变量的位置开始到函数结束。局部变量包含自动变量(auto)静态变量

(static)以及函数参数。形参不能是静态的。所以选择B项。

16. 在私有继承的情况下,基类成员在派生类中的访问权限 ()

- A. 受限制
- B. 保持不变
- C. 受保护
- D. 不受保护

答案: A

解析: (P132)私有继承下,基类中的公有或保护成员在派生类中也是私有的,所以选择A选项。

17. 使用地址作为实参传给形参,下列说法正确的是 ()

- A. 实参是形参的备份
- B. 实参与形参无联系
- C. 形参是实参的备份
- D. 实参与形参是同一对象

答案: D

解析: (P51)地址作为实参,表示实参与形参代表同一个对象。如果实参是数值,形参也是普通变量,此时形参是实参的备份。所以选择D项。

18. C++的继承性允许派生类继承基类的 ()

- A. 部分特性,并允许增加新的特性或重定义基类的特性
- B. 部分特性,但不允许增加新的特性或重定义基类的特性
- C. 所有特性,并允许增加新的特性或重定义基类的特性
- D. 所有特性,但不允许增加新的特性或重定义基类的特性

答案: A

解析: (P129)派生类有两类成员: 一是基类,二是自身类。派生类中的成员不能访问基类中的私有成员,可以访问基类中的公有成员和保护成员。

19. 对于int *pa [5];的描述,正确的是 ()

- A. pa是一个指向数组的指针,所指向的数组是5个int型元素
- B. pa是一个指向某个数组中第5个元素的指针,该元素是int型变量
- C. pa [5]表示某个数组的第5个元素的值

D. pa是一个具有5个元素的指针数组，每个元素是一个int型指针

答案: D

解析: (P117)指针数组: 数组元素都是相同类型的指针，相同类型的指针是指针所指向的对象类型是相同的。例如，语句 `int *pa[5]` ;定义了一个指针数组。在指针数组的定义中有两个运算符: *和[]，运算符[]的优先级高于*，所以 `*pa[5]` 等价于 `*(pa[5])`，`pa[5]` 表示一个数组，而*表示后面的对象为指针变量，合在一起 `*pa[5]` 表示一个指针数组。该数组包含5个元素，每个元素都是指向int型的指针。所以选择D选项。

20. 以下基类中的成员函数表示纯虚函数的是 ()

A. `virtual void tt()=0`

B. `void tt(int)=0`

C. `virtual void tt(int)`

D. `virtual void tt(int){}`

答案: A

解析: (P173)当在基类中不能为虚函数给出一个有意义的实现时，可以将其声明为纯虚函数，实现由派生类完成。格式: `virtual<函数返回类型说明符><函数名>(<参数表>)=0;`。

二、填空题

1. 单目运算符作为类成员函数重载时，形参个数为__个。

答案: (P189)0

[解析] 单目运算符使用成员函数重载可以不用形参，双目运算符使用一个参数。

2. 抽象类中至少要有一个__函数。

答案: (P173)纯虚

[解析] 至少有一个纯虚函数的类就称为抽象类，即不能实例化。

3. 设类A有成员函数 `void f(void)`；若要定义一个指向类成员函数的指针变量pf来指向f，该指针变量的声明语句是: __。

答案: (P117)`void (A::*pf)(void)=&A::f;`

[解析] `void(A::*pf)(void)=&A::f;`指向成员函数的指针，它相当于两条语句: `void(A::*pf)(void);`和 `pf=&A::f;`。

4. 执行下列程序

```
double a=3.1415926,b=3.14;
```

```
cout<<setprecision(5)<<a<<"", "<<setprecision(5)<<b<<endl;
```

程序的输出结果是__。

答案: 3.1416, 3.14

[解析] 题目设置精度即有效数字都是5，a四舍五入是3.1416，b是3.14。

5. vector类中用于删除向量中的所有对象的方法是__。

答案: (P151)`clear()`

[解析] 向量的使用。返回向量中对象的方法有: `front()``back()` `operator []`，在向量中删除对象的方法 `pop_back` `erase` `clear`。

6. 重载的运算符保持其原有的__、优先级和结合性不变。

答案: (P183)操作数

[解析] 运算符重载时要遵循以下规则:

(1)除了类属关系运算符“.”、成员指针运算符“.”、“*”、作用域运算符

“::”、sizeof运算符和三目运算符“?:”以外，C++中的所有运算符都可以重载。

(2) 重载运算符限制在C++语言中已有的运算符范围内的允许重载的运算符之中，不能创建新的运算符。

(3) 重载之后的运算符不能改变运算符的优先级和结合性，也不能改变运算符操作数的个数及语法结构。

7. 编译时的多态性通过___函数实现。

答案: (P165) 重载

〔解析〕编译多态性，实现的方法主要通过函数的重载或运算符的重载。

8. 基类的公有成员在派生类中的访问权限由___决定。

答案: (P132) 访问控制方式或继承方式

9. 假设类X的对象x是类Y的成员对象，则“Y Obj”语句执行时，先调用类___的构造函数。

答案: (P130) X

〔解析〕派生类中的构造函数的执行顺序，先基类后派生类。

10. 下列程序段的输出结果是___。

```
cout.setf(ios::showpos);
cout<<509.3<<endl;
```

答案: (P193) +509.3

〔解析〕输入、输出格式ios::showpos用于输出数据的符号位。

11. 下列程序段的输出结果是___。

```
for(i=0, j=10, k=0; i<=j; i++, j-=3, k=i+j); cout<<k;
```

答案: 4

〔解析〕for循环结构，三个表达式的作用，初始化、循环判断条件和循环变量变化。循环执行了三次，k的作用是计算i、j的和。

12. C++中ostream的直接基类___。

答案: (P193) ios

〔解析〕istream和ostream的直接基类是ios。

13. int n=0;

```
while (n=1) n++;
```

while循环执行次数是___。

答案: 无限次

〔解析〕=是赋值运算符，不是关系运算符，且不等0，所以死循环。

14. C++中有两种继承：单继承和___。

答案: (P138) 多继承

〔解析〕单继承和多继承，多继承即有多个基类。

15. 在C++中，利用向量类模板定义一个具有10个int的向量A，其元素均被置为1，实现此操作的语句是___。

答案: (P151) vector<int>A(10, 1)

〔解析〕定义向量列表vector<int>A(10, 1)，使用两个参数，10表示长度，1表示数值。

16. vector类中向向量尾部插入一个对象的方法是___。

答案: (P157) push_back

17. C++语言中如果调用函数时，需要改变实参或者返回多个值，应该采取___

方式。

答案: (P51)传地址或引用

[解析] 传地址即指针, 在函数中通过指针修改它指向的变量的值时, 实参也就变化了。使用引用, 直接修改变量的别名即引用的值, 该变量也就随着变化。

18. 若函数的定义处于调用它的函数之前, 则在程序开始可以省去该函数的____语句。

答案: 声明

[解析] 函数使用有两部分: 声明和定义。定义在前, 可以无声明; 但函数定义在后, 调用在前的话, 需要先声明函数的原型。

19. 在C++中有两种参数传递方式: 传值和____。

答案: (P51)传引用

[解析] (1) 传值调用又分为数据传值调用和地址传值调用。(2) 引用调用是将实参变量值传递给形参, 而形参是实参变量的引用名。引用是给一个已有变量起的别名, 对引用的操作就是对该引用变量的操作。

20. 将指向对象的引用作为函数的形参, 形参是对象的引用, 实参是 ____。

答案: (P53)对象名

[解析] 实参与形参类型要一致, 形参是对象的引用, 实参应该是对象名。

C++程序设计模拟试卷(四)

一、单项选择题

1. 当一个类的某个函数被说明为virtual时, 该函数在该类的所有派生类中()

- A. 都是虚函数
- B. 只有被重新说明时才是虚函数
- C. 只有被重新说明为virtual时才是虚函数
- D. 都不是虚函数

答案: A

解析: (P170)在基类声明为virtual的函数为虚函数, 在派生类中只要有相同的函数(函数名相同、返回值相同、形参类型和个数相同)即使不用virtual说明, 也都是虚函数。

2. 要禁止修改指针p本身, 又要禁止修改p所指向的数据, 这样的指针应定义为()

- A. `const char *p= "ABCD" ;`
- B. `char *const p= "ABCD" ;`
- C. `char const *p= "ABCD" ;`
- D. `const char * const p= "ABCD" ;`

答案: D

解析: (P12)`const char *p`说明禁止通过p修改所指向的数据。`char * const p`则说明不能修改指针p的地址。因此`const char * const p= "ABCD" ;`它禁止修改指针p本身, 又禁止修改p所指向的数据。

3. 函数调用`func((exp1, exp2), (exp3, exp4, exp5))`中所含实参的个数为()

- A. 1

- B. 2
- C. 4
- D. 5

答案: B

解析: (exp1, exp2)、(exp3, exp4, exp5)表示是两个逗号表达式, 值是最后一个值, 相当于两个参数。因此实参的个数是2。

4. 设有函数模板

```
template <class Q>
Q Sum(Q x, Q y)
{return (x)+(y);}
```

则下列语句中对该函数模板错误的使用是 ()

- A. Sum(10, 2);
- B. Sum(5.0, 6.7);
- C. Sum(15.2f, 16.0f);
- D. Sum(“AB”, “CD”);

答案: D

解析: (P40)由Q Sum(Q x, Q y)可知形参和函数返回值都是同一种数据类型。

A、B、C三项都正确。而D项用字符串作为实参, 字符串的操作与数值类型不同, 要用特殊方法进行字符串的连接和运算。

5. 类B是类A的公有派生类, 类A和类B中都定义了虚函数func(), p是一个指向类A对象的指针

, 则p->A::func()将 ()

- A. 调用类A中的函数func()
- B. 调用类B中的函数func()
- C. 根据p所指的物体类型而确定调用类A中或类B中的函数func()
- D. 既调用类A中函数, 也调用类B中的函数

答案: A

解析: (P117)指向类成员指针的使用, A::func()是明确调用A类的func函数, 所以不管p指向基类或者派生类对象, 都执行基类虚函数。注意p->A::func()和p->fun();进行区分。如果使用p->fun(), 因为p指向派生类对象, 由动态多态性可知要调用派生类的虚函数。

6. 在面向对象的程序设计中, 首先在问题域中识别出若干个 ()

- A. 函数
- B. 类
- C. 文件
- D. 过程

答案: B

解析: (P31)面向过程的和面向对象都具有、函数、文件和过程这些概念, 而面向对象程序才有类和对象的特征。所以选择B。

7. 已知f1和f2是同一类的两个成员函数, 但f1不能直接调用f2, 这说明 ()

- A. f1和f2都是静态函数
- B. f1不是静态函数, f2是静态函数
- C. f1是静态函数, f2不是静态函数
- D. f1和f2都不是静态函数

答案: C

解析: (P107)普通成员函数可以调用静态函数, 相反静态函数不能调用普通成员函数, 这与普通函数与常成员函数相同。因此选择C项。

8. 下列有关模板和继承的叙述正确的是 ()

- A. 模板和继承都可以派生出一个类系
- B. 从类系的成员看, 模板类系的成员比继承类系的成员较为稳定
- C. 从动态性能看, 继承类系比模板类系具有更多的动态特性
- D. 相同类模板的不同实例一般没有联系, 而派生类各种类之间有兄弟父子等关系

答案: D

解析: (P145)类是相同类型事物的抽象, 具有不同的操作。而模板是不同类型的事物, 具体相同的操作的抽象。类模板的实例化后, 各个对象没有任何关系。而类对象是通过派生、继承等关系的关系。

9. 有关C++编译指令, 以下叙述正确的是 ()

- A. C++每行可以写多条编译指令
- B. #include指令中的文件名可含有路径信息
- C. C++的编译指令可以以#或//开始
- D. C++中不管# if后的常量表达式是否为真, 该部分都需要编译

答案: B

解析: (P96)编译指令以#作为开头, 只能一行写一条, #if有选择进行编译, 所以选择B项。

10. 在C++中不返回任何类型的函数应该说明为 ()

- A. int
- B. char
- C. void
- D. double

答案: C

解析: 无形参或无返回值都可以用void来声明, int char double分别是整型、字符型和实型。

11. 若Sample类中的一个成员函数说明如下:

void set(Sample& a), 则Sample& a的含义是 ()

- A. 指向类Sample的名为a的指针
- B. a是类Sample的对象引用, 用来作函数Set () 的形参
- C. 将a的地址赋给变量Set
- D. 变量Sample与a按位与的结果作为函数Set的参数

答案: B

解析: (P53)成员函数使用对象的引用作为形参。该函数的功能是将已知对象的所有数据成员的值拷贝给相应对象的所有数据成员, 不会建立临时对象, 这里是对象的引用所以选择B。

12. 下列关于静态数据成员的描述中正确的是 ()

- A. 静态数据成员是类的所有对象所共有的
- B. 静态数据成员要在构造函数内初始化
- C. 类的每个对象有自己的静态数据成员
- D. 静态数据成员不能通过类的对象调用

答案: D

解析: (P107)静态成员属于类的即所有对象所共享的, 只能在外部进行初始化。使用时可以使用形式有两种, 类名::静态成员或者对象.静态成员。所以选择D项。

13. 在编译指令中, 宏定义使用哪个指令 ()

- A. #if
- B. #include
- C. #define
- D. #error

答案: C

解析: (P7)#if条件编译, #include文件包含, #error错误处理。

14. 类的析构函数是对一个对象进行以下哪种操作时自动调用的是 ()

- A. 建立
- B. 撤销
- C. 赋值
- D. 引用

答案: B

解析: (P80)删除对象或结束程序时, 自动调用析构函数。

15. 关于new运算符的下列描述中, 错误的是 ()

- A. 它可以用来动态创建对象和对象数组
- B. 使用它创建的对象或对象数组可以使用运算符delete删除
- C. 使用它创建对象时要调用构造函数
- D. 使用它创建对象数组时必须指定初始值

答案: D

解析: (P107)静态成员的特性是静态成员只有一个拷贝(副本), 这个副本被所有属于这个类的对象共享。这种共享与全局变量或全局函数相比, 既没有破坏数据隐藏的原则, 又保证了安全性。静态成员表示整个类范围的信息, 其声明以关键字static开始, 包括静态数据成员和静态成员函数。在对静态数据成员初始化时应注意:

(1)应在类体外对静态数据成员进行初始化(静态数据成员的初始化与它的访问控制权限无关)。

(2)静态数据成员初始化时前面不加static关键字, 以免与一般静态变量或对象混淆。

(3)由于静态数据成员是类的成员, 因此在初始化时必须使用作用域运算符(::)限定它所属的类。因此选择D项。

16. 如果类A被说明成类B的友元, 则 ()

- A. 类B不一定是类A的友元
- B. 类B的成员即类A的成员
- C. 类A的成员即类B的成员
- D. 类A的成员函数不得访问类B的成员

答案: A

解析: (P113)友元关系不能被继承, 友元关系是单向的, 友元关系不具有传递性。但是友元函数不是类的成员, 所以选择A项。

17. 假定一个类的构造函数为A(int aa, int bb) {a=aa++;b=a*++bb;}，则执行Ax(4, 5);语句

后，x.a和x.b的值分别为（）

- A. 4和5
- B. 4和20
- C. 4和24
- D. 20和5

答案: C

解析: (P75)执行构造函数将数据成员进行赋值，aa++是后加，先赋值a=4，++bb, bb变量值先自加为6，再与a相乘，所以b=24。

18. 下列运算符中，在C++语言中不能重载的是（）

- A. *
- B. >=
- C. ::
- D. /

答案: C

解析: (P186)除了类属关系运算符"."、成员指针运算符".*"、作用域运算符 "::"、sizeof运算符和三目运算符"?:"以外，C++中的所有运算符都可以重载。

19. C++语言建立类族是通过（）

- A. 类的嵌套
- B. 类的继承
- C. 虚函数
- D. 抽象类

答案: B

解析: (P174)类族即同一个类派生出来的类，各个类是兄弟或父子关系。

20. 在C++语言中，数据封装要解决的问题是（）

- A. 数据的规范化
- B. 便于数据转换
- C. 避免数据丢失
- D. 防止不同模块之间数据的非法访问

答案: D

解析: (P39)封装是指把对象属性和操作结合在一起，构成独立的单元，它的内部信息对外界是隐蔽的，不允许外界直接存取对象的属性，只能通过有限的接口与对象发生联系。类是数据封装的工具，对象是封装的实现。类的访问控制机制体现在类的成员中可以有公有成员、私有成员和保护成员。对于外界而言，只需要知道对象所表现的外部行为，而不必了解内部实现细节。封装体现了面向对象方法的“信息隐蔽和局部化原则”。

二、填空题

1. 若要使用string类，需要引入的头文件是_____。

答案: (P40)string.h

〔解析〕编译时要将头文件引入才能使用标准库中的方法或成员。

2. 在函数前面用_____保留字修饰时，则表示该函数表为内联函数。

答案: (P59)inline

〔解析〕内联函数，用来提高程序运行速度。在类内部定义的函数也是内联函

数。

3. 向量操作方法中___方法返回向量中的第一个对象。

答案: (P151)front

[解析] 考察向量方法的使用。front():返回向量中的第1个对象。back():返回向量中的最后一个对象。operator[](size_type,n):返回向量中的第n+1个对象(下标为n的向量元素)。

4. C++派生类使用两种基本的面向对象技术: 第一种称为性质约束, 即对基类的性质加以限制; 第二种称为___, 即增加派生类的性质。

答案: (P129)性质扩展

[解析] 派生类通过继承可以从基类中获得成员, 也可以自定义成员。

5. 重载的运算符保持其原有的___、优先级和结合性不变。

答案: (P183)操作数

[解析] 运算符重载时要遵循以下规则:

(1)除了类属关系运算符“.”、成员指针运算符“.*”、作用域运算符“::”、sizeof运算符和三目运算符“?:”以外, C++中的所有运算符都可以重载。

(2)重载运算符限制在C++语言中已有的运算符范围内的允许重载的运算符之中, 不能创建新的运算符。

(3)重载之后的运算符不能改变运算符的优先级和结合性, 也不能改变运算符操作数的个数及语法结构。

6. 编译时的多态性通过___函数实现。

答案: (P165)重载

[解析] 编译多态性, 实现的方法主要通过函数的重载或运算符的重载。

7. 预处理语句有三种, 分别是宏定义、文件包含和___。

答案: (P7)条件编译

[解析] 宏定义#define, 文件包含#include和条件编译#if等。

8. 构造函数、析构函数和友元函数中, 不是该类成员的是___。

答案: (P109)友元函数

[解析] 友元函数不是类成员, 但可以访问类中成员。

9. 控制格式输入输出的操作中, 函数___是设置域宽的。要求给出函数名和参数类型)。

答案: (P193)setw(int)

[解析] setw(int n):用来设置n输出宽度。

10. 派生类的成员一般分为两部分, 一部分是___, 另一部分是自己定义的新成员。

答案: (P127)从基类继承的成员

[解析] 派生类成员一个来自继承基类成员, 一个来自本身增加的成员。

11. C++中ostream的直接基类___。

答案: (P193)ios

[解析] istream和ostream的直接基类是ios。

12. vector的___方法返回向量中的最后一个对象。

答案: (P151)back

[解析] front():返回向量中的第1个对象。back():返回向量中的最后一个对象。

operator [] (size_type, n):返回向量中的第n+1个对象（下标为n的向量元素）。

13. 执行下列代码

```
int i=230;
cout <<"i="<<hex <<i<<endl;
```

程序的输出结果为___。

答案: (P193)i=e6

【解析】流类库中使用格式符，输出十六进制数据。

14. 在C++中有两种参数传递方式即值传递和___传递。

答案: (P51)引用

【解析】函数参数传递有传值和传引用两种。

15. 使用new为int数组动态分配10个存储空间是___。

答案: (P10)new int [10];

【解析】new delete动态开辟空间和删除空间。new int [10]，注意不要写成new int (10)，使用小括号只能开辟一个空间，使用10来初始化该值。

16. 面向对象的四个基本特性是多态性、继承性、和封装性___。

答案: (P37)抽象

【解析】考察面向对象的四个特征。程序由一组抽象的对象组成，一组对象的共同特征抽象出类的概念，类是对象的抽象，对象是类的实例。封装即将数据和操作紧密结合提供访问的接口，外部通过接口实现访问数据，提供安全性。继承继承解决了类的扩展性。多态性不同对象调用相同的函数名，但调用不同的函数，实现不同的功能，解决了接口统一的问题。

17. 定义虚函数所用的关键字是___。

答案: (P170)virtual

【解析】在成员函数前加virtual修饰的函数就是虚函数。但不是所有成员函数都可以定义为虚函数的。比如构造函数，不能定义虚函数。

18. 执行下列代码

```
cout<<"oct:"<<oct<<34;
```

程序的输出结果是___。

答案: (P193)Oct:42

【解析】oct表示八进制，hex表示十六进制，但它们只能输出整型的数据。

19. 在C++中要创建一个文件输入流对象fin，同时该对象打开文件

“Test.txt”用于输入，则正确的声明语句是___。

答案: (P200)ifstream fin(“Test.txt”);

【解析】文件操作中ifstream用于文件的输入，可以调用它的构造函数与要打开的文件进行关联

20. 如果一个派生类只有一个唯一的基类，则这样的继承关系称为___。

答案: (P130)单一

【解析】根据派生类所拥有的基类数目不同，可以分为单继承和多继承。一个类只有一个直接基类时，称为单继承；而一个类同时有多个直接基类时，则称为多继承。基类与派生类之间的关系如下：

(1) 基类是对派生类的抽象，派生类是对基类的具体化，是基类定义的延续。

(2) 派生类是基类的组合。多继承可以看作是多个单继承的简单组合。

(3) 公有派生类的对象可以作为基类的对象处理。

C++程序设计模拟试卷(五)

一、单项选择题

1. 静态成员函数没有 ()

- A. 返回值
- B. this指针
- C. 指针参数
- D. 返回类型

答案：B

解析：(P107)静态成员函数是普通的函数前加入static,它具有函数的所有的特征：返回类型、形参，所以使用(P107)静态成员函数，指针可以作为形参，也具有返回值。静态成员是类具有的属性，不是对象的特征，而this表示的是隐藏的对象指针，因此静态成员函数没有this 指针。静态成员函数当在类外定义时，要注意不能使用static关键字作为前缀。由于静态成员函数在类中只有一个拷贝（副本），因此它访问对象的成员时要受到一些限制：静态成员函数可以直接

访问类中说明的静态成员，但不能直接访问类中说明的非静态成员；若要访问非静态成员时，必须通过参数传递的方式得到相应的对象，再通过对象来访问。

2. 在类的定义中，用于为对象分配内存空间，对类的数据成员进行初始化并执行其他内部管理操作的函数是 ()

- A. 友元函数
- B. 虚函数
- C. 构造函数
- D. 析构函数

答案：C

解析：(P75)定义构造函数作用就是初始化对象，而析构函数释放对象空间。虚函数用于完成多态性，友元增加访问方便性。

3. 所有在函数中定义的变量，都是 ()

- A. 全局变量
- B. 局部变量
- C. 静态变量
- D. 寄存器变量

答案：B

解析：(P136)变量存储类可分为两类：全局变量和局部变量。

(1) 全局变量：在函数外部定义的变量称为全局变量，其作用域为：从定义变量的位置开始到源程序结束。全局变量增加了函数之间数据联系的渠道，全局变量作用域内的函数，均可使用、修改该全局变量的值，但是使用全局变量降低了程序的可理解性，软件工程学提倡尽量避免使用全局变量。

(2) 局部变量：在函数内部定义的变量称为局部变量，其作用域为：从定义变量的位置开始到函数结束。局部变量包含自动变量(auto)静态变量

(static)以及函数参数。auto变量意味着变量的存储空间的分配与释放是自动进行的。说明符auto可以省略。函数中的局部变量存放在栈空间。在函数开

始运行时，局部变量被分配内存单元，函数结束时，局部变量释放内存单元。因此，任两个函数中的局部变量可以同名，因其占有不同的内存单元而不影响使用。这有利于实现软件开发的模块化。static变量是定义在函数体内的变量，存放在静态存储区，不用栈空间存储，其值并不随存储空间的释放而消失。

4. 假定AB为一个类，则执行“AB a(2), b [3], *p [4];”语句时调用该类构造函数的次数

为 ()

- A. 3
- B. 4
- C. 5
- D. 9

答案: B

解析: (P79) a(2)调用1次带参数的构造函数，b [3]调用3次无参数的构造函数，指针没有给它分配空间，没有调用构造函数。所以共调用构造函数的次数为4。

5. 如果表达式++a中的“++”是作为成员函数重载的运算符，若采用运算符函数调用格式，则可表示为 ()

- A. a.operator++(1)
- B. operator++(a)
- C. operator++(a, 1)
- D. a.operator++()

答案: D

解析: (P186)运算符的重载，前缀先让变量变化。调用++a，等价于a.operator++(), 注意无参的形式。后缀的话a++, 等价于a.operator(0), 带形参，形参名可省。

6. 已知f1和f2是同一类的两个成员函数，但f1不能直接调用f2，这说明 ()

- A. f1和f2都是静态函数
- B. f1不是静态函数，f2是静态函数
- C. f1是静态函数，f2不是静态函数
- D. f1和f2都不是静态函数

答案: C

解析: (P107)普通成员函数可以调用静态函数，相反静态函数不能调用普通成员函数，这与普通函数与常成员函数相同。因此选择C项。

7. 一个函数功能不太复杂，但要求被频繁调用，则应把它定义为 ()

- A. 内联函数
- B. 重载函数
- C. 递归函数
- D. 嵌套函数

答案: A

解析: (P59)内联函数特征代码少，频繁调用，执行效率高。重载函数解决统一接口的问题；递归是子程序调用，程序调用要耗费很多空间和时间，循环/迭代都比递归有效率得多，递归只是从形式上，逻辑比较简洁。嵌套函数即反复调用，速度较慢。所以选择A项。

8. 解决定义二义性问题的方法有（）

- A. 只能使用作用域分辨运算符
- B. 使用作用域分辨运算符或成员名限定
- C. 使用作用域分辨运算符或虚基类
- D. 使用成员名限定或赋值兼容规则

答案：B

解析：(P139)解决二义性问题主要要两种方法：(1)赋值兼容规则；(2)虚基类。

9. 在main函数中可以用p. a的形式访问派生类对象p的基类成员a，其中a是（）

- A. 私有继承的公有成员
- B. 公有继承的私有成员
- C. 公有继承的保护成员
- D. 公有继承的公有成员

答案：D

解析：(P132)公有成员可以在类外访问，保护类型成员可以在派生类中访问，但不能在类外访问，在main函数中访问，说明a是公有成员。只有公有继承时a才能是公有的，所以D项正确。

10. 在C++中不返回任何类型的函数应该说明为（）

- A. int
- B. char
- C. void
- D. double

答案：C

解析：无形参或无返回值都可以用void来声明，int char double分别是整型、字符型和实型。

11. 若Sample类中的一个成员函数说明如下：

void set(Sample& a)，则Sample& a的含义是（）

- A. 指向类Sample的名为a的指针
- B. a是类Sample的对象引用，用来作函数Set（）的形参
- C. 将a的地址赋给变量Set
- D. 变量Sample与a按位与的结果作为函数Set的参数

答案：B

解析：(P53)成员函数使用对象的引用作为形参。该函数的功能是将已知对象的所有数据成员的值拷贝给相应对象的所有数据成员，不会建立临时对象，这里是对象的引用所以选择B。

12. 要实现动态联编必须（）

- A. 通过成员名限定来调用虚函数
- B. 通过对象名来调用虚函数
- C. 通过派生类对象来调用虚函数
- D. 通过对象指针或引用来调用虚函数

答案：D

解析：(P170)通过基类指针或基类引用来调用虚函数实现动态多态性，静态多态性通过重载来实现的。所以选择D项。

13. 在派生类中定义虚函数时，可以与基类中相应的虚函数不同的是（）

- A. 参数类型
- B. 参数个数
- C. 函数名称
- D. 函数体

答案: D

解析: (P170)虚函数在基类和派生类, 具有相同的返回类型、形参类型和形参个数, 而函数体可以根据不同的派生类或基类实现不同的操作, 即不同函数体。

14. 实现两个相同类型数加法的函数模板的声明是 ()

- A. `add(T x, T y)`
- B. `T add(x, y)`
- C. `T add(T x, y)`
- D. `T add(T x, T y)`

答案: D

解析: (P63)实现两个相同类型数加法结果应该和操作数具有相同类型。进行加法运算后结果也是和参数具有相同类型, 需要返回值。A无返回值时要用void, B形参无类型, C形参y没有类型, 所以选择D项。

15. 下列不是描述类的成员函数的是 ()

- A. 构造函数
- B. 析构函数
- C. 友元函数
- D. 拷贝构造函数

答案: C

解析: (P109)友元函数虽然不是成员函数但是可以访问类所有成员。构造函数、析构函数和拷贝构造函数(复制构造函数)都是类的特殊函数用于对象的创建和撤销, 所以选择C项。

16. 继承机制的作用是 ()

- A. 信息隐藏
- B. 数据封装
- C. 定义新类
- D. 数据抽象

答案: C

解析: (P40)面向对象设计中的类的特点: 抽象、封装、继承和多态等, 继承用于对类的扩展, 所以选择C项。

17. 已知: p是一个指向类A数据成员m的指针, A1是类A的一个对象。如果要给m赋值为5, 正确的是 ()

- A. `A1.p=5;`
- B. `A1->p=5;`
- C. `A1.*p=5;`
- D. `*A1.p=5;`

答案: C

解析: (P118)A中p是指针即地址, 错误; B选项中A1不是指针不能使用指向运算符->, 错误; “*”比“.”级别要高, 所以D选项`*A1.p=5`相当于`(*A1).p=5`; 错误。另外涉及到指向成员函数时注意以下几点:

指向成员函数的指针必须于其赋值的函数类型匹配的三个方面: (1) 参数类型和个数; (2) 返回类型; (3) 它所属的类类型。

成员函数指针的声明: 指向short型的Screen类的成员的指针定义如下:

```
short Screen::* ps_Screen;
```

ps_Screen可以用_height的地址初始化如下: short

```
Screen::*ps_Screen=&Screen::_height;
```

类成员的指针必须总是通过特定的对象或指向改类型的对象的指针来访问。是通过使用两个指向成员操作符的指针(针对类对象和引用的*, 以及针对指向类对象的指针的->*)。

18. 如果采用动态多态性, 要调用虚函数的是 ()

- A. 基类对象指针
- B. 对象名
- C. 基类对象
- D. 派生类名

答案: A

解析: (P171)基类指针或者基类的引用调用虚函数都会产生动态多态性

19. 若有以下定义, 则说法错误的是 ()

```
int a=100,*p=&a;
```

- A. 声明变量p, 其中*表示p是一个指针变量
- B. 变量p经初始化, 获得变量a的地址
- C. 变量p只可以指向一个整型变量
- D. 变量p的值为100

答案: D

解析: 指针变量如同其他变量一样, 在使用之前必须先声明。声明指针变量的格式为:

```
<类型名>*<变量名>;
```

其中, <类型名>是指针变量所指向对象的类型, 它可以是C++语言预定义的类型, 也可以是用户自定义类型。<变量名>是用户自定义的标识符。符号*表示<变量>是指针变量。而不是普通变量。 *表示指针, p是变量, p指向一个整型的变量, 值为a的地址值, *p=100。

20. C++语言建立类族是通过 ()

- A. 类的嵌套
- B. 类的继承
- C. 虚函数
- D. 抽象类

答案: B

解析: (P174)类族即同一个类派生出来的类, 各个类是兄弟或父子关系。

二、填空题

1. 假设int a=1,b=2;则表达式(++a/b)*b--的值为___。

答案: 2

[解析] 前缀++或--表示先使变量值变化, 再使用, 这和后缀恰恰相反。但是编译

cout<<(++a/b)*b--时, 先++a/b值为1, 后1*b--, 先取b=2, 结果为2, 再让b=1。

2. 复制构造函数使用___作为形式参数。

答案: (P80)对象的引用

[解析] 复制构造函数使用对象的引用来初始化一个新对象, 避免临时产生对象。

3. 通过C++语言中的___机制, 可以从现存类中构建其子类。

答案: (P127)继承

[解析] 继承概念, 从现有的类生成新类, 原有的类称为父类或基类, 新类又称子类或派生类或衍生类, 它是对基类的扩充。

4. 静态成员函数、友元函数、构造函数和析构函数中, 不属于成员函数的是___。

答案: (P109)友元函数

[解析] 友元函数不是类成员, 但可以访问类成员。类的封装性保证了数据的安全, 但引入友元, 虽然访问类是方便了, 但确实破坏类访问的安全性。

5. 在下面的类定义中, 私有成员有___。

```
class Location
{
    int X, Y;
protected:
    int zeroX, zeroY;
    int SetZero(int zeroX, int zeroY);
private:
    int length, height;
public:
    void init(int initX, int initY);
    int GetX();
    int GetY();
};
```

答案: (P69)X, Y, length, height

6. 在C++程序设计中, 建立继承关系倒挂的树应使用___继承。

答案: (P138)单

[解析] 一个基类可以派生多个子类, 一个子类可以再派生出多个子类, 这样就形成了一个倒立的树。

7. C++支持的两种多态性分别是___多态性和运行多态性。

答案: (P165)静态或编译

[解析] 多态性包括静态(编译时)的和动态(运行时)的动态性。

8. C++中语句const char * const p= "hello"; 所定义的指针p和它所指的内容都不能被___。

答案: (P12)修改

[解析] 使用const修改的内容不能修改, 这里同时修饰地址和值, 表示地址和值都不变。

9. 在C++中, 定义虚函数的关键字是___。

答案: (P170)virtual

[解析] 在普通函数前面用virtual修饰的函数, 就称为虚函数。

10. 采用私有派生方式, 基类的public成员在私有派生类中是___成员。

答案: (P132) 私有

11. 对赋值运算符进行重载时, 应声明为___函数。

答案: (P183) 类成员

[解析] 运算符重载的方法有友元或者成员函数两种途径, 但是赋值运算符只能使用成员函数的方法来实现。

12. 在C++中有两种参数传递方式即值传递和___传递。

答案: (P51) 引用

[解析] 函数参数传递有传值和传引用两种。

13. 预处理命令以___符号开头。

答案: (P183) operator

[解析] 文件包含、预处理和编译都是以#开头。

14. 在构造函数和析构函数中调用虚函数时采用___。

答案: (P167) 静态联编

[解析] 在析构或构造函数调用虚函数发生静态多态性。

15. C++是通过引用运算符___来定义一个引用的。

答案: (P10) &

[解析] 引用是C不具有使用方法, 它表示变量的别名, 在函数中使用很频繁, 因为调用形式同传值调用, 但修改形参实参也会相应改变的特征。

16. 如果要把类B的成员函数void fun()说明为类A的友元函数, 则应在类A中加入语句___。

答案: (P111) friend void B::fun();

[解析] 声明成员函数作为另外一个类的友元函数时, 使用类作用域运算符: :。

17. 如果要把PI声明为值为3.14159类型为双精度实数的符号常量, 该声明语句是___。

答案: (P6) const double PI(3.14159);或者const double PI=3.14159;

[解析] 使用const声明符号常量, 常量和常量值可以用括号也可以赋值号。

18. 在C++四个流对象中, ___用于标准屏幕输出。

答案: (P194) cout

[解析] cin、cout、cerr和clog中cin用于输入, cout用于输出, cerr、clog错误处理。

19. 执行下列代码

```
int a=32;
double c=32;
cout.setf(ios::hex);
cout<<"hex:a"<<a<<",c"<<c<<endl;
cout.unsetf(ios::hex);
```

程序的输出结果为___。

答案: (P193) hex:a=20, c=32

[解析] 用十六进制只能输出整型数据, 而不能将其它类型数据转换成十六进制的数据输出。所以double类型不变仍然是32(double类型)。

20. 已知有20个元素int类型向量V1, 若用V1初始化为V2向量, 语句是___。

答案: (P151) vector<int>V2(V1);

[解析] 采用向量初始化另一个向量的形式: `vector <type> name1(name);`

C++程序设计模拟试卷(六)

一、单项选择题

1. 当一个类的某个函数被说明为virtual时, 该函数在该类的所有派生类中 ()

- A. 都是虚函数
- B. 只有被重新说明时才是虚函数
- C. 只有被重新说明为virtual时才是虚函数
- D. 都不是虚函数

答案: A

解析: (P170)在基类声明为virtual的函数为虚函数, 在派生类中只要有相同的函数(函数名相同、返回值相同、形参类型和个数相同)即使不用virtual说明, 也都是虚函数。

2. 以下基类中的成员函数表示纯虚函数的是 ()

- A. `virtual void vf(int)`
- B. `void vf(int)=0`
- C. `virtual void vf()=0`
- D. `virtual void yf(int) {}`

答案: C

解析: (P173)纯虚函数是特殊的虚函数, 没有函数体, 形式为: `virtual`返回类型函数名(形参列表)=0; 因此选C项。

3. 下面对静态数据成员的描述中, 正确的是 ()

- A. 静态数据成员可以在类体内进行初始化
- B. 静态数据成员可以直接用类名或者对象名来调用
- C. 静态数据成员不能用private控制符修饰
- D. 静态数据成员不可以被类的对象调用

答案: B

解析: (P107)静态成员可用类名或者对象名来调用, 静态数据成员必须在类外进行初始化。静态成员可以用public、private和protected修饰。所以选B项。

4. 所谓数据封装就是将一组数据和与这组数据有关操作组装在一起, 形成一个实体, 这实体

也就是 ()

- A. 类
- B. 对象
- C. 函数体
- D. 数据块

答案: A

解析: (P39)类即数据和操作的组合体, 数据是类的静态特征, 操作是类具有的动作。

5. 类B是类A的公有派生类, 类A和类B中都定义了虚函数func(), p是一个指向类A对象的指针

, 则`p->A::func()`将 ()

- A. 调用类A中的函数func()
- B. 调用类B中的函数func()
- C. 根据p所指向的对象类型而确定调用类A中或类B中的函数func()
- D. 既调用类A中函数, 也调用类B中的函数

答案: A

解析: (P117)指向类成员指针的使用, `A::func()` 是明确调用A类的func函数, 所以不管p指向基类或者派生类对象, 都执行基类虚函数。注意`p->A::func()`和`p->fun()`;进行区分。如果使用`p->fun()`, 因为p指向派生类对象, 由动态多态性可知要调用派生类的虚函数。

6. 在面向对象的程序设计中, 首先在问题域中识别出若干个 ()

- A. 函数
- B. 类
- C. 文件
- D. 过程

答案: B

解析: (P31)面向过程的和面向对象都具有、函数、文件和过程这些概念, 而面向对象程序才有类和对象的特征。所以选择B。

7. 在下列成对的表达式中, 运算结果类型相同的一对是 ()

- A. $7.0 / 2.0$ 和 $7.0 / 2$
- B. $5 / 2.0$ 和 $5 / 2$
- C. $7.0 / 2$ 和 $7 / 2$
- D. $8 / 2$ 和 $6.0 / 2.0$

答案: A

解析: 小数默认的类型为double类型, 整数除以整数结果是取整的结果。A、B、C和D项的第一项分别是double、double、double和int类型的, 第二项分别是double、int、int和double类型, 所以选择A项。

8. 下列不具有访问权限属性的是 ()

- A. 非类成员
- B. 类成员
- C. 数据成员
- D. 函数成员

答案: A

解析: 类成员包括成员函数和数据成员, 都可以使用访问权限public、private和protected来修饰, 而普通的变量不能使用访问权限来说明。

9. 以下有关继承的叙述正确的是 ()

- A. 构造函数和析构函数都能被继承
- B. 派生类是基类的组合
- C. 派生类对象除了能访问自己的成员以外, 不能访问基类中的所有成员
- D. 基类的公有成员一定能被派生类的对象访问

答案: C

解析: (P129)构造函数和析构函数不能被派生类继承, A项错误。派生类是基类的扩展, B项错。

派生类可以访问基类公有和保护类型的成员, 不能访问基类私有成员。D项基类

是公有的成员，若采用私有继承，派生类对象不能直接访问。选择C项。

10. 下列有关模板和继承的叙述正确的是 ()

- A. 模板和继承都可以派生出一个类系
- B. 从类系的成员看，模板类系的成员比继承类系的成员较为稳定
- C. 从动态性能看，继承类系比模板类系具有更多的动态特性
- D. 相同类模板的不同实例一般没有联系，而派生类各种类之间有兄弟父子等关系

答案: D

解析: (P145)类是相同类型事物的抽象，具有不同的操作。而模板是不同类型的事物，具体相同的操作的抽象。类模板的实例化后，各个对象没有任何关系。而类对象是通过派生、继承等关系的关系。

11. 适宜采用inline定义函数情况是 ()

- A. 函数体含有循环语句
- B. 函数体含有递归语句
- C. 函数代码少、频繁调用
- D. 函数代码多、不常调用

答案: C

解析: (P59)内联函数具有程序代码少、频繁调用和执行效率高的特征，所以选择C项。

12. 要采用动态多态性，说法正确的是 ()

- A. 基类指针调用虚函数
- B. 派生类对象调用虚函数
- C. 基类对象调用虚函数
- D. 派生类指针调用虚函数

答案: A

解析: (P170)使用基类的指针或引用，由指向或引用的对象来决定调用不同类的虚函数。所以选择A。

13. C++类体系中，不能被派生类继承的有 ()

- A. 转换函数
- B. 构造函数
- C. 虚函数
- D. 静态成员函数

答案: B

解析: (P127)构造函数不能被继承，而转换函数、虚函数和静态成员函数都可以被继承，所以选择B项。

14. 下列不是描述类的成员函数的是 ()

- A. 构造函数
- B. 析构函数
- C. 友元函数
- D. 拷贝构造函数

答案: C

解析: (P109)构造函数、析构函数、拷贝构造函数都是特殊的成员函数，友元则不是成员函数。

所以选择C项。

15. 下列不能作为类的成员的是 ()

- A. 自身类对象的指针
- B. 自身类对象
- C. 自身类对象的引用
- D. 另一个类的对象

答案: B

解析: 类的定义, 如果有自身类对象, 使得循环定义, B项错误。在类中具有自身类的指针, 可以实现链表的操作, 当然也可以使用对象的引用。类中可以有另一个类的对象, 即成员对象。所以选择B选项。

16. 下列不是描述类的成员函数的是 ()

- A. 构造函数
- B. 析构函数
- C. 友元函数
- D. 拷贝构造函数

答案: C

解析: (P113)友元函数不是类成员, 只是它可以访问类中的成员。

17. 关于对象概念的描述中, 说法错误的是 ()

- A. 对象就是C语言中的结构变量
- B. 对象代表着正在创建的系统中的一个实体
- C. 对象是类的一个变量
- D. 对象之间的信息传递是通过消息进行的

答案: A

解析: (P37)A对象在C++中才有, 包括数据和操作两项, 而C中的变量只有数据, 没有操作。所以A项错误。

18. 派生类的构造函数的成员初始化列表中, 不能包含 ()

- A. 基类的构造函数
- B. 基类的对象初始化
- C. 派生类对象的初始化
- D. 派生类中一般数据成员的初始化

答案: C

解析: (P130)派生类的构造函数的成员初始化, 包括基类成员、基类对象成员和派生类成员。因为本身就是初始化定义的对象, 在构造函数中再进行该类对象的初始化产生了循环定义, 或者类中不能包括本身类的成员对象。所以选择C项。

19. 关于new运算符的下列描述中, 错误的是 ()

- A. 它可以用来动态创建对象和对象数组
- B. 使用它创建的对象或对象数组可以使用运算符delete删除
- C. 使用它创建对象时要调用构造函数
- D. 使用它创建对象数组时必须指定初始值

答案: D

解析: (P78)new创建的对象数组不能指定初始值, 所以调用无参的构造函数, 选择D项。

20. 假定一个类的构造函数为A(int aa, int bb) {a=aa++;b=a*++bb;} , 则执行Ax(4, 5);语句

后, x.a和x.b的值分别为 ()

- A. 4和5
- B. 4和20
- C. 4和24
- D. 20和5

答案: C

解析: (P75) 执行构造函数将数据成员进行赋值, aa++是后加, 先赋值a=4, ++bb, bb变量值先自加为6, 再与a相乘, 所以b=24。

二、填空题

1. 在C++中, 编译指令都是以___(符号)开始。

答案: #

C++的源程序可包括各种编译指令, 以指示编译器对源代码进行编译之前先对其进行预处理。所

有的编译指令都以#开始, 每条指令单独占用一行, 同一行不能有其他编译指令和C++语句(注释例外)

2. 在函数前面用___保留字修饰时, 则表示该函数表为内联函数。

答案: (P59) inline

[解析] 内联函数, 用来提高程序运行速度。在类内部定义的函数也是内联函数。

3. 执行下列程序

```
int a=21, j=16;
cout.setf(ios::hex);
cout<<a<<" ";
cout.unsetf(ios::hex);
cout<<j<<endl;
```

程序的输出结果是___。

答案: (P196) 1516

[解析] a=21以十六进制输出, j=16以十进制输出。

4. 在单继承和多继承方式中, 面向对象的程序设计应尽量使用___继承。

答案: (P138) 单

[解析] 多继承是单继承的扩展, 且多继承易产生二义性等问题。

5. 函数模板中紧随template之后尖括号内的类型参数都要冠以保留字___。

答案: (P145) class

[解析] 类模板的使用。template <class T>, 也可以引入多参数的如:

```
template <class
T1, class T2, ..., class Tn>
```

6. 在C++中, 访问指针所指向的对象的成员使用___运算符。

答案: ->或.

[解析] 使用指针访问成员有两种方法: 成员运算符“.”或指向运算符“>”。

7. 定义类的动态对象数组时, 系统只能够自动调用该类的___构造函数对其进行初始化。

答案: (P80) 无参

[解析] 使用new创建对象数组, 调用无参构造函数。

8. 局部对象和全局对象中, 在同一程序中___生存期最长。

答案: 全局对象

变量或对象的生命期或者作用域的不同, 全局对象生命期长。

9. this指针始终指向调用成员函数的___。

答案: 对象

this指针是隐藏的指针, 它指向调用函数的对象。

10. 派生类的主要用途是可以定义其基类中___。

答案: (P127)不具有的成员

[解析] 继承的特点, 扩充基类, 即在派生类中增加基类不具有的成员。

11. 在用class定义一个类时, 数据成员和成员函数的默认访问权限是___。

答案: (P69)private

[解析] 定义类时的成员默认为私有, 而结构体则是公有。

12. 使用new为int数组动态分配10个存储空间是___。

答案: (P10)new int [10];

[解析] new delete动态开辟空间和删除空间。new int [10], 注意不要写成new

int (10), 使用小括号只能开辟一个空间, 使用10来初始化该值。

13. 类模板用来表达具有___的模板类对象集。

答案: (P145)相同处理方法

[解析] 模板特点是不同的数据具有相同的处理方法的抽象。

14. 如果通过同一个基类派生一系列的类, 则将这些类总称为___。

答案: (P174)类族

[解析] 单继承方式派生的众多的类。

15. 面向对象的四个基本特性是多态性、继承性、和封装性___。

答案: (P37)抽象

[解析] 考察面向对象的四个特征。程序由一组抽象的对象组成, 一组对象的共同特征抽象出类

的概念, 类是对象的抽象, 对象是类的实例。封装即将数据和操作紧密结合提供访问的接口, 外

部通过接口实现访问数据, 提供安全性。继承继承解决了类的扩展性。多态性不同对象调用相同

的函数名, 但调用不同的函数, 实现不同的功能, 解决了接口统一的问题。

16. 所有模板都是以___关键字和一个形参表开头的。

答案: (P61)template

[解析] 类模板, 函数模板都要使用template这一关键字。

17. 在C++语言中, 访问一个对象的成员所用的成员运算符是___。

答案: .

[解析] 成员运算符“.”, 如果是指针可以使用“->”。

18. 开发一个C++语言程序的步骤通常包括编辑、___、连接、运行和调试。

答案: (P21)编译

[解析] vc开发过程, 要经过编辑、编译、连接和运行四个步骤, 与其它高级语言相同。

数据结构

- 一个向量第一个元素的存储地址是100，每个元素的长度为2，则第5个元素的地址是（ 1 ）。
选择一项：
 1. 108
 2. 110
 3. 100
 4. 120
- 在 n 个结点的顺序表中，算法的时间复杂度是 $O(1)$ 的操作是（ b ）。
选择一项：
 a. 删除第 i 个结点 ($1 \leq i \leq n$)
 b. 访问第 i 个结点 ($1 \leq i \leq n$) 和求第 i 个结点的直接前驱 ($2 \leq i \leq n$)
 c. 将 n 个结点从小到大排序
 d. 在第 i 个结点后插入一个新结点 ($1 \leq i \leq n$)
- 以下说法错误的是（ d ）。
选择一项：
 a. 由于顺序存储要求连续的存储区域，所以在存储管理上不够灵活
 b. 顺序存储的线性表可以随机存取
 c. 求表长、定位这两种运算在采用顺序存储结构时实现的效率不比采用链式存储结构时实现的效率低
 d. 线性表的链式存储结构优于顺序存储结构
- 单链表的存储密度（ b ）。
选择一项：
 a. 不能确定
 b. 小于1
 c. 大于1
 d. 等于1
- 向一个有127个元素的顺序表中插入一个新元素并保持原来顺序不变，平均要移动 的元素个数为（ c ）。
选择一项：
 a. 63
 b. 7
 c. 63.5
 d. 8
- 在一个长度为 n 的顺序表中，在第 i 个元素 ($1 \leq i \leq n+1$) 之前插入一个新元素时须向后移动（ b ）个元素。
选择一项：
 a. $n-i$
 b. $n-i+1$
 c. i
 d. $n-i-1$
- 在单链表中，要将 s 所指结点插入到 p 所指结点之后，其语句应为（ a ）。
选择一项：
 a. $s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s;$
 b. $(*p). \text{next} = s; (*s). \text{next} = (*p). \text{next};$
 c. $s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s \rightarrow \text{next};$
 d. $s \rightarrow \text{next} = p+1; p \rightarrow \text{next} = s;$
- 在双向循环链表中，在 p 指针所指的结点后插入 q 所指向的新结点，其修改指针的操作是（ b ）。
选择一项：
 a. $p \rightarrow \text{next} = q; q \rightarrow \text{prior} = p; p \rightarrow \text{next} \rightarrow \text{prior} = q; q \rightarrow \text{next} = q;$
 b. $q \rightarrow \text{prior} = p; q \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} \rightarrow \text{prior} = q; p \rightarrow \text{next} = q;$
 c. $p \rightarrow \text{next} = q; p \rightarrow \text{next} \rightarrow \text{prior} = q; q \rightarrow \text{prior} = p; q \rightarrow \text{next} = p \rightarrow \text{next};$
 d. $q \rightarrow \text{prior} = p; q \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = q; p \rightarrow \text{next} \rightarrow \text{prior} = q;$
- 在双向链表存储结构中，删除 p 所指的结点时须修改指针（ c ）。
选择一项：
 a. $p \rightarrow \text{prior} = p \rightarrow \text{next} \rightarrow \text{next}; p \rightarrow \text{next} = p \rightarrow \text{prior} \rightarrow \text{prior};$
 b. $p \rightarrow \text{next} = p \rightarrow \text{next} \rightarrow \text{next}; p \rightarrow \text{next} \rightarrow \text{prior} = p;$
 c. $p \rightarrow \text{next} \rightarrow \text{prior} = p \rightarrow \text{prior}; p \rightarrow \text{prior} \rightarrow \text{next} = p \rightarrow \text{next};$
 d. $p \rightarrow \text{prior} \rightarrow \text{next} = p; p \rightarrow \text{prior} = p \rightarrow \text{prior} \rightarrow \text{prior};$
- 将两个各有 n 个元素的有序表归并成一个有序表，其最少的比较次数是（ c ）。
选择一项：
 a. $2n$
 b. $n-1$
 c. n
 d. $2n-1$
- 线性表 $L = (a_1, a_2, \dots, a_n)$ ，下列说法正确的是（ b ）。
选择一项：

- a. 表中诸元素的排列必须是由小到大或由大到小
b. 除第一个和最后一个元素外, 其余每个元素都有一个且仅有一个直接前驱和直接后继。
c. 每个元素都有一个直接前驱和一个直接后继
d. 线性表中至少有一个元素
12. 线性表若采用链式存储结构时, 要求内存中可用存储单元的地址 (d)。
选择一项:
a. 部分地址必须是连续的
b. 一定是不连续的
c. 必须是连续的
d. 连续或不连续都可以
13. 线性表 L 在 (d) 情况下适用于使用链式结构实现。
选择一项:
a. L 中结点结构复杂
b. L 中含有大量的结点
c. 需经常修改 L 中的结点值
d. 需不断对 L 进行删除插入
14. 若指定有 n 个元素的向量, 则建立一个有序单链表的时间复杂性的量级是 (b)。
选择一项:
a. $O(n \log n)$
b. $O(n^2)$
c. $O(n)$
d. $O(1)$
15. 链接存储的存储结构所占存储空间 (b)。
选择一项:
a. 只有一部分, 存储表示结点间关系的指针
b. 分两部分, 一部分存放结点值, 另一部分存放表示结点间关系的指针
c. 分两部分, 一部分存放结点值, 另一部分存放结点所占单元数
d. 只有一部分, 存放结点值
16. 一个具有 1025 个结点的二叉树的高 h 为 (a)。
选择一项:
a. 11 至 1025 之间
b. 10 至 1024 之间
c. 10
d. 11
17. 一棵完全二叉树上有 1001 个结点, 其中叶子结点的个数是 (d)。
选择一项:
a. 254
b. 500
c. 250
d. 501
18. 一棵非空的二叉树的先序遍历序列与后序遍历序列正好相反, 则该二叉树一定满足 (a)。
选择一项:
a. 只有一个叶子结点
b. 是任意一棵二叉树
c. 所有的结点均无左孩子
d. 所有的结点均无右孩子
19. 利用二叉链表存储树, 则根结点的右指针是 (c)。
选择一项:
a. 指向最右孩子
b. 指向最左孩子
c. 空
d. 非空
20. 在下列存储形式中, (a) 不是树的存储形式?
选择一项:
a. 顺序存储表示法
b. 孩子兄弟表示法
c. 孩子链表表示法
d. 双亲表示法
21. 对二叉树的结点从 1 开始进行连续编号, 要求每个结点的编号大于其左、右孩子的编号, 同一结点的左右孩子中, 其左孩子的编号小于其右孩子的编号, 可采用 (d) 遍历实现编号。
选择一项:
a. 从根开始按层次遍历
b. 先序
c. 中序
d. 后序
22. 引入二叉线索树的目的是 (d)。
选择一项:
a. 为了能方便的找到双亲

- b. 使二叉树的遍历结果唯一
c. 为了能在二叉树中方便地进行插入与删除
d. 加快查找结点的前驱或后继的速度
23. 把一棵树转换为二叉树后, 这棵二叉树的形态是 (c)。
- 选择一项:
- a. 有多种
b. 有多种, 但根结点都没有右孩子
c. 唯一的
d. 有多种, 但根结点都没有左孩子
24. 某二叉树的前序序列和后序序列正好相反, 则该二叉树一定是 (d) 的二叉树。
- 选择一项:
- a. 任一结点无右子树
b. 空或只有一个结点
c. 任一结点无左子树
d. 高度等于其结点数
25. 由3个结点可以构造出多少种不同的二叉树? (b)
- 选择一项:
- a. 2
b. 5
c. 4
d. 3
26. 线索二叉树是一种 (a) 结构。
- 选择一项:
- a. 物理
b. 逻辑和存储
c. 逻辑
d. 线性
27. 若 X 是二叉中序线索树中一个有左孩子的结点, 且 X 不为根, 则 X 的前驱为 (c)。
- 选择一项:
- a. X 的右子树中最左的结点
b. X 的左子树中最右叶结点
c. X 的左子树中最右结点
d. X 的双亲
28. 若二叉树采用二叉链表存储结构, 要交换其所有分支结点左右子树的位置, 利用 (b) 遍历方法最合适。
- 选择一项:
- a. 中序
b. 后序
c. 前序
d. 按层次
29. 设 F 是一个森林, B 是由 F 变换得的二叉树。若 F 中有 n 个非终端结点, 则 B 中右指针域为空的结点有 (b) 个。
- 选择一项:
- a. $n+2$
b. $n+1$
c. $n-1$
d. n
30. G 是一个非连通无向图, 共有28条边, 则该图至少有 (d) 个顶点。
- 选择一项:
- a. 10
b. 8
c. 7
d. 9
31. n 个顶点的连通图用邻接矩阵表示时, 该矩阵至少有 (c) 个非零元素。
- 选择一项:
- a. n
b. $n/2$
c. $2(n-1)$
d. $2n$
32. 下面 (a) 方法可以判断出一个有向图是否有环。
- 选择一项:
- a. 拓扑排序
b. 求关键路径
c. 求最短路径
d. 深度优先遍历
33. 下面 (d) 算法适合构造一个稠密图 G 的最小生成树。
- 选择一项:
- a. Dijkstra 算法
b. Floyd 算法

- c. Kruskal 算法 稀疏
d. Prim 算法
34. 具有 n 个顶点的有向图最多有 (b) 条边。
选择一项:
a. $2n$
b. $n(n-1)$
c. $n(n+1)$
d. n
35. 图的 BFS 生成树的树高比 DFS 生成树的树高 (a)。
选择一项:
a. 小或相等
b. 大或相等
c. 相等
d. 小
36. 在一个有向图中, 所有顶点的入度之和等于所有顶点的出度之和的 (a) 倍。
选择一项:
a. 1
b. $1/2$
c. 2
d. 4
37. 广度优先遍历类似于二叉树的 (c)。
选择一项:
a. 中序遍历
b. 后序遍历
c. 层次遍历
d. 先序遍历
38. 深度优先遍历类似于二叉树的 (a)。
选择一项:
a. 先序遍历
b. 后序遍历
c. 层次遍历
d. 中序遍历
40. 用邻接表表示图进行广度优先遍历时, 通常借助 (a) 来实现算法。
选择一项:
a. 队列
b. 栈
c. 树
d. 图
41. 用邻接表表示图进行深度优先遍历时, 通常借助 (d) 来实现算法。
选择一项:
a. 图
b. 树
c. 队列
d. 栈
42. 若从无向图的任意一个顶点出发进行一次深度优先搜索可以访问图中所有的顶点, 则该图一定是 (a) 图。
选择一项:
a. 连通
b. 有向
c. 非连通
d. 强连通
43. 在一个图中, 所有顶点的度数之和等于图的边数的 (b) 倍。
选择一项:
a. 4
b. 2
c. $1/2$
d. 1
44. m 阶 B-树是一棵 (b)。
选择一项:
a. m 叉排序树
b. m 叉平衡排序树
c. $m+1$ 叉平衡排序树
d. $m-1$ 叉平衡排序树
45. 下列关于 m 阶 B-树的说法错误的是 (d)。
选择一项:
a. 所有叶子都在同一层次上
b. 根结点至多有 m 棵子树
c. 非叶结点至少有 $m/2$ (m 为偶数) 或 $m/2+1$ (m 为奇数) 棵子树
d. 根结点中的数据是有序的

46. 下面关于 B-和 B+树的叙述中, 不正确的是 (a)。

选择一项:

- a. B-树和 B+树都能有效地支持顺序检索
- b. B-树和 B+树都可用于文件的索引结构
- c. B-树和 B+树都是平衡的多叉树
- d. B-树和 B+树都能有效地支持随机检索

47. 下面关于哈希查找的说法, 不正确的是 (c)。

选择一项:

- a. 用链地址法处理冲突, 适合表长不确定的情况
- b. 用链地址法处理冲突, 不会引起二次聚集现象
- c. 采用链地址法处理冲突时, 查找一个元素的时间是相同的
- d. 采用链地址法处理冲突时, 若插入规定总是在链首, 则插入任一个元素的时间是相同的

48. 下面关于哈希查找的说法, 正确的是 (d)。

选择一项:

- a. 哈希函数构造的越复杂越好, 因为这样随机性好, 冲突小
- b. 哈希表的平均查找长度有时也和记录总数有关
- c. 除留余数法是所有哈希函数中最好的
- d. 不存在特别好与坏的哈希函数, 要视情况而定

49. 分别以下列序列构造二叉排序树, 与用其它三个序列所构造的结果不同的是 (d)。

小左

大右

选择一项:

- a. (100, 80, 60, 90, 120, 130, 110)
- b. (100, 120, 110, 130, 80, 60, 90)
- c. (100, 80, 90, 60, 120, 110, 130)
- d. (100, 60, 80, 90, 120, 110, 130)

50. 在平衡二叉树中插入一个结点后造成了不平衡, 设最低的不平衡结点为 A, 并已知 A 的左孩子的平衡因子为 0 右孩子的平衡因子为 1, 则应作 (d) 型调整以使其平衡。

选择一项:

- a. LL
- b. LR
- c. RR
- d. RL

51. 对 22 个记录的有序表作折半查找, 当查找失败时, 至少需要比较 (a) 次关键字。

选择一项:

- a. 4
- b. 6
- c. 5
- d. 3

52. 当在一个有序的顺序表上查找一个数据时, 既可用折半查找, 也可用顺序查找, 但前者比后者的查找速度 (c)。

选择一项:

- a. 不一定
- b. 取决于表递增还是递减
- c. 在大部分情况下要快
- d. 必定快

53. 折半搜索与二叉排序树的时间性能 (c)。

选择一项:

- a. 完全不同
- b. 相同
- c. 有时不相同
- d. 数量级都是 $O(\log 2n)$

54. 折半查找有序表 (4, 6, 10, 12, 20, 30, 50, 70, 88, 100)。若查找表中元素 58, 则它将依次与表中 (c) 比较大小, 查找结果是失败。

选择一项:

- a. 20, 50
- b. 30, 88, 70, 50
- c. 20, 70, 30, 50
- d. 30, 88, 50

55. 设哈希表长为 14, 哈希函数是 $H(key) = key \% 11$, 表中已有数据的关键字为 15, 38, 61, 84 共四个, 现要将关键字为 49 的元素加到表中, 用二次探测法解决冲突, 则放入的位置是 (b)。

选择一项:

- a. 5
- b. 9
- c. 8
- d. 3

56. 适用于折半查找的表的存储方式及元素排列要求为 (b)。

选择一项:

- a. 顺序方式存储, 元素无序
- b. 顺序方式存储, 元素有序

- c. 链接方式存储，元素无序
d. 链接方式存储，元素有序
57. 采用线性探测法处理冲突，可能要探测多个位置，在查找成功的情况下，所探测的这些位置上的关键字（d）。
- 选择一项：
a. 一定都不是同义词
b. 都相同
c. 一定都是同义词
d. 不一定都是同义词
58. 对 n 个元素的表做顺序查找时，若查找每个元素的概率相同，则平均查找长度为（c）。
- 选择一项：
a. $(n-1)/2$
b. n
c. $(n+1)/2$
d. $n/2$
59. 下列关键字序列中，（c）是堆。
- 选择一项：
a. 94, 23, 31, 72, 16, 53
b. 16, 53, 23, 94, 31, 72
c. 16, 23, 53, 31, 94, 72
d. 16, 72, 31, 23, 94, 53
60. 下列排序算法中，（b）不能保证每趟排序至少能将一个元素放到其最终的位置上。
- 选择一项：
a. 冒泡排序
b. 希尔排序
c. 堆排序
d. 快速排序
61. 下述几种排序方法中，要求内存最大的是（c）。
- 选择一项：
a. 希尔排序
b. 堆排序
c. 归并排序
d. 快速排序
62. 下述几种排序方法中，（c）是稳定的排序方法。
- 选择一项：
a. 快速排序
b. 希尔排序
c. 归并排序
d. 堆排序
63. 从未排序序列中挑选元素，并将其依次放入已排序序列（初始时空）的一端的方法，称为（c）。
- 选择一项：
a. 冒泡排序
b. 插入排序
c. 选择排序
d. 归并排序
64. 堆是一种（a）排序。
- 选择一项：
a. 选择
b. 归并
c. 交换
d. 插入
65. 堆的形状是一棵（b）。
- 选择一项：
a. 满二叉树
b. 完全二叉树
c. 二叉排序树
d. 平衡二叉树
66. 对 n 个不同的关键字由小到大进行冒泡排序，在下列（b）情况下比较的次数最多。
- 选择一项：
a. 元素基本有序
b. 从大到小排列好的
c. 从小到大排列好的
d. 元素无序
67. 对 n 个不同的排序码进行冒泡排序，在元素无序的情况下比较的次数最多为（b）。
- 选择一项：
a. $n-1$
b. $n(n-1)/2$
c. $n+1$

- d. n
68. 对 n 个关键字作快速排序，在最坏情况下，算法的时间复杂度是 (c)。
- 选择一项：
- a. $O(n \log 2n)$
 - b. $O(n^3)$
 - c. $O(n^2)$
 - d. $O(n)$
69. 快速排序在下列 (b) 情况下最易发挥其长处。
- 选择一项：
- a. 被排序的数据中含有多个相同排序码
 - b. 被排序的数据完全无序
 - c. 被排序的数据已基本有序
 - d. 被排序的数据中的最大值和最小值相差悬殊
70. 数据表中有10000个元素，如果仅要求求出其中最大的10个元素，则采用 (c) 算法最节省时间。
- 选择一项：
- a. 堆排序
 - b. 冒泡排序
 - c. 简单选择排序
 - d. 快速排序
71. 若一组记录的排序码为 (46, 79, 56, 38, 40, 84)，则利用快速排序的方法，以第一个记录为基准得到的一次划分结果为 (b)。
- 选择一项：
- a. 40, 38, 46, 79, 56, 84
 - b. 40, 38, 46, 56, 79, 84
 - c. 38, 40, 46, 56, 79, 84
 - d. 40, 38, 46, 84, 56, 79
72. 若一组记录的排序码为 (46, 79, 56, 38, 40, 84)，则利用堆排序的方法建立的初始堆为 (a)。
- 选择一项：
- a. 84, 79, 56, 38, 40, 46
 - b. 84, 79, 56, 46, 40, 38
 - c. 79, 46, 56, 38, 40, 84
 - d. 84, 56, 79, 40, 46, 38
73. 从未排序序列中依次取出元素与已排序序列中的元素进行比较，将其放入已排序序列的正确位置上的方法，这种排序方法称为 (b)。
- 选择一项：
- a. 冒泡排序
 - b. 插入排序
 - c. 归并排序
 - d. 选择排序

面试相关

中科大复试准备

数据结构->操作系统->计算机网络->通信原理->微机原理->

软件工程, 编译原理, 数据库

数据结构

1. 时间复杂度

时间复杂度是指执行算法所需要的计算工作量，因为整个算法的执行时间与基本操作重复执行的次数成正比，所以将算法中基本操作的次数作为算法时间复杂度的度量，一般情况下，按照基本操作次数最多的输入来计算时间复杂度，并且多数情况下我们去最深层循环内的语句所描述的操作作为基本操作。

2. 循环队列的顺序表中，为什么要空一个位置？

这是为了用来区分队空与队满的情况。如果不空一个位置，则判断队空和队满的条件是一样的。

3. 什么是二叉排序树？以及它的原理，算法。(二叉排序树的查找过程)

二叉排序树又称二叉查找树，它或者是一颗空树，或者满足一下性质的二叉树：

- ① 若左子树不空，则左子树上所有结点的值均小于根结点的值；
- ② 若右子树不空，则右子树上所有结点的值均大于根结点的值；
- ③ 左右子树也分别是二叉排序树。

原理步骤：

若根结点的关键字值等于查找的关键字，成功。

否则，若小于根结点的关键字值，递归查左子树。

若大于根结点的关键字值，递归查右子树。

若子树为空，查找不成功。

4. 哈夫曼树

定义：

给定 n 个权值作为 n 个叶子结点，构造一棵二叉树，若带权路径长度达到最小，称这样的二叉树为最优二叉树，也称为哈夫曼树(Huffman tree)。

构造方法：

假设有 n 个权值，则构造出的哈夫曼树有 n 个叶子结点。 n 个权值分别设为 w_1 、 w_2 、...、 w_n ，则哈夫曼树的构造规则为：

- (1) 将 w_1 、 w_2 、...、 w_n 看成是有 n 棵树的森林(每棵树仅有一个结点)；
- (2) 在森林中选出两个根结点的权值最小的树合并，作为一棵新树的左、右子树，且新树的根结点权值为其左、右子树根结点权值之和；
- (3) 从森林中删除选取的两棵树，并将新树加入森林；
- (4) 重复(2)、(3)步，直到森林中只剩一棵树为止，该树即为所求得的哈夫曼树。

特点：

- ① 权值越大的结点，距离根节点越近；
- ② 树中没有度为一的结点。

应用：

哈夫曼编码，减少编码的长度。哈夫曼编码就是长度最短的前缀编码。

5. 什么是哈希冲突？以及如何解决。

散列（哈希）表：

根据关键码值(Key value)而直接进行访问的数据结构。根据给定的关键字来计算出关键字在表中的地址，

以加快查找的速度。

冲突：指的是多个关键字映射同一个地址的情况。

解决办法：

(1) 开放定址法

① 线性探查法（产生堆积问题）；

② 平方探查法（不能探查哈希表上所有的地址，但至少能探查一半的地址）

(2) 链地址法

把所有的同义词用单链表连接起来。

补充（常见的哈希函数构造方法）

直接定址法，数字分析法，平方取中法，除留余数法。

6. 深度优先搜索遍历和广度优先搜索遍历的过程

深度优先搜索遍历

基本思想：首先访问出发点 V ，并将其标记为已访问；然后选取与 V 邻接的未被访问的邻接顶点 W ，访问 W ；再选取与 W 邻接的未被访问的顶点访问，以此类推。当一个顶点所有的邻接顶点都被访问过时，则依次退回最近被访问过的顶点，若该顶点还有其他邻接顶点未被访问，则从这些顶点中去一个顶点进行上述的过程，直至图中所有顶点都被访问过为止。

广度优先搜索遍历

基本思想：首先访问起始顶点 V ，然后选取与 V 邻接的全部顶点 w_1, w_2, \dots, w_n 进行访问，再一次访问与 w_1, w_2, \dots

, w_n 邻接的全部顶点（不包括已访问过的顶点），以此类推，直至所有顶点都被访问过为止。

7. 迪杰斯特拉算法的过程

该算法可以求得某一顶点到其余各顶点的最短路径。

算法思想：设有两个顶点集合 S 和 T ，其中集合 S 中存放的是图中已找到最短路径的顶点，集合 T 中存放的是图中的剩余顶点。

初始状态时，集合 S 中只包含源点 V_0 ，然后不断从集合 T 中选取到顶点 V_0 路径最短的顶点 V_u 并加入集合 S 中。集合 S 每加入一个新的顶点 V_u ，都要修改 V_0 到集合 T 中各个顶点的最短路径的长度值。不断重复这个过程，直至集合 T 中的顶点全部并入到 S 中为止。

8. 链表查找某个元素，平均的时间复杂度是多少？

$O(n)$ 链表是顺序存储，故 $(1+n)/2$ 。

9. 图的存储方式

- ① 邻接矩阵：是图的顺序存储结构，用两个数组分别存储数据元素（顶点）信息和数据元素之间的关系（边/弧）的信息。图的邻接矩阵表示是唯一的，无向图的邻接矩阵是对称的。
- ② 邻接表：是图的链式存储结构，由单链表的表头形成的顶点表和单链表其余结点所形成的边表两部分组成。
- ③ 十字链表：有向图的另一种链式存储结构。
- ④ 邻接多重表：无向图的链式存储结构。

10. 图的深度遍历是否唯一

不一定是唯一。我们可以取图中任一顶点进行深度遍历。

11. 图的相关概念

图：由结点的有穷集合 V 和边的集合 E 组成。

类别：有向图和无向图。

顶点的度：出度和入度。

有向完全图和无向完全图：若有向图有 n 个顶点，则最多有 $n(n-1)$ 条边，则称为有向完全图；

若无向图有 n 个顶点，则最多有 $n(n-1)/2$ 条边，则称为无向完全图。

路径：相邻顶点序列所构成的序列。

简单路径：序列中的顶点和边不重复出现的路径。

回路：路径中第一个顶点和最后一个顶点相同的路径。

连通：无向图中，如果 V_i 到 V_j 有路径，则称这两个顶点连通。如果图中任意两个顶点之间都连通，则称该图为连通图。

有向图中，如果 V_i 到 V_j 有路径，则称这两个顶点连通。如果图中每一对顶点 V_i 和 V_j ，从 V_i 到 V_j 和 V_j 到 V_i 都有路径，则称该图为强连通图。

12. 最小生成树的概念

一个有 n 个结点的连通图的生成树是原图的极小连通子图，且包含原图中的所有 n 个结点，并且有保持图连通的最少的边。如果在最小生成树中添加一条边，必定成一个环。

相关算法：

- ① 普里姆算法
- ② 克鲁斯卡尔算法

N 个结点的最小生成树有几个结点，几条边： n 个结点， $n-1$ 条边。

13. 平衡二叉树

平衡二叉树又称 AVL 树，是一种特殊的二叉排序树，其左右子树都是平衡二叉树，且左右子树的高度差的绝对值不超过 1。

平衡因子：左子树高度减去右子树高度的差。

平衡调整：先找到失去平衡的最小子树，即以距离插入结点最近，且平衡因子绝对值大于 1 的结点最为根结点的子树，分为 LL, LR, RL, RR 四中调节方式。

14. 二叉树的存储

- ① 顺序存储结构：用一个数组来存储一颗二叉树，二叉树中的结点值按照编号依次存入一个一维数组中。适用于完全二叉树，若用于一般的二叉树则会浪费大量存储空间。

Lchild	Data	Rchild
--------	------	--------

- ② 链式存储结构：二叉树中的每一个结点用一个链结点来存放。

15. M 阶 B-树和 M 阶 B+树的主要区别

- ① B+树所有有效数据全在叶子节点，而 B-树所有节点分散在树中，B-树中的关键字不重复。
- ② B+树种有几个关键字就有几个子树，B-树中具有 n 个关键字的节点含有 $(n+1)$ 棵子树。
- ③ B+树有两个指针，根指针和只想最小节点的指针，叶子节点连接成一个不定长的线性链表
- ④ B+树中，每个节点（除根节点外）中的关键字个数 n 的取值范围是 $\lceil m/2 \rceil \leq n \leq m$ ，根节点 n 的取值

- ⑤ 范围是 $2 \leq n \leq m$ 。B-树中，每个节点（除根节点外的所有最底层非叶子节点）中的关键字取值范围是
- ⑥ $\lceil m/2 \rceil - 1 \leq n \leq m - 1$ ，根节点 n 的取值范围是 $1 \leq n \leq \lceil m/2 \rceil$ 。
- ⑦ B+树中的所有非叶子节点仅仅起到索引的作用，节点中的每个索引项只包含对应子树的最大关键字和指向该子树的指针，不含有该关键字对应记录的存储地址。而在 B-树中，每个关键字对应记录的存储地址。

16. 折半查找，以及其适用范围和时间复杂度

又称二分查找，基本思路：

在当前的查找区间 $[low...high]$ 中，首先确定 $mid = (low + high) / 2$ ，然后拿关键字与 mid 比较，若相等则查找成功，返回该位置，否则确定新的查找区间， $mid > K$, $[low...mid - 1]$

$mid < K$, $[mid + 1...high]$

直至查找自区间长度小于 1 时查找结束。

适用范围：顺序结构存储并按照关键字大小有序排列。

时间复杂度： $O(\log_2 N)$

17. 完全二叉树

若一棵二叉树至多只有最下面的两层上的结点的度数可以小于 2，并且最下层上的结点都集中在该层最左边的若干位置上，则此二叉树成为完全二叉树。

完全二叉树特点：

叶子结点只可能在最大的两层上出现，对任意结点，若其右分支下的子孙最大层次为 L ，则其左分支下的子孙的最大层次必为 L 或 $L + 1$ ；

18. 什么是堆？有什么作用？

堆是一种数据结构，可以把堆看成一个完全二叉树，并且这个完全二叉树满足：

任何一个非叶节点的值都不大于（或不小于）其左右子树的结点的值。若父亲大孩子小，则为大顶堆，若父亲肖孩子大，则为小顶堆。

作用：应用于堆排序。

19. 如何实现循环队列？有何好处？

如何实现：把数组弄成一个环，让 $rear$ 和 $front$ 指针沿着环走，这样就可以产生循环队列。

好处：循环队列是顺序队列的改进，在顺序队列中，在元素进队的时候， $rear$ 要向后移动，元素出队的时候， $front$ 也要向后移动，这样经过一系列的出队和入队操作之后，两个指针最后会达到数组的末端，此时虽然队中已经没有元素了，但是还是不能让元素入队，即出现了“假溢出”的现象。循环队列就能避免出现这个现象。

20. 深度优先搜索形成的是什么？森林唯一么？

（森林，不能说树）（不唯一，因为邻接表可能不唯一）

21. 满二叉树的结点个数（ n 层）

2 的 n 次方减一 ($2^n - 1$)

22. 二叉查找树查找的时间复杂度以及中序遍历后得到什么样的序列

递增有序序列

23. 什么图可以进行拓扑排序?

有向无环图

24. 顺序队列的特征

队列是一种操作受限的线性表，只允许队尾入队，在队头进行出队。最大的特点是先进先出。

25. 非连通图如何访问每一个结点?

排序总结

	排序方法	时间复杂度			空间复杂度	稳定性
		平均情况	最坏情况	最好情况		
插入排序	直接插入	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
	折半插入	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
	希尔排序	$O(n \log_2 n)$		$O(n \log_2 n)$	$O(1)$	不稳定
交换排序	冒泡排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
	快速排序	$O(n \log_2 n)$	$O(n^2)$	$O(n \log_2 n)$	$O(\log_2 n)$	不稳定
选择排序	简单选择	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
	堆积排序	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(1)$	不稳定
其他排序	二路归并	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n)$	稳定
	基数排序	$O(d(n+r))$	$O(d(n+r))$	$O(d(n+r))$	$O(r)$	稳定

各类排序的算法详见书本。(需要说出每个算法的基本思想)

查找：相关章节过一遍。

操作系统

1. 进程和程序的区别

- ① 进程是动态的，程序是静止的。进程是程序的执行，程序是有序代码的集合。
- ② 进程是暂时的，程序是永久的。进程是一个状态变化的过程，程序可以长久保存。
- ③ 进程和程序的组成不同：进程包括程序，数据和进程控制块。
- ④ 进程和程序是密切相关的。通过多次执行，一个程序可以对应多个进程；通过调度关系，一个进程可以包括多个程序。
- ⑤ 进程可以创建其他进程，但是程序不能形成新的程序。

2. 进程和线程的区别

- ① 调度：线程是独立调度的基本单位，进程是资源拥有的基本单位。在同一进程中，线程的切换不会引起进程切换。在不同进程中进行线程切换，将会引起进程切换。
- ② 拥有资源：进程是拥有资源的基本单位，而线程不拥有系统资源（除了少量资源，比如栈，程序计数器，寄存器），不过线程可以访问其隶属进程的系统资源。
- ③ 并发性：在引入线程的操作系统中，不仅进程之间可以并发执行，而且同一个进程内的多个线程之间也可以并发执行，能提高系统的吞吐量，系统的并发性也更好。
- ④ 系统开销：在创建进程和撤销进程时，系统都要为之分配或回收资源，所以操作系统为进程付出的系统开销远大于创建线程或撤销线程的开销。
- ⑤ 同步和通信：多线程之间的同步和通信容易实现。

3. 什么是微内核？

微内核操作系统能有效地支持多处理机运行，非常适用于分布式系统环境。

什么是微内核操作系统到现在没有一致公认的定义，但是可以从四个方面对微内核操作系统进行描述：

- ① 足够小的内核：在微内核操作系统中，内核是指精心设计的，能实现现代 OS 最基本核心功能的部分，并非是一个完整的 OS，而只是 OS 中最基本的部分。
- ② 基于 C/S 模式：将操作系统中最基本的部分放入内核中，而把操作系统的绝大部分功能都放于微内核外面的一组服务器中实现。
- ③ 应用“极致与策略分离”原理：在传统 OS 中，讲极致放在 OS 的内核的较低层，把策略放在内核的较高层中。而在微内核 OS 中，通常把机制放在 OS 的微内核中，这样才有可能将内核做得很小。
- ④ 采用面向对象技术。

基本功能：

- ① 进程（线程）管理
- ② 低级存储器管理
- ③ 中断和陷入处理

优点：

- ① 提高了系统的可扩展性
- ② 增强系统的可靠性
- ③ 可移植性
- ④ 提供了对分布式系统的支持
- ⑤ 融入了面向对象技术

4. 什么是 DMA? 什么是中断? 两者的区别。

5. 硬中断和软中断是什么? 区别是什么?

软中断:

- 1、编程异常通常叫做软中断
- 2、软中断是通讯进程之间用来模拟硬中断的一种信号通讯方式。
- 3、中断源发中断请求或软中断信号后,CPU 或接收进程在适当的时机自动进行中断处理或完成软中断信号对应的功能
- 4、软中断是软件实现的中断,也就是程序运行时其他程序对它的中断;而硬中断是硬件实现的中断,是程序运行时设备对它的中断。

硬中断:

- 1、硬中断是由外部事件引起的因此具有随机性和突发性;软中断是执行中断指令产生的,无外部施加中断请求信号,因此中断的发生不是随机的而是由程序安排好的。
- 2、硬中断的中断响应周期,CPU 需要发中断回合信号(NMI 不需要),软中断的中断响应周期,CPU 不需发中断回合信号。
- 3、硬中断的中断号是由中断控制器提供的(NMI 硬中断中断号系统指定为 02H);软中断的中断号由指令直接给出,无需使用中断控制器。
- 4、硬中断是可屏蔽的(NMI 硬中断不可屏蔽),软中断不可屏蔽。

区别:

- 1、软中断发生的时间是由程序控制的,而硬中断发生的时间是随机的
- 2、软中断是由程序调用发生的,而硬中断是由外设引发的
- 3、硬件中断处理程序要确保它能快速地完成它的任务,这样程序执行时才不会等待较长时间。

6. 页面置换算法有哪些? 什么是 LRU?

- ① 最佳置换算法 (OPT): 在预知一个进程的页面号引用串的情况下,每次都淘汰以后不再使用的页以后最迟再被使用的页面。该算法不能实现,只能作为一个标准来衡量其他置换算法的优劣。
- ② 先进先出算法 (FIFO): 每次总是淘汰最先进入内存的页面,也就是将在内存中驻留时间最长的页面淘汰。(可能会产生 Belady 异常,缺页次数随着分配的物理块的增加而增加)。
- ③ 最近最少使用算法 (LRU): 选择最近最少未被使用的页面淘汰,其思想是用以前的页面引用情况来预测将来会出现的页面引用情况。利用了局部性原理。
- ④ 时钟置换算法 (CLOCK): 是 LRU 和 FIFO 的折中,具体方法略。
- ⑤ 工作集算法
- ⑥ 工作集时钟算法
- ⑦ 第二次机会算法

⑧ 最近未使用 (NRU)

7. 操作系统中的磁盘调度算法

磁盘调度算法目的：使磁盘的平均寻道时间最少。

调度算法	算法思想	优点	缺点
先来先服务算法 FCFS	按照进程请求访问磁盘的先后顺序进行调度。	简单，公平。	未对寻道进行优化，平均寻道时间较长，仅适用于磁盘请求较少的场合。
最短寻道时间优先算法 SSTF	选择与当前磁头所在磁道距离最近的请求作为下一次服务的对象。	较 FCFS 有较好的寻道性能以及较少的寻道时间。	会导致饥饿现象
扫描（电梯调度）算法 SCAN	在磁头当前移动方向上选择与当前磁头所在磁道距离最近的请求最为下一次服务的对象。	具有较好的寻道性能，而且防止了饥饿现象。	存在一个请求刚好被错过，要等待很久的情形。
循环扫描算法 CSCAN	规定磁头单向移动，如自里向外移动，当磁头移动到最外的磁道时立即返回到最里磁道，如此循环进行扫描。	兼顾较好的寻道性能，防止饥饿现象，同时解决了一个请求等待时间过长的的问题。	可能出现磁臂长时间停留，处理不懂的情况（磁臂黏着）
N-Step-SCAN 算法，对 SCAN 算法的优化。	将磁盘请求队列分成若干个长度为 N 的子队列，磁盘调度将按照 FCFS 依次处理这些子队列，而每处理一个队列时又是按照 SCAN 算法，对一个队列处理后再处理其他队列，将新请求队列放入新队列。	无磁臂黏着。	
FSCAN 算法，对 SCAN 算法的优化。	将请求队列分成两个子队列，将新出现请求磁盘 IO 的进程放入另一个子队列。	无磁臂黏着。	

8. 操作系统中的信号量

信号量是一个确定的二元组 (s, q) ，其中 s 是一个具有非负初值的整型变量， q 是一个初始状态为空的队列。整型变量 s 表示系统中某类资源的数目，当其值大于 0 时，表示系统中当前可用资源的数目；当其值小于 0 时，其绝对值表示系统中因请求该类资源而被阻塞的进程数目。

信号量分类：

- ① 整型信号量：所谓整型信号量就是一个用于表示资源个数的整型量
- ② 记录型信号量（资源信号量）：就是用了一个结构体实现，里面包含了表示资源个数的整型量和一个等待队列。

信号量的应用：

- ① 实现进程同步
- ② 实现进程互斥

9. PV 操作

信号量的值除了初值外，仅能由这 PV 原语加以改变。P、V 操作以原语形式实现，保证了对信号量进行操作过程中不会被打断或阻塞。P 操作相当于申请资源，V 操作相当于释放资源。P 操作和 V 操作必定成对出现，但未必在同一个进程中。

```
Struct semaphore{
    Int count;
    queueType queue;
};

Wait (semaphore s)      // P
{
    s.count --;
    if(s.count<0)
    {    阻塞该进程;
        将该进程插入等待序列 s.queue;
    }
}

signal (semaphore s)    // V
{
    s.count ++;
    if(s.count<=0)
    {    从等待队列 s.queue 取出第一个进程 p;
        将 p 插入就绪队列;
    }
}
```

10. 什么是操作系统?

操作系统（Operating System，简称 OS）是管理和控制计算机硬件与软件资源的计算机程序，是直接运行在“裸机”上的最基本的系统软件，任何其他软件都必须在操作系统的支持下才能运行。操作系统是用户和计算机的接口，同时也是计算机硬件和其他软件的接口。操作系统的功能包括管理计算机系统的硬件、软件及数据资源，控制程序运行，改善人机界面，为其它应用软件提供支持等，使计算机系统所有资源最大限度地发挥作用，提供了各种形式的用户界面，使用户有一个好的工作环境，为其它软件的开发提供必要的服务和相应的接口。

11. 操作系统的组成

12. 操作系统中用到了哪些数据结构中的数据结构？请举例说明

13. 简述操作系统中系统调用过程

系统调用提供了用户程序和操作系统之间的接口，应用程序通过系统调用实现其余 OS 的通信，并取得它的服务。系统调用不仅可供所有的应用程序使用，而且也可供 OS 本身的其它部分，如命令处理程序。

系统调用的处理步骤（三步）：

首先，将处理机状态由用户态转为系统态；然后由硬件和内核程序进行系统调用的一般性处理，即首先保护被中断进程的 CPU 环境，将处理机状态字 PSW、程序计数器 PC、系统调用号、用户栈指针以及通用寄存器内容等压入堆栈；再然后将用户定义参数传送到指定的地址保存起来。

其次，分析系统调用类型，转入相应的系统调用处理子程序。（通过查找系统调用入口表，找到相应处理子程序的入口地址转而去执行它。）

最后，在系统调用处理子程序执行完后，应恢复被中断的进程设置新进程的 CPU 现场，然后返回被中断进程或新进程，继续往下执行。

14. 虚拟存储器，以及相关算法。

基于局部性原理，应用程序在运行之前，仅将那些当前要运行的少数页面或段先装入内存便可运行，其余部分暂时留在盘上。程序运行时，如果它要访问的页已调入内存，便可继续执行下去；但如果程序要访问的页或段尚未调入内存（即缺页），此时程序应利用请求调入功能将它们调入内存，以使程序能继续执行下去。如果此时内存已满，无法装入新的页或段，则需要利用页面置换功能，将内存中暂不使用的页面或段调至盘上，腾出空间用于页面调入内存，是程序继续执行下去。这样，就实现了大的用户程序能在较小的内存空间里运行，也可以在内存中同时装入更多的进程使它们并发运行。从用户角度出发，该系统的内存容量比实际内存容量大很多，故成这样的存储器为虚拟存储器。

相关算法：

页面置换算法

- ① 最佳置换算法（OPT）：在预知一个进程的页面号引用串的情况下，每次都淘汰以后不再使用的页以后最迟再被使用的页面。该算法不能实现，只能作为一个标准来衡量其他置换算法的优劣。
- ② 先进先出算法（FIFO）：每次总是淘汰最先进入内存的页面，也就是将在内存中驻留时间最长的页面淘汰。（可能会产生 Belady 异常，缺页次数随着分配的物理块的增加而增加）。
- ③ 最近最少使用算法（LRU）：选择最近最少未被使用的页面淘汰，其思想是用以前的页面引用情况来预测将来会出现的页面引用情况。利用了局部性原理。
- ④ 时钟置换算法（CLOCK）：是 LRU 和 FIFO 的折中，具体方法略。
- ⑤ 工作集算法
- ⑥ 工作集时钟算法
- ⑦ 第二次机会算法
- ⑧ 最近未使用（NRU）

15. 存储器管理应具有的功能

存储管理的主要任务是为多道程序的运行提供良好的环境，方便用户使用存储器，提高存储器的利用率以及从逻辑上扩充存储器，故应具有以下功能：

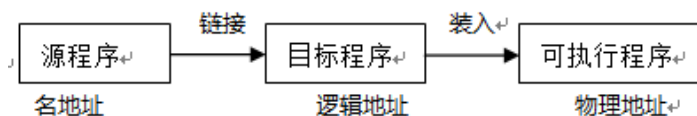
- ① 内存的分配和回收：实施内存的分配，回收系统或用户释放的内存空间。
- ② 地址变换：提供地址变换功能，将逻辑地址转换成物理地址。
- ③ 扩充内存：借助于虚拟存储技术或其他自动覆盖技术，为用户提供比内存空间大的地址空间，从逻辑上扩充内存。
- ④ 存储保护：保证进入内存的各道作业都在自己的存储空间内运行，互不干扰。

16. 什么是 TLB?

TLB 的作用是在处理器访问内存数据的时候做地址转换。TLB 的全称是 Translation Lookaside Buffer，可以翻译做旁路缓冲，是一个具有并行查询能力的特殊高速缓冲寄存器。TLB 中存放了一些页表文件，文件中记录了虚拟地址和物理地址的映射关系。当应用程序访问一个虚拟地址的时候，会从 TLB 中查询出对应的物理地址，然后访问物理地址。TLB 通常是一个分层结构，使用与 Cache 类似的原理。处理器使用一定的算法把最常用的页表放在最先访问的层次。

17. 程序的装入方式有哪些?

补充：应用程序从用户编写的源文件到内存中执行的进程大致分为三个阶段，经过编译程序将源代码便以为若干个目标模块，在通过链接程序将编译好的目标模块以及所需的库函数链接到一起，形成完整的装入模块，最后通过装入程序将这些装入模块装入内存并执行。（编译，链接，装入）



装入方式：

- ① 绝对装入：在编译时就知道程序将要驻留在内存的物理地址，编译程序产生含有物理地址的目标代码，不适合多道程序设计。
- ② 可重定位装入：根据内存当前情况，将装入模块装入到内存的适当位置，地址变换通常在装入时一次完成，之后不再改变，也称静态重定位。当操作系统为程序分配一个以某地址为起始地址的连续主存区域后，重定位时将程序中指令或操作数的逻辑地址加上这个起始地址就得到了物理地址。
- ③ 动态运行装入：允许程序运行时在内存中移动位置，把装入模块装入到内存后的所有地址都是相对地址，在程序执行过程中每当访问到相应指令或数据时，才将要访问的程序或数据的相对地址转换为物理地址。动态重定位的实现要依靠硬件地址变换机构。

18. 程序的链接方式有哪些?

- ① 静态链接：在程序运行之前，先把各个目标模块及所需库链接为一个完整的可执行程序，以后不再拆开。
- ② 装入时动态链接：将应用程序编译后所得到的一组目标模块在装入内存时采用边装入边链接的链接方式。
- ③ 运行时动态链接：知道程序运行过程中需要一些模块时，才对这些模块进行链接。

19. 交换技术，覆盖技术，以及两者的区别。

覆盖技术：把一个大的程序划分为一系列覆盖，每个覆盖是一个相对独立的程序单位，把程序执行时并不要求同时装入内存的覆盖组成一组，成为覆盖段，这个覆盖段分配到同一个存储区域，这个存储区域成为覆盖区，它与覆盖段一一对应。覆盖段的大小由覆盖段中最大的覆盖来确定。（为了解决内存容量太小的问题，打破了必须将一个程序全部信息装入内存后才能运行的限制）

交换技术：把暂时不用的某个程序及数据部分从内存移到外存中去，以便腾出必要的内存空间；或者把指定的程序或数据从外存读到相应的内存中，并将控制权交给他，让其在系统上运行的一种内存扩充技术。处理器的中级调度就是采用交换技术。

区别：

- ① 与覆盖技术相比，交换技术不要求程序员给出的程序段之间的覆盖结构；

- ② 交换技术主要在进程和作业之间进行，覆盖技术主要在同一个进程或作业中进行；
- ③ 覆盖技术只能覆盖于覆盖程序段无关的程序段，交换进程由换出和换入两个过程组成。

20. 内存连续分配管理方式有哪些？

- ① 单一连续分配（静态分配）
- ② 固定分区分配（分区大小可以不等，但事先必须确定，运行时不能改变）
- ③ 动态分区分配

P131 详细

21. 动态分区分配的算法有哪些？

- ① 首次适应算法 First Fit
- ② 循环首次适应算法 Next Fit
- ③ 最佳适应算法 Best Fit
- ④ 最差适应算法 Worst Fit

22. 什么叫拼接技术？

在分区管理方式下，系统运行一段时间后，内存中会出现相当一部分的碎片，拼接技术是解决碎片问题的方法。

即将存储器中所有已分配分区移动到主存的一端，使本来分散的多个小空闲区连成一个大的空闲区，这种通过移动把多个分散的小分区拼接成一个大分区的方法即为拼接技术。

23. 什么叫原子操作？

24. 内部碎片和外部碎片

内部碎片：分配给作业的存储空间中未被利用的部分。

外部碎片：系统中无法利用的小存储块，比如通过动态内存分配技术从空闲内存区上分配内存后剩下的那部分内存块。

25. 常用的存储保护方法

- (1) 界限寄存器

上下界寄存器方法

基址、限长寄存器方法

- (2) 存储保护键：给每个存储块分配一个单独的存储键，它相当于一把锁。

26. 连续分区分配和非连续分区分配的比较

27. 什么是页表？有什么作用。

为了便于在内存中找到进程的每个页面所对应的物理块，系统为每个进程建立一张页面映射表。页表由页表项组成，页表项有页号和块号组成，根据页表项就可以找到每个页号对于物理内存中物理块的块号。

28. 什么是段寄存器？

29. 进程线程树图

进程树是一个形象化的比喻，比如一个进程启动了一个程序，而启动的这个进程就是原来那个进程的子进程，依此形成的一种树形的结构，我们可以在进程管理器选择结束进程树，就可以结束其子进程和派生的子进程。

30. 作业和进程的区别

一个进程是一个程序对某个数据集的执行过程，是分配资源的基本单位。作业是用户需要计算机完成的某项任务，是要求计算机所做工作的集合。一个作业的完成要经过作业提交、作业收容、作业执行和作业完成 4 个阶段。而进程是对已提交完毕的程序所执行过程的描述，是资源分配的基本单位。

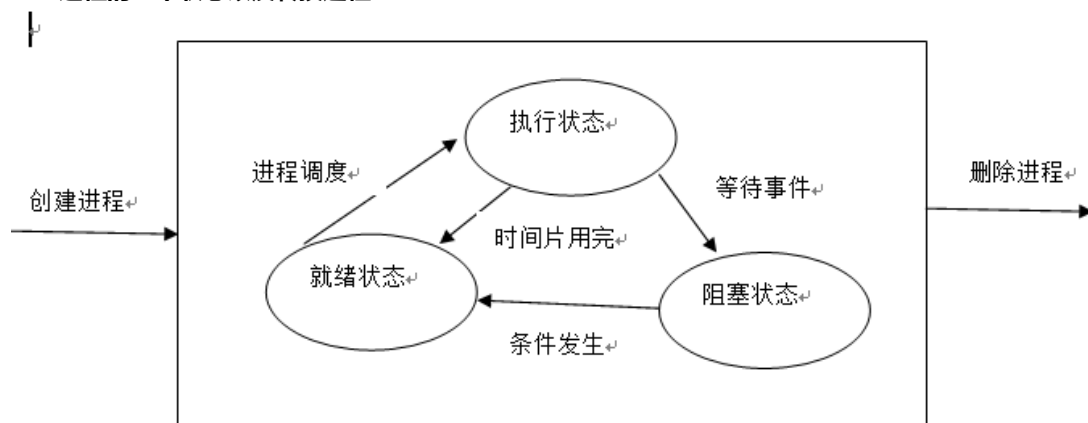
其主要区别如下。

(1) 作业是用户向计算机提交任务的任务实体。在用户向计算机提交作业后，系统将它放入外存中的作业等待队列中等待执行。而进程则是完成用户任务的执行实体，是向系统申请分配资源的基本单位。任一进程，只要它被创建，总有相应的部分存在于内存中。

(2) 一个作业可由多个进程组成，且必须至少由一个进程组成，反过来则不成立。

(3) 作业的概念主要用在批处理系统中，像 UNIX 这样的分时系统中就没有作业的概念。而进程的概念则用在几乎所有的多道程序系统中进程是操作系统进行资源分配的单位。在 Windows 下，进程又被细化为线程，也就是一个进程下有多个能独立运行的更小的单位。

31. 进程的三个状态以及转换过程



32. 进程调度算法有哪些？（再详细了解其中的算法思想）

- ① 先来先服务调度 FCFS
- ② 短作业优先调度 SJF
- ③ 优先级调度 Priority
- ④ 时间片轮转调度 RR

- ⑤ 高响应比优先调度
- ⑥ 多级队列调度
- ⑦ 多级反馈队列调度

33. 死锁

死锁：是指两个或两个以上的进程在执行过程中，因争夺资源而造成的一种互相等待的现象，若无外力作用，它们都将无法推进下去。此时称系统处于死锁状态或系统产生了死锁，这些永远在互相等待的进程称为死锁进程。

死锁原因：

- ① 系统资源不足
- ② 进程推进顺序不当

产生死锁的必要条件：

- ① 互斥条件：指进程对所分配到的资源进行排它性使用，即在一段时间内某资源只由一个进程占用。
- ② 请求和保持条件：指进程已经保持至少一个资源，但又提出了新的资源请求，而该资源已被其它进程占有，此时请求进程阻塞，但又对自己已获得的其它资源保持不放。
- ③ 不剥夺条件：指进程已获得的资源，在未使用完之前，不能被剥夺，只能在使用完时由自己释放。
- ④ 环路等待条件：指在发生死锁时，必然存在一个进程——资源的环形链，即进程集合{P0, P1, P2, ..., Pn}中的 P0 正在等待一个 P1 占用的资源；P1 正在等待 P2 占用的资源，.....，Pn 正在等待已被 P0 占用的资源。

处理死锁的基本方法：

- ① 预防死锁：这是一种较简单和直观的事先预防的方法。方法是通过设置某些限制条件，去破坏产生死锁的四个必要条件中的一个或者几个，来预防发生死锁。预防死锁是一种较易实现的方法，已被广泛使用。但是由于所施加的限制条件往往太严格，可能会导致系统资源利用率和系统吞吐量降低。
- ② 避免死锁：该方法同样属于事先预防的策略，但它并不须事先采取各种限制措施去破坏产生死锁的四个必要条件，而是在资源的动态分配过程中，用某种方法去防止系统进入不安全状态，从而避免发生死锁。
- ③ 检测死锁：这种方法并不须事先采取任何限制性措施，也不必检查系统是否已经进入不安全区，此方法允许系统在运行过程中发生死锁。但可通过系统所设置的检测机构，及时地检测出死锁的发生，并精确地确定与死锁有关的进程和资源，然后采取适当措施，从系统中将已发生的死锁清除掉。
- ④ 解除死锁：这是与检测死锁相配套的一种措施。当检测到系统中已发生死锁时，须将进程从死锁状态中解脱出来。常用的实施方法是撤销或挂起一些进程，以便回收一些资源，再将资源分配给已处于阻塞状态的进程，使之转为就绪状态，以继续运行。

34. 什么是饥饿？与死锁有什么差别？

等待时间给进程推进和响应带来明显影响时成为进程饥饿。

饥饿并不代表系统一点死锁，但至少有一个程序的执行被无限期地推迟。

差别：

- ① 进入饥饿的进程可以只有一个，但是死锁必须大于等于两个；
- ② 出于饥饿状态的进程可以是一个就绪进程，但是死锁状态的进程必定是阻塞进程。

35. 分段和分页的区别

36. 银行家算法

主要思想是避免系统进入不安全状态，在每次进行资源分配时，它首先检查系统是否有足够的资源满足要求，如果有，则先试行分配，并对分配后的新状态进行安全性检查。如果新状态安全，则正式分配上述资源，否则拒绝分配上述资源。这样就保证系统始终处于安全状态，从而避免死锁现象的发生。

37. RAID 磁盘阵列

独立磁盘冗余阵列（RAID, redundant array of independent disks, redundant array of inexpensive disks）是把相同的数据存储在多个硬盘的不同的地方（因此，冗余地）的方法。通过把数据放在多个硬盘上，输入输出操作能以平衡的方式交叠，改良性能。因为多个硬盘增加了平均故障间隔时间（MTBF），储存冗余数据也增加了容错。

38. 控制管理模块是写在哪个文件里的？那个文件叫做什么文件？

1. 连接 2 个局域网需要什么 在那一层

2. TCP 与 UDP 的连接区别及适用情况

3. 路由和交换的区别

4. 七层网络结构

物理层

数据链路层 (PPP、HDLC、CSMA/CD)

网络层(IP、ARP (IP→MAC) /RARP(MAC→IP)、ICMP)

传输层 (TCP、UDP)

会话层

表示层

应用层 (telnet 23、FTP 20<数据>+21<控制>、SMTP 161、DNS、SNMP、DHCP、HTTP 80)

TCP/IP 是四层 (网络接口层、网际层、运输层、应用层)

5. 时分复用的时隙

6. IPV4 和 IPV6 的位数

IPV4 是 32 位; IPV6 是 128 位

7. 单工、半双工、全双工

单工: 又称为单向通信, 即只能有一个方向的通信而没有反方向的交互。例: 无线电广播, 电视广播

半双工: 又称为双向交替通信, 即通信的双方都可以发送信息, 但不能双方同时发送 (当然也就不能同时接受)。

全双工: 又称为双向同时通信, 即通信的双方可以同时发送和接受信息。

备注: 单工只要一条信道, 而半双工和全双工都需要两条信道 (每个方向各一条)。

9. 网络通信过程

10. 简述一下停等协议

由于 IP 层是不可靠的, 因此 TCP 需要采取措施使得传输层之间的通信变得可靠。停止等待协议就是保证可靠传输, 以流量控制为目的的一个协议。其工作原理简单的说就是每发送一个分组就停止发送, 等待对方的确认, 在收到确认后再发送下一个分组, 如果接受方不返回应答, 则发送方必须一直等待。

细节描述: ①如果发送方如果一段时间仍没有收到确认, 就认为刚才发送的分组丢失了, 因而重传前面发送过的分组。

②如果接收方的确认分组丢失或者因其他原因, 收到了重传分组, 则: 丢弃这个重传分组, 并且向发送方发送确认。

11. 应用层有什么协议, 举出两个协议的作用

DNS(域名系统), 将网址解析成 IP 地址。

DHCP (动态主机设置协议), 是一个局域网的网络协议, 使用 UDP 协议工作, 主要有两个用途: 给内部网络或网络服务供应商自动分配 IP 地址, 给用户或者内部网络管理员作为对所有计算机作中央管理的手段。

FTP(文件传输协议), FTP 客户机可以给服务器发出命令来下载文件, 上载文件, 创建或改变服务器上的目录。

TELNET(远程终端协议), 能将用户的击键传到远程主机, 同时也能将远程主机的输出通过 TCP 连接返回到用户屏幕。

12. 数据链路层的作用

数据链路层在物理层提供的服务的基础上向网络层提供服务，将原始的、有差错的物理线路改进成逻辑上无差错的数据链路，从而向网络层提供高质量的服务。功能主要有：①链路管理；②封装成帧；③透明传输；④差错检测

13. 路由协议有哪些

域内：

①RIP（路由信息协议）。距离矢量协议；16跳；收敛速度慢；适用小规模网络；仅和邻居交换信息，且交换的是整个路由表，按固定时间周期性交换；实现简单，开销小；传输层使用UDP；RIP报文限制大小为512字节。

②OSPF（开放最短路由协议）。链路状态协议；基于IP；向自治域中所有路由器发送信息（防洪法），且发送的是链路状态表（LSA），触发更新；更新过程收敛快；根据代价选择最佳路由；支持多路径负载均衡。

域间：

BGP（外部网关协议）是不同AS之间的路由器之间交换路由信息的协议。

14. 频分复用如何避免各路信号间的干扰

15. 简述计算机网络中各层作用 这问题回答之后 老师一直追着网络层问了好多 问得我发懵了。。

16. 列举数据链路层的协议。。2个即可

局域网：

①CSMA/CD（带冲突检测的载波侦听多路访问协议），每一个站在发送数据之前要先检测一下总线上是否有其他计算机在发送数据，如果有，则暂时不发送数据，以免发生冲突；如果没有，则发送。并且计算机在发送数据的同时边检测信道上是否有冲突发生。如果有，则采用截断二进制指数类型退避算法来等待一段随机时间后再次重发。总的来说，可概括为“先听后发，边听边发，冲突停发，随机重发”。

广域网：

①PPP（点对点协议），面向字节；不需要的功能：纠错（PPP协议只负责检错）、流量控制（由TCP负责）、序号（PPP协议是不可靠传输协议，故不需要对帧进行编号）、多点线路（PPP协议是点对点的通信方式）、半双工或单工（PPP只支持全双工链路）。

②HDLC（高级数据链路控制协议），面向比特；标记字段（01111110），地址字段（全1是广播，全0为无效）；控制字段（根据前两位取值可以将HDLC帧划分为三类（无奸细）：信息帧，监督帧，无编号帧），信息字段，帧检验序列FCS（CRC循环冗余码）

17. 网络各层的设备分别是什么

物理层：集线器，中继器（均不能隔离冲突域和广播域）

数据链路层：交换机（转发/学习机制、存储转发、隔离冲突域不能隔离广播域），网桥

网络层：路由器（隔离冲突域和广播域）

18. 什么是滑动窗口协议

滑动窗口协议，是TCP使用的一种流量控制方法。该协议允许发送方在停止并等待确认前可以连续发送多个分组。由于发送方不必每发一个分组就停下来等待确认，因此该协议可以加速数据的传输。

19. PPP 协议

20. 网络按地理范围分

个域网、局域网、城域网、广域网

21. 保护频带 就是插入一些 空白的频段

22. 一个网络安全有哪些方面，还有个 p2p 协议

23. DNS DHCP DNS 倒是记得是让网址转换 IP 地址的 映射 IP 地址 动态主机。。协议

DHCP 是动态分配 ip 吗

24. 流量控制在哪些层实现

传输层 (TCP)、数据链路层 (ARQ)

25. 频分复用 时分复用 波分复用 码分复用

频分复用：给每个信号分配唯一的载波频率并通过单一媒体来传输多个独立信号的方法。

时分复用：把多个信号复用到单个硬件传输信道，它允许每个信号在一个很短的时间使用信道，接着再让下一个信号使用。

波分复用：就是光的频分复用。用一根光纤同时传输多个频率很接近的光载波信号。

码分复用：码分复用是用一组包含互相正交的码字的码组携带多路信号。每一个用户可以在同样的时间使用同样的频带进行通信。由于各用户使用经过特殊挑选的不同码型，各用户之间不会造成干扰，因此这种系统发送的信号有很强的抗干扰能力。

26. CSMA/CD 协议 如果两端同时发送信息会出现什么情况，为什么？

两端都检测到冲突，均停止发送数据，等待一个随机时间再重发。

27. 电路交换，分组交换

见高分笔记 P27

29. 简述下 CSMA/CD 协议的实现原理

30. 描述网络某一层的原理

31. 说一下路由器的原理

传统地，路由器工作于 OSI 七层协议中的第三层，其主要任务是接收来自一个网络接口的数据包，根据其中所含的目的地址，决定转发到下一个目的地址。因此，路由器首先去掉数据包的二层头，取出目的 IP 地址，在转发路由表中查找它对应的下一跳地址，若找到，就在数据包的帧格前添加下一个 MAC 地址，同时 IP 数据包头的 TTL (Time To Live) 域也减一，并重新计算校验和。当数据包被送到输出端口时，它需要按顺序等待，以便被传送到输出链路上。

计算机网络

1. 网络里时延和带宽的概念

时延是指一个报文或分组从一个网络的一端传送到另一个端所需要的时间。它包括了发送时延，传播时延，处理时延，排队时延。(时延=发送时延+传播时延+处理时延+排队时延)一般，发送时延与传播时延是我们主要考虑的。对于报文长度较大的情况，发送时延是主要矛盾；报文长度较小的情况，传播时延是主要矛盾。

带宽又叫频宽，是指在固定的时间可传输的资料数量，亦即在传输管道中可以传递数据的能力。在数字设备中，频宽通常以 bps 表示，即每秒可传输之位数。在模拟设备中，频宽通常以每秒传送周期或赫兹 (Hz)来表示（指通过的频率的宽度）。

2. 网络拥塞

网络中存在太多的数据包导致数据包被延迟或丢失，从而降低了整个网络的传输性能，这种情况叫做拥塞。如果产生网络拥塞，则网络的性能明显下降，整个网络的吞吐量将随着输入负荷的增大而下降。

拥塞控制：可分为闭环控制和开环控制。开环控制是在设计网络时事先将有关拥塞的因素考虑到，力求网络在工作时不产生拥塞。闭环拥塞是基于反馈环路的概念。

拥塞控制的 4 种算法：

- ① 慢开始算法
- ② 拥塞避免算法
- ③ 快重传算法
- ④ 快恢复算法

3. CSMA/CD 的原理（载波侦听多路访问/冲突检测方法）

工作原理：发送数据前，先侦听信道是否空闲，若空闲，则立即发送数据，在发送数据时，边发送边继续侦听，若侦听到冲突，则立即停止发送数据，等待一段随机时间，再重新尝试。**（先听后发，边发边听，冲突停发，随机延迟后重发。）**

4. 三网指哪三网？

电信网，广播电视网，互联网。

5. 组成网络协议的三个要素

- ① 语义：对构成协议元素的含义的解释；
- ② 语法：数据域控制信息的结构和格式；
- ③ 同步：规定事件的执行顺序。

6. 电路交换，报文交换，分组交换之间的区别以及联系

	电路交换	报文交换	分组交换
建立被双方独占物理通路 (建立连接)	需要	不要求	不要求
数据交换单位	比特流	报文	分组
传输方式	比特流直接到达	存储转发	存储转发

每个分组沿着规定路径	是		不是
分组按序到达	是		不是
优点	通信时延小； 实时性强； 适用范围广； 控制简单； 避免冲突。	无需建立连接； 动态分配路线； 可靠性高； 线路利用率高； 可提供多目标服务。	加速传输； 简化存储管理； 减少出错几率和重发数据
缺点	建立连接时间长； 信道利用率低； 缺乏统一标准； 灵活性差。	由于存储转发方式导致转发时延； 对报文大小没有控制，需要较大的存储缓存空间。	存在传输时延； 工作量大，可能出现分组丢失等情况。

7. 分组交换的优点和缺点

与报文交换相比较，分组交换的优点和缺点如下。

优点：

- ① **加速传输**：因为分组交换是逐个传输，所以可以使后一个分组的存储操作与前一个分组的转发操作并行，这种流水线式传输方式减少了报文的传输时间，此外，传输一个分组所需的缓冲区比传输一个报文所需的缓冲区小得多，这样因缓冲区不足而等待发送的几率及等待的时间也必然减少。
- ② **简化了存储管理**：分组的长度固定，相应的缓冲区大小也固定。
- ③ **减少出错几率和重发数据量**：因为分组较短，其出错几率必然减少，所以每次重发的数据量也就大大减少，这样不仅提高了可靠性，也减少了传输时延。

缺点：

- ① **存在传输时延**：尽管与报文交换相比的传输时延较短，但是与电路交换相比仍然存在存储转发时延，而且对其结点交换机的处理能力有更高的要求。
- ② **当分组交换采用数据报服务时**，可能出现失序、丢失或重复分组，分组到达的目的地的结点时，要对分组按编号进行排序工作，工作量较大。

8. 流量控制在哪些层实现？

流量控制就是要控制发送方数据传输的速率，使接收方来得及接收。

数据链路层：相邻结点的流量控制，

传输层：端到端的流量控制，利用滑动窗口机制在 TCP 连接上实现流量控制。

9. CDMA 及原理

码分多路复用，既共享信道的频率，又共享时间，是一种真正的动态复用技术。

原理：

10. 二层交换机和三层交换机的区别

二层交换技术是发展比较成熟，二层交换机属数据链路层设备，可以识别数据包中的 MAC 地址信息，根据 MAC 地址进行转发，并将这些 MAC 地址与对应的端口记录在自己内部的

一个地址表中。

三层交换机就是具有部分路由器功能的交换机

三层交换技术就是二层交换技术 + 三层转发技术。传统交换技术是在 OSI 网络标准模型第二层——数据链路层进行操作的，而三层交换技术是在网络模型中的第三层实现了数据包的高速转发，既可实现网络路由功能，又可根据不同网络状况做到最优网络性能。

11. 频分复用如何避免频带间的干扰，保护频带？

12. 停止等待协议

13. IPv4 地址缺乏的解决办法以及 IPv4 的替代方案以及 IPv4 和 IPv6 如何相互通信？

14. 传统的搜索引擎基本原理，基于内容的搜索？原理和实现？

15. 什么是非对称加密？什么是数据安全的特征？

16. 网络安全有哪些方面

17. 网络的分类

分类	作用范围	使用范围	拓扑结构
	局域网，城域网，广域网，互联网	公用网，专用网	星状，网状，总线，树形，令牌环

18. 有关于信道划分的问题

多路复用：当传输介质的带宽超过了传输单个信号所需的带宽时，通过在一条介质上同时携带多个传输信号的方法来提高传输系统的利用率。

多路复用技术能把多个信号组合在一条物理信道上进行传输，使多个计算机或终端设备共享信道资源，提高信道利用率。

频分多路复用 FDM	所有用户在同样的时间占用不同的带宽资源 为了防止子信道之间的干扰，相邻信道之间要加入保护频带
时分多路复用 TDM, STDM 统计时分复用	所有用户在不用的时间占用相同的频带资源； 统计时分复用是一种动态的时间分配，又称异步时分复用。
波分多路复用 DWDM	每个用户在同一时间使用同样的频带进行通信
码分多路复用	既共享信道频率，又共享时间。

CDMA	
------	--

信道共享技术：

信道共享技术又称为多点接入（multiple access）技术，包括随机接入和受控。

从层次上讲，信道共享是由数据链路层的媒体接入控制 MAC 子层来完成的。总的来说，在计算机网络中使用的信道共享技术可以分为三种，即随机接入、受控接入和信道复用。

- ① 随机接入，特点是所有的用户都可以根据自己的意愿随机地向信道上发送信息。当两个或两个以上的用户都在共享的信道上发送信息的时候，就产生了冲突(collision)，它导致用户的发送失败。随机接入技术主要就是研究解决冲突的网络协议。随机接入实际上就是争用接入，争用胜利者可以暂时占用共享信道来发送信息。随机接入的特点是：站点可随时发送数据，争用信道，易冲突，但能够灵活适应站点数目及其通信量的变化。典型的随机接入技术有 ALOHA、CSMA、CSMA/CD。将会在后面章节中详细介绍。
- ② 受控接入，特点是各个用户不能随意接入信道而必须服从一定的控制。又可分为集中式控制和分散式控制。

集中式控制的主要方法是轮询技术，又分为轮叫轮询和传递轮询，轮叫轮询主机按顺序逐个询问各站是否有数据，传递轮询主机先向某个子站发送轮询信息，若该站完成传输或无数据传输，则向其临站发轮询，所有的站依次处理完后，控制又回到主机。

分散式控制的主要方法有令牌技术，最典型的应用有令牌环网，其原理是网上的各个主机地位平等，没有专门负责信道分配的主机，在环状的网络上有一个特殊的帧，称为令牌，令牌在环网上不断循环传递，只有获得的主机才有权发送数据。

- ③ 信道复用，指多个用户通过复用器(multiplexer)和分用器(demultiplexer)来共享信道，信道复用主要用于将多个低速信号组合为一个混合的高速信号后，在高速信道上传输。其特点是需要附加设备，并集中控制，其接入方法是顺序扫描各个端口，或使用中断技术。

19. 曾经问到过的协议

ICMP、PPP、DNS、DHCP、

20.

21. 计算机网络各层设备及工作原理总结

层	设备	工作原理	其他
物理层	集线器，中继器	中继器：作用就是讲已经衰竭得不完整的信号经过整理，重新产生出完整的信号再继续传送。（放大器放大的是模拟信号）	集线器不能分割冲突域
		集线器 Hub：即多端口的中继器，假设一台 8 个接口的集线器连接 8 台计算机，计算机 1 发送信息给计算机 8，计算机 1 的网卡会将信息通过双绞线传送到集线器中，集线器将信息进行广播，其他 7 个端口的计算机接收到广播时，会对信息进行检查，如果是自己的则接受否则不予理睬。	
数据链路层	交换机，网桥	网桥：网桥至少有两个端口，每个端口与一个网段连接，网桥每从一个端口接收到一个帧，就先暂存在缓存中，若该帧未出现差错，且欲发往的目的地 MAC 地址属于另一个网段（同一网段无需转发，应该丢弃），则通过查找转发表，从相应端口发出。	
		交换机：实质上是一个多端口网桥，每个端口都直接与主机或集线器相连，并且一般都工作在全双工方式。当主机需要通信时，交换机能同时连通许多对的端口，使每一对相互通信的主机都能像独占通信媒体一样，进行无冲突地传输数据，通信完成后断开连接。	
网络层	路由器	路由器的主要作用是转发数据包，通过查询其的路由信息表，将每一个 IP 数据包由一个端口转发到另一个端口。	

设备名称	隔离冲突域	隔离广播域
集线器	✗	✗
中继器	✗	✗
交换机	✓	✗
网桥	✓	✗
路由器	✓	✓

22. 计算机网络协议总结（按层总结：什么层，协议，作用，特点等）

层	传输单位	任务	功能	设备	协议
物理层	比特	透明地传输比特流	为数据端设备提供传送数据通路	集线器/中继器	
数据链路层	帧	将网络层传下来的 IP 数据报组成帧	① 链路的建立，拆除，分离； ② 帧定界和帧同步 ③ 差错检测	交换机/网桥	PPP、HD ARQ
网络层	数据包	① 将传输层传下来的报文段封装成分组 ② 选择适当的路由，是传输层传下来的分组能够交付到目标主机	为传输层提供服务； 组包和拆包； 路由选择； 拥塞控制。	路由器	ICMP ARP RARP IP IGMP
传输层	报文段 TCP,UDP	负责主机两个进程之间的通信	为端到端提供可靠的传输服务； 为端到端连接提供流量控制、差错控制、服务质量等管理服务。		TCP UDP

会话层	应用层		提供系统与用户的接口	文件传输； 访问和管理； 电子邮件服务。		FTP、SMTP、POP3、H DHCP、R BGP
-----	-----	--	------------	----------------------------	--	----------------------------------

计算机组成，微机原理，通信原理

1. RAM 和 ROM 的原理和区别（在基本存储单元上存在本质区别）

RAM:随机通道存储器（random access memory）。一般用来存放各种现场的输入输出数据、中间计算结果、与外存交换的信息以及作为堆栈。一般分为双极型 RAM 和 MOS RAM（又分为静态 SRAM、动态 DRAM）。

双极型 RAM 以晶体管的触发器作为基本存储电路，所以管子多，速度快、功耗大、集成度较低、成本高，一般用于高速计算机或者 cache。

MOS RAM 一般分为静态 SRAM 和动态 DRAM。静态 RAM 用由六管构成的触发器作为基本电路，集成度适中，功耗适中，速度较快，不需要刷新电路（为了维持数据，动态 RAM 中需要一个刷新电路，在短周期内对所有基本存储单元进行充电，维持电平状态，来达到数据的保存）。

动态 DRAM 基本存储电路用单管线路组成（电平靠电容存储，如一般计算机中的内存），集成度高，功耗低，成本比 SRAM 低，由于靠电容的电荷保持电平，所以需要刷新电路（典型要求每隔 1ms 刷新一次）。

总的说 RAM 使用晶体管打开或关闭通往位于交点处的电容器来访问电路。它有如下特性：A、易挥发 B、速度快 C、成本高，即不易制作大容量的 RAM。所以这类存储器一般用在计算机设备的内存部分，而且往往容量不会很大。

ROM: read-only memory，只读存储器。一般用来存放固定的程序以及存放各种常数、函数表。一般分为掩模 ROM、可编程的 ROM（PROM）和可重复擦除的 ROM 三种（按出现时间顺序排列）。

掩模 ROM: 由厂家固化一次性程序，不可擦写。原理是：由晶体管构成，‘0’：存储单元通过晶体管导通连地，上电后该单元视为‘0’电平。‘1’：与‘0’逻辑相反，存储单元没有晶体管接地。出现时间最早。

PROM: 可编程 rom，本身不带有程序，只允许用户进行一次擦写操作，出现时间较晚。

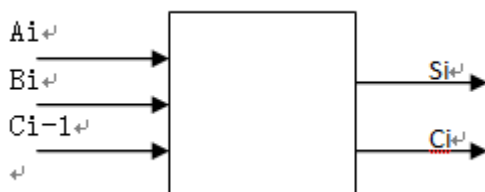
可擦写可编程 rom：有紫外线擦除（EPROM，70 年代初）、电擦除（EEPROM）以及 FLASH（闪存）。他们都是可以多次重复擦写的 ROM。

EPROM 原理是通过击穿绝缘层向浮空栅注入电子进行写操作；利用紫外线使浮空栅电子泄露进行擦操作。

EEPROM: 原理与 EPROM 相似，在绝缘层间加入隧道二极管，于是可以通过电场作用，使浮空栅带上电子或消去。

FLASH：（太高端了，我也不懂 0.0。）

2. 一位全加器的真值表以及逻辑表达式



上图是一位全加器示意图：Ai、Bi 是两个二进制加数，Ci-1 是上一级加法器的进位，Si 是本级一位加法输出，Ci 是加法进位。真值表如下：

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = (A_i \oplus B_i) C_{i-1} + A_i B_i \quad (\oplus \text{表示异或逻辑})$$

3. 什么是芯片组

芯片组 (Chipset) 是构成主板电路的核心。一定意义上讲，它决定了主板的级别和档次。它就是"南桥"和"北桥"的统称，就是把以前复杂的电路和元件最大限度地集成在几颗芯片内的芯片组。芯片组是整个身体的神经，芯片组几乎决定了这块主板的功能，进而影响到整个电脑系统性能的发挥，芯片组是主板的灵魂。芯片组性能的优劣，决定了主板性能的好坏与级别的高低。这是因为目前 CPU 的型号与种类繁多、功能特点不一，如果芯片组不能与 CPU 良好地协同工作，将严重地影响计算机的整体性能甚至不能正常工作。

北桥芯片：提供对 CPU 类型和主频的支持、系统高速缓存的支持、主板的系统总线频率、内存管理（内存类型、容量和性能）、显卡插槽规格，ISA/PCI/AGP 插槽、ECC 纠错等支持；

南桥芯片：提供了对 I/O 的支持，提供对 KBC（键盘控制器）、RTC（实时时钟控制器）、USB（通用串行总线）、Ultra DMA/33(66)EIDE 数据传输方式和 ACPI（高级能源管理）等的支持，以及决定扩展槽的种类与数量、扩展接口的类型和数量（如 USB2.0/1.1，IEEE1394，串口，并口，笔记本的 VGA 输出接口）等；

4. 触发器相关的问题

三类：电位触发、主从触发、边沿触发
等等。。。

5. 中断的软件实行过程

中断源产生中断请求，CPU 一般要先屏蔽该中断源的中断，防止错误的中断嵌套->CPU 对现场进行保存，存储断点程序地址并将当前数据压入栈中->PC 指向对应的中断入口，转入执行中断向量指向的中断服务程序->完成中断响应后，恢复现场，程序回到断点处，将栈中的数据重新读出->重新开中断。

6. 查询传输和中断传输，解释比较

中断传输：中断式传送是指处理器进行的各种处理。执行主程序时，输入设备准备好数据，或者输出设备已把数据输出，向处理器发出中断申请，使处理器中止当前程序的执行，转向中断服务程序，读取输入设备的数据，或者把数据送往输出设备。

无条件传送和查询传输：程序控制传送分为无条件传送（同步传送）和程序查询传送（条件传送、异步传送）。**无条件传送**只对固定的外设（如开关、继电器、7 段显示器等）在规定的时间内用 IN 或 OUT 指令进行信息的传输，实质是用程序定时同步传送数据。**条件查询**的条件是在执行输入或输出指令前，要先查询接口中状态寄存器的状态。输入时，由它只是要输入的数据是否已准备就绪；而输出时，由它只是输出设备是否空闲，由此条件决定执行输入或输出。

可以看出，从 CPU 的角度看，中断的响应相对查询来说是被动的、实时的，如网络的通讯请求，键盘、鼠标的操作等等。查询则一般用于主动的、可控的场合，通常在需要输入/输出数据时，对外设的状态进行查询，以防冲突。而很多外设的速度是很慢的（如打印机），CPU 就会长时间处于等待的状态，无法进行其他操作，这就降低了 CPU 的效率，所以检测到这类设备忙碌的状态时，我们会用设置中断的方法避免这种等待，提高 CPU 的使用率。

中断是难以或者无法预测的，查询的操作是用户在程序中制定的，所以一般是可以控制的。

7. 过程调用具体执行了哪些操作

8. PC 机的端口是同步的还是异步的？什么是异步？

USB 好像是同步的。

PC 机的串口是异步的，即 UART，采用 RS-232 标准。

异步通信：它用一个起始位表示字符的开始，用停止位表示字符的结束来构成一帧。它的传输帧是由起始位、内容和停止位构成的。每个字符可能还包含有奇偶校验位。波特率一般在 50-9600 baud 。数据信号传输速率单位是 字符/秒 或 bps 。

同步通信：为了提高速度，去掉异步通信中使用的起始位和停止位，而在数据块开始前使用一段同步字符指示通信开始，同步通信对整块数据进行传输，速度较快，通常为几十到几百 Kbaud。

9. 控制单元设计·分为组合逻辑和微程序·两者区别·优缺点·

电路的输出仅由该时刻电路的输入决定的电路，称为组合逻辑电路。

上面那句是概念性的东西，具体在数字电路组合逻辑部分。

郎艳峰上的 EDA 技术用的 FPGA 就是组合逻辑控制的一种，不过那个太低端了。。

程序的特点是：为了达到一定目的，程序按照一定的逻辑关系一条一条地执行，程序执行的快慢取决于机器周期的快慢、CPU 的性能以及软件算法的优劣等。

1、**速度上**：组合逻辑电路由用户制定好逻辑关系后，输入信号经过电路的响应，直接得出结果，很大程度上是纯硬件的响应，因此在时刻间隙内可以完成可能需要通过多条程序才能完成的算法，不难看出，组合逻辑电路设计的控制单元比程序控制的单元具有更快的处理速度，而且系统越复杂，组合逻辑在速度上的优势越明显。

2、**复杂度上**：组合逻辑电路是由设计者按照自己的设想将众多逻辑器件组合而成的，对于较为复杂的系统，设计一个逻辑电路往往需要庞大的工作量，检验和仿真等都是费时费力的工程。在这一点上，程控的方法显得方便的多，程序结构的有条不紊是它在设计复杂度上占优势的主要因素。一个无比复杂的逻辑结构可能只需要设计者为 CPU 设计一些精炼的算法就可以实现，在对处理速度没有十分苛刻的要求情况下，通过程序设计会大大地减轻设计者的工作量。

3、**成本**：一般来说，大型可编程逻辑器件的成本要比一些较为高端的处理器或单片机高得多，而且工作量一般比较大，花费的工比较多。

总的来说，工程设计上，要综合实际应用的要求和成本，设计者要尽量选择全方位的设计方案。

10. 关于 IO 接口的，微机原理中的内容，要仔细看下。

接口：接口是位于主机和外设之间的一种缓冲电路，包括了硬件电路和软件控制，现在接口通常为可编程的大规模集成电路芯片。

端口：接口电路通常有三种寄存器：数据输入输出缓冲寄存器、控制寄存器、状态寄存器。这些寄存器称为（数据、控制、状态）端口。对 I/O 设备的访问，就是通过对端口访问实现的，每个端口赋予一个唯一的地址码，称为端口地址。

端口的编址：独立编址和统一编址。统一编址是将端口当做存储器单元，与内存统一编址，内存和端口的操作指令是统一的，所以指令功能强，但是端口占用了存储器的地址空间，使存储器容量减少。

独立编址也称为 I/O 映射编址，即端口和内存分开各自独立地拥有自己的地址空间，端口不占用存储器空间。

80x86 采用独立编址。

I/O 端口地址译码：1、用逻辑门电路进行译码 2、用译码器进行译码 3、比较器译码 4、PROM（可编程 rom）和 GAL（通用逻辑阵列）。

11. 中断（概念，补充：中断可不可以被打断，有哪些情况）

指当出现需要时，CPU 暂时停止当前程序的执行转而执行处理新情况的程序和执行过程。即在程序运行过程中，系统出现了一个必须由 CPU 立即处理的情况，此时，CPU 暂时中止程序的执行转而处理这个新的情况的过程就叫做中断。

在响应中断时，CPU 执行两个连续的中断响应周期，每个周期都输出中断响应信号/INTA，第一个周期 CPU 使 0~15 位的地址总线(8086 有 20 根地址总线)浮空，第二个周期被响应的外设向数据总线输送一个字节的**中断向量号**，而后 CPU 读取中断向量号，转向中断服务程序，响应中断。

有些不可屏蔽中断是可以打断其它中断的。如掉电、复位等。

23. 简述 RAM、ROM、PROM、EPROM、EEPROM 的区别（第一题说过了...）

24. 什么是指令，时钟，总线周期，有什么关系

我们把要求计算机执行的各种操作用命令的形式写下来，就是指令。

2018 年硕士研究生招生（第一志愿）复试安排

根据学校 2018 年硕士研究生招生相关规定，经学院研究决定，现将第一志愿报考我院软件工程硕士的复试安排通知如下。

一、 复试时间及地点

时段	复试内容	复试地点
2018 年 3 月 18 日 上午 8:30—12:00	面试（专业及英语）	科大南区 软件学院大楼各面试室

2018 年 3 月 18 日 下午 13:30—17:30	面试（专业及英语）	科大南区 软件学院大楼各面试室
2018 年 3 月 18 日	综合素质测试机试	科大东区教二楼

具体的复试流程及注意事项另行通知，请及时关注学院主页及招生系统通知。

二、 复试名单及复试分组

所有第一志愿报考我院软件工程硕士（含专业学位和科学学位）并过复试分数线的考生均参加本批次复试，具体名单请考生登录我院招生系统查询。达到我院专业学位分数线未达科学学位分数线的科学学位一志愿考生可直接参加专业学位复试。截止到**2018 年 3 月 16 日下午 17:00**仍未在我院招生系统上注册的第一志愿考生视为放弃复试处理。

具体的复试时段及分组安排将于**3 月 17 日下午**在招生系统上开放查询，考生也可在复试当天在软件学院大楼内查看。

三、 专业方向调整

考虑到复试当天进行的考生动员大会，学院领导将介绍各专业方向和联合培养项目，学院招生系统将在复试当天及后**1 天**开放本批次复试考生的专业选择权限（**2018 年 3 月 20 日下午 5 点**关闭本批次复试考生的专业选择权限）。请各位参加复试的考生及时登录学院招生系统调整自己的专业方向。便于后续的录取工作正常进行。

四、 复试内容与形式

（1）综合素质和专业基础测评（上机）：综合素质测评主要考核考生个性心理特征、诚信状况、意志品质等。专业基础测评主要考察考生有关**C 语言**和数据结构的基础知识及基本运用能力。本部分成绩为“通过/不通过”，只有“通过”考生方可录取。

（2）英语综合面试（满分**50 分**）：重点考核考生的英语听说能力。

（3）专业综合面试（满分**100 分**）：主要考察考生对本学科（专业）理论知识和应用技能的掌握程度，包括考生对本专业基础知识的掌握和理解程度，综合应用所学知识解决实际问题的能力；考察考生在本专业以外的学习、科研、社会实践或实际工作等方面的经历等。

(4) 最终成绩和录取规则：见招生简章

(http://yz.ustc.edu.cn/sszs_2018/index_9.shtml)

五、 复试所需携带材料

1、身份证原件与复印件

2、（应届生）学生证 / （历届生）本科毕业证和学位证的原件与复印件

3、本科学业成绩单原件（教务部门盖章）

软件学院

2018 年 3 月 7 日



18 科软录取信息分析

录取总人数	545 人 220 一志愿 32 校内调剂 293 校外调剂
-------	---

一志愿录取情况（一志愿报考中科大软件学院）

录取人数	录取均分	录取最低分	录取最高分
220 人	340 分	310	398

310-319	41 人
320-329	46 人
330-339	35 人
340-349	21 人
350-359	30 人
360-369	22 人
370-379	18 人
380-389	5 人
390-399	2 人 396 398

校内调剂录取情况（一志愿报考中科大其他学院）			
录取人数	录取均分	录取最低分	录取最高分
32 人	323 分	310	348
310-319	17 人		
320-329	6 人		
330-339	5 人		
340-349	4 人		
调剂来源专业（括号为均分）			
计算机科学与技术	17 人（318）		
电子与通信工程	4 人（333）		
控制工程	4 人（316）		
计算机技术	3 人（331）		
材料学	1 人（342）		
环境工程	1 人（332）		
机械工程	1 人（348）		
网络空间安全	1 人（316）		

校外调剂录取情况			
录取人数	录取均分	录取最低分	录取最高分
293 人	349 分	325	400
320-329	23 人		
330-339	71 人		
340-349	64 人		
350-359	63 人		
360-369	37 人		
370-379	23 人		
380-389	9 人		
390-400	3 人（390、391、400）		

调剂来源—志愿报考学校、报考专业（括号为均分）			
安徽大学 1 人	计院	软件工程	1 人（339）
北京大学共 5 人	信科院	计算机学硕	2 人（349）
	软微	计算机专硕	2 人（337）
	软微	电子通讯工程	1 人（341）
北京航空航天大学 15 人	计院	计算机专硕	1 人（342）
	软院	软件工程	13 人（352）
	自动院	控制工程	1 人（340）
北京交通大学 1 人	计院	计算机学硕	1 人（334）
北京理工大学 1 人	计院	计算机学硕	1 人（335）
北京邮电大学 2 人	计院	计算机学硕	1 人（347）
	软院	软件工程	1 人（329）
东北大学 5 人	计院	计算机学硕	4 人（334）
	计院	计算机专硕	1 人（335）
东南大学 3 人	自动院	控制工程	1 人（343）
	电气院	电气工程	1 人（336）
	信科院	信息与通讯工程	1 人（331）
哈尔滨工业大学 11 人	计院	计算机学硕	1 人（339）
	计院	计算机专硕	3 人（342）
	深圳院	计算机学硕	4 人（338）
	深圳院	计算机专硕	3 人（340）
湖南大学 2 人	电气院	电气工程	1 人（359）
	机械院	机械工程	1 人（375）
华中科技大学 76 人	计院	计算机学硕	8 人（353）
	计院	计算机专硕	41 人（359）
	软院	软件工程	13 人（342）
	电气院	电气工程	3 人（346）
	电信院	信息工程	6 人（350）
	国光院	光电工程	1 人（384）
	自动院	控制工程	4 人（335）
华南理工大学 1 人	电力院	电气工程	1 人（375）
吉林大学 2 人	计院	计算机专硕	2 人（356）
南京航空航天大学 3 人	电气院	电气工程	2 人（362）
	计院	计算机专硕	1 人（330）
清华大学 2 人	计院	计算机学硕	2 人（345）
厦门大学 4 人	计院	计算机学硕	1 人（331）
	计院	计算机专硕	1 人（325）
	智能院	计算机专硕	2 人（366）
上海交通大学 24 人	船舶院	力学	1 人（357）
	电信院	电气工程	2 人（369）
	电信院	计算机学硕	5 人（345）
	电信院	计算机专硕	9 人（344）
	电信院	控制工程	3 人（376）
	电信院	软件工程	2 人（342）

	电信院 网络安全	1 人 (340)
	电信院 仪表工程	1 人 (337)
上海科技大学 1 人	计院 计算机学硕	1 人 (359)
天津大学 2 人	计院 计算机专硕	1 人 (348)
	软院 软件工程	1 人 (347)
同济大学 7 人	电信院 计算机学硕	3 人 (329)
	电信院 计算机专硕	2 人 (326)
	电信院 控制工程	1 人 (336)
	中德院 控制工程	1 人 (332)
武汉大学 31 人	测绘院 测量工程	1 人 (341)
	机械院 控制工程	1 人 (341)
	国软 软件工程	1 人 (333)
	计院 计算机学硕	3 人 (336)
	计院 计算机专硕	25 人 (351)
西安交通大学 2 人	软院 软件工程	1 人 (338)
	机械院 机械工程	1 人 (354)
西北工业大学 4 人	软院 软件工程	1 人 (387)
	计院 计算机专硕	2 人 (358)
	管院 工业工程	1 人 (347)
浙江大学 79 人	电气院 电气工程	1 人 (339)
	机械院 机械工程	1 人 (352)
	计院 计算机学硕	43 人 (349)
	计院 计算机专硕	12 人 (347)
	计院 软件工程	1 人 (333)
	计院 网络安全	2 人 (352)
	软院 软件工程	19 人 (349)
中国科学院大学 5 人	自动院 模式识别	2 人 (339)
	沈计 计算机专硕	2 人 (349)
	计研 计算机学硕	1 人 (375)
中南大学 2 人	计院 计算机学硕	1 人 (334)
	计院 计算机专硕	1 人 (376)
中山大学 1 人	数据院 软件工程	2 人 (352)

2018 年调剂招生简章

中国科学技术大学软件学院是国家教育部和原国家计委首批批准设立的示范性软件学院之一。软件学院秉承我校在创新和进取中凝练出的“红专并进、理实交融”的校训和“我创新故我在”的教育理念，坚持“质量优异、规模适度、夯实基础、注重实践”，依托我校计算机、数学、电子、通信和微电子学科的基础与专业基础课师资力量和丰富教学资源进行办学。同时，学院聘请海内外著名企业和大学的专家、教授授课，课程设置与国际接轨，课程内容面向软件市场和企业，强化实践教学，并提供优良的硬件条件和教学环境。

目前, 软件学院已与微软、IBM、HP、思科、阿尔卡特、朗讯、中国科学院计算所和科大讯飞等许多国内外著名研究院所和 IT 企业建立了长期的合作关系, 以不断改进课程设置和教学内容。学院在北京、上海、深圳、合肥和苏州等地的百余家知名软件企业和日本软件企业建立了实习基地, 并与英国伯明翰大学、美国克莱姆森大学和日本法政大学合作办学, 建立了联合培养软件工程硕士计划, 符合一定条件的在校生可申请参加合作培养计划。学院在培养模式和教学方法上进行了大胆改革, 以强化实践和外语能力, 提高系统开发和管理能力。已逐步形成“高质量、重实践、厚基础、宽视野”的办学特色。

1、招生类型

中国科学技术大学软件学院 2018 年计划通过全国硕士研究生入学统一考试调剂招收全日制软件工程专业硕士研究生 200 名左右。

2、报名条件

报名申请参加调剂的考生必须同时具备以下条件:

- (1) 教育部认可的理工类专业重点大学本科历届或应届毕业生;
- (2) 参加 2017 年 12 月份统考, 报考理工科类专业且专业课一为数学一或数学二, 原则上第一志愿报考单位应该是国内知名高校或中科院科研院所等重要科研机构;
- (3) 统考分数达到软件学院一志愿复试分数线, 政治 50, 英语 50, 专业一 65, 专业二 80, 总分 310。

凡同时满足上述条件的考生均可报名申请调剂。我院将根据教育部 2018 年硕士研究生招生工作精神和相关规定, 以及申请调剂我校软件学院的人数等因素, 制定相应的审核条件, 尽快确定审核通过的复试名单并组织复试。

3、报名方式

符合条件的考生需在中国科学技术大学软件学院招生系统(以下简称“招生系统”)上报名注册相关信息(可从软件学院主页 <http://www.sse.ustc.edu.cn/> 上链接进入“招生系统”)并获取“软件学院报名序号”。

复试时间安排请查阅我院主页和中国科学技术大学软件学院招生系统, 届时我们将在系统上通知, 也会通过其他方式通知到学生本人。

在我院招生系统报名时, 其中须填写专业方向和培养地志愿等信息。专业方向栏内有:

- (1) 软件系统设计专业方向
- (2) 网络与信息安全专业方向
- (3) 嵌入式系统设计专业方向
- (4) 大数据和人工智能专业方向

注意事项:

(1) 复试资格审查及复试安排与通知基于考生在我院招生系统上所注册的相关信息, 而最终录取则基于原始材料以及教育部研招网上的相关信息, **如因考生本人在系统上误填信息导致不符合复试条件而被通知参加复试, 一切责任由考生本人承担:**

(2) 考生报名前应仔细核对本人是否符合报名条件, 凡不符合报考条件或弄虚作假者, 不论何时, 一经查实, 即按有关规定取消复试资格、录取资格、入学资格或

学籍；

（3）教育部的研招网调剂系统开通后考生须尽快在研招网调剂系统上注册并申请调剂至中国科学技术大学软件学院，并及时接受我校在研招网上发布的复试通知；复试结束后我校除在软件学院招生系统上发布复试结果，同时还将在研招网上发布待录取通知。在研招网上收到我校待录取通知的考生须在规定时间内接受我校待录取，否则将视为自动放弃而取消录取资格。

4、复试

复试内容：

（1）综合素质和专业基础测评（上机）：综合素质测评主要考核考生个性心理特征、诚信状况、意志品质等。专业基础测评主要考察考生有关 C 语言和数据结构的基础知识及基本运用能力。本部分成绩为“通过/不通过”，只有“通过”考生方可录取。

（2）英语综合面试（满分 50 分）：重点考核考生的英语听说能力。

（3）专业综合面试（满分 100 分）：主要考察考生对本学科（专业）理论知识和应用技能的掌握程度，包括考生对本专业基础知识的掌握和理解程度，综合应用所学知识解决实际问题的能力；考察考生在本专业以外的学习、科研、社会实践或实际工作等方面的经历等。

复试时须带材料：

（1）身份证原件及复印件；

（2）应届考生需携带加盖学校教务部门（注：仅加盖院系公章的成绩单无效）的本科成绩单原件；历届考生需携带加盖档案管理部门公章的本科成绩单（注：为保持档案的完整性，请使用档案中的本科成绩单的复印件加盖档案管理部门公章，勿直接使用档案中的本科成绩单原件）；

（3）应届考生需携带学生证原件及复印件，历届考生需携带本科毕业证和学位证原件及复印件；

（4）一寸免冠近照 6 张。

注意事项：

（1）参加复试时需按要求携带相关原始材料，否则将会影响录取后续工作；

（2）无论是否录取，上述材料复印件及本科成绩单原件一律不予退还，请各位考生自己做好相关材料备份。

5、录取

我校根据国家 2018 年研究生录取的有关规定，综合评定考生参加全国硕士研究生入学统考成绩和复试成绩择优录取。经我校研究生院审核并报上级主管部门批准后，即可向考生发放我校统一的硕士研究生录取通知书。录取情况将会及时在招生系统和我院主页上公布。

6、培养与管理

学制年限：学制为 2.5-3 年（计划内联合培养项目的学制原则上 3 年）。我院采用学分制，允许学生在不超过 5 年内修满学分（40 个学分），完成学业。

培养地点：苏州•中国科学技术大学软件学院（苏州工业园区独墅湖高等教育区仁爱路 166 号，在中国科学技术大学苏州研究院内）

合肥•中国科学技术大学软件学院（合肥市徽州大道 1129 号，中科大南校区）

培养模式: 全日制培养, 其中授课与毕业论文答辩时间为一年半, 实习和毕业论文(设计)时间为一年。

7、学历与学位证书

在规定的年限内修满规定学分, 成绩合格, 通过学位论文答辩, 可申请工程硕士专业学位, 经校学位委员会审核批准后, 将颁发国家统一印制的硕士研究生毕业证书和软件工程专业硕士学位证书。毕业生就业时, 采取毕业研究生与用人单位“双向选择”的方式, 落实就业去向。学校及安徽省毕业生调配部门负责办理派遣等相关手续。

8、学费及住宿费

学费总计 40000 元(不含教材费、食宿费等其他费用)。分两次分别在第一、二学年开学前缴清。(教材费、食宿费等其他费用自理)。

住宿费按照中国科学技术大学或苏州独墅湖高等教育区统一标准收费。中国科学技术大学住宿费每学年约 1000 元; 苏州独墅湖高等教育区住宿费每学年约 3000 元(双人间, 有卫生间、热水器、空调等)。

9、开学时间

开学时间按学校统一规定, 预计为 2018 年 9 月初, 具体时间见录取通知书。为提高部分新生的计算机基础理论知识和实践能力, 学院将在 2018 年 7 月底或 8 月初开始为部分新生开设四门左右本科计算机基础理论课程, 具体事宜将于 2018 年新生正式录取后另行通知。

10、联系方式

招生办办公地点:

中国科学技术大学软件学院招生办公室(合肥)

(合肥市徽州大道 1129 号中科大南校区软件学院教学大楼四层)

中国科学技术大学软件学院学生工作部(苏州)

(苏州工业园区独墅湖高等教育区仁爱路 188 号中科大苏州研究院唯真楼 503

室)

招生政策咨询电话: 0551-63492278

传真: 0551-63492016

E-mail: zsse@ustc.edu.cn

网址: <http://www.sse.ustc.edu.cn/>

中国科学技术大学软件学院 2018 软件工程硕士

3 月 18 日第一志愿复试注意事项

考生需在复试以及录取过程中注意并办理以下事宜:

1、复试程序: 所有考生务必完成以下前 6 项内容(详见该页背面复试程序)

(1) 参加考生动员大会: 软件学院大楼 103 教室;

(2) 领取材料、相关咨询等: 软件学院大楼一楼大厅;

- (3) 材料审核、签署协议：软件学院大楼 101，102 教室；
- (4) 专业综合面试：软件学院大楼 5 楼各面试室；
- (5) 英语听说面试：软件学院大楼 4 楼各面试室；
- (6) 综合素质测试和专业测试（机考）：科大东区教二楼 8 楼，按规定时间提前半小时到达验身份证入场；
- (7) 填写 2018 级联合培养项目报名表（仅限报名申请的考生）：软件学院大楼一楼大厅。

2、面试分组序号查询：考生可于复试前一天下午开始在软件学院招生系统，或复试当天在软件学院大门对面墙上现场查询面试分组序号。

3、录取过程：按“计划录取→拟录取→正式录取”的流程确定正式录取名单：

- (1) 预计在复试后 3 天内软件学院招生系统上公布本批次复试的计划录取名单；
- (2) 拟录取的考生经审核以及报学校上级主管部门审批通过后，成为 18 级正式录取研究生。预计 6 月中下旬（我校本学期结束前）由我校研究生院统一寄发录取通知书。

附：本次复试所发材料

表 1《2018 中国科大攻读硕士学位研究生情况表》：该表在“材料审核处”（软件学院大楼 101，102 教室）分组领取。考生对表上已有信息进行核对并补充完成该表，连同相关附件在专业综合面试时交给面试老师；

表 2《自筹经费协议书》：该表在“材料审核处”（软件学院大楼 101，102 教室）当场签署，勿带走；

表 3《复试注意事项与复试流程》：即本表，在“领取材料处”（软件学院大楼一楼大厅）领取；

表 4《2018 级联合培养项目简介》：在“领取材料处”（软件学院大楼一楼大厅）领取；

2018 年 3 月 18 日第一志愿复试流程

分项	时间安排	地点	内容	具体事项
1	3 月 18 日 7:20 开始	软件学院大楼一楼大厅	领取材料 接受咨询	需领取： 表 3《复试注意事项与复试流程》（本表） 表 4《2018 级联合培养项目简介》 （仅限 报名“所系结合”申请联合培养的考生）
			递交联合培养项目申请 8:00-11:30	需填写： 表 5 《2018 级联合培养项目报名表》 注：本表为联合培养项目报名表，其他国际合作项目具体申请时间及报名方式将另行通知。

2	3月18日 7:40-8:10	软件学院大楼 103 教室	考生动员大会	学院领导介绍学院以及本次复试基本情况
3 注： 此部分需按流程依次进行	3月18日 8:10-10:30	软件学院大楼 101 教室 102 教室	材料审核 签署协议 (按分组进行)	<u>需交验：</u> ①身份证原件与复印件 ②(应届生)学生证/(历届生)本科毕业证和学位证的原件与复印件 ③本科学业成绩单原件 需领取： 表1《2018 中国科大攻读硕士学位研究生情况表》(装订好相关附件) 需签署： 表2《自筹经费攻读硕士学位研究生协议书》 (此协议当场签署，勿带走)
	3月18日 8:30-12:00	4-5 楼 2、3 号楼梯口	身份认证	需交验：身份证原件；
	13:30-17:00	5 楼 各面试室	专业面试 (按分组进行)	需交给专业面试组老师： 已核对并补充完成的《2018 中国科学技术大学攻读硕士学位研究生情况表》及附件；
4	3月18日 8:30-12:00 13:30-17:00	3-4 楼 1 号楼梯口	身份认证	需交验：身份证原件；
		4 楼 各面试室	英语面试 (不按分组进行)	按身份认证处的安排进入英语面试室进行面试 (无需材料审核可直接凭身份证参加，进入面试室后须在签到表上签到)
5	3月18日 16:00-17:30	科大东区 教二楼8楼 (考场分布见现场通知)	综合素质测试 和专业测试 (机试)	仅限当天上午复试单元的考生参加 需携带：身份证原件； (具体复试名单复试当天可在软件学院一楼大厅或科大东区教二楼8楼查看， 请按规定时间提前半小时到达验证入场)
	3月18日 19:00-20:30			仅限当天下午复试单元的考生参加

备注：

- 第3项的专业面试必须按照“材料审核→身份认证→专业面试”的顺序进行；
- 第3、4项无先后顺序，第4项的英语面试无需材料审核即可直接凭身份证参加，考生可根据现场和自身实际情况自行安排先参加哪一项面试；
- 材料审核必须在当天上午完成，安排在上午复试单元的考生优先进行；
- 专业面试和英语面试均为安排在本复试单元(以半天为一个复试单元)的考生优先进行。闲暇时方可安排其他复试单元的考生进行。

3月24-25日调剂复试注意事项

考生需在复试以及录取过程中注意并办理以下事宜:

1、所有调剂考生必须上教育部研招网调剂系统三次操作:

- (1) 第一次操作: 在研招网上注册并申请调剂至我院。截止时间为 3 月 25 日 14:00;
- (2) 第二次操作: 我院发复试通知, 考生须接受我院的复试通知。截止时间见研招网短信提示;
- (3) 第三次操作 (仅限被我校计划录取的考生): 我院发待录取通知 (预计将于 3 月 27 日发布), 考生须接受待录取。截止时间见研招网短信提示。

2、复试程序: 所有考生务必完成以下前 6 项内容 (详见背面复试程序)

注意: 所有考生必须在安排的复试单元当天完成以下所有工作 (如, 复试单元为 3 月 24 日上午和 3 月 24 日下午的所有考生必须在 24 日当天完成以下事项, 24 日当天不接待复试单元为 25 日的考生)

- (1) 参加考生动员大会: 软件学院大楼 103 教室;
- (2) 领取材料、接受咨询等: 软件学院大楼一楼大厅;
- (3) 材料审核、签署协议: 软件学院大楼 101, 102 教室;
- (4) 专业综合面试: 软件学院大楼 5 楼各面试室;
- (5) 英语听说面试: 软件学院大楼 4 楼各面试室;
- (6) 综合素质测试和专业测试 (机考): 科大东区教二楼 8 楼, 按规定时间提前半小时到达验证入场;
- (7) 填写 2018 级联合培养项目报名表 (仅限报名申请的考生): 软件学院大楼一楼大厅。

3、面试分组序号查询: 考生可于复试前一天下午在软件学院招生系统, 或复试当天在软件学院大门对面墙上现场查询面试分组序号。

4、录取过程: 按“计划录取→拟录取→正式录取”的流程确定正式录取名单:

- (1) 预计于 3 月 27 日左右在软件学院招生系统上公布本批次复试的计划录取名单, 同时在研招网调剂系统上发布待录取通知;
- (2) 调剂考生在规定时间内 (研招网短信提醒) 在研招网上操作接受我校待录取, 届时不操作“接受待录取”的考生, 将被取消待录取资格;
- (3) 拟录取的考生经审核以及报学校上级主管部门审批通过后, 成为 18 级正式录取研究生。预计 6 月中下旬 (我校本学期结束前) 由我校研究生院统一寄发录取通知书。

5、专业调整: 考虑到复试当天进行的考生动员大会上, 学院领导将介绍各专业方向和联合培养项目, 学院招生系统将在复试当天及后 1 天重新开放本批次复试考生的专业选择权限 (2018 年 3 月 26 日下午 5 点关闭本批次复试考生的专业选择权限)。请各位参加复试的考生及时登

录学院招生系统调整自己的专业方向。便于后续的录取工作正常进行。

附：本次复试所发材料

表 1《2018 中国科大攻读硕士学位研究生情况表》：该表在“材料审核处”（软件学院大楼 101，102 教室）分组领取。考生对表上已有信息进行核对并补充完成该表，连同相关附件在专业综合面试时交给面试老师；

表 2《自筹经费协议书》：该表在“材料审核处”（软件学院大楼 101，102 教室）当场签署，勿带走；

表 3《复试注意事项与复试流程》：即本表，在“领取材料处”（软件学院大楼一楼大厅）领取；

表 4《2018 级联合培养项目简介》：在“领取材料处”（软件学院大楼一楼大厅）领取；

表 5《2018 级联合培养项目报名表》：该表在一楼大厅“材料领取处”领取和提交，仅限报名申请“所系结合”联合培养的考生填写。

2018 年 3 月 24-25 日校外调剂复试流程

注意：所有考生必须在安排的复试单元当天完成以下所有工作（如，复试单元为 3 月 24 日上午和 3 月 24 日下午的所有考生必须在 24 日当天完成以下事项，24 日当天不接待复试单元为 25 日的考生）

分项	时间安排	地点	内容	具体事项
1	3 月 24 日（25 日） 7:20 开始	软件学院大楼 一楼大厅	领取材料 接受咨询	需领取： 表 3《复试注意事项与复试流程》（本表） 表 4《2018 级联合培养项目简介》
			递交联合培养项目申请	（仅限 报名申请联合培养的考生）需填写：表 5 《2018 级联合培养项目报名表》 注：本表为联合培养项目报名表，其他国际合作项目 具体申请时间及报名方式将另行通知。
2	3 月 24 日（25 日） 7:40-8:10	软件学院大楼 103 教室	考生动员大会	学院领导介绍学院以及本次复试基本情况
3 注： 此部分 需按流 程依 次进 行	3 月 24 日（25 日） 8:10-10:30	软件学院大楼 101 教室 102 教室	材料审核 签署协议 （按分组进行）	需交验： ①身份证原件与复印件 ②（应届生）学生证/（历届生）本科毕业证和学位 证的原件与复印件 ③本科学业成绩单原件 需领取： 表 1《2018 中国科大攻读硕士学位研究生情况表》 （装订好相关附件） 需签署： 表 2《自筹经费攻读硕士学位研究生协议书》 （此协议当场签署，勿带走）
	3 月 24 日（25 日） 8:30-12:00 13:30-17:00	4-5 楼 2、3 号楼梯口	身份认证	需交验：身份证原件；
		5 楼 各面试室	专业面试 （按分组进行）	需交给专业面试组老师： 已核对并补充完成的《2018 中国科学技术大学攻读 硕士学位研究生情况表》及附件；

4	3月24日（25日） 8:30-12:00 13:30-17:00	3-4楼 1号楼梯口	身份认证	需交验：身份证原件；	
		4楼 各面试室	英语面试 （不按分组进行）	按身份认证处的安排进入英语面试室进行面试 （无需材料审核可直接凭身份证参加，进入面试室后 须在签到表上签到）	
5	3月24日 18:00-19:30 3月25日 16:00-17:30	科大东区 教二楼8楼 （考场分布见 现场通知）	综合素质测试 和专业测试 （机试）	仅限当天上午 复试单元的考 生参加	需携带：身份证原件； （具体复试名单复试当天可在软件学 院一楼大厅或科大东区教二楼8楼查 看，请按规定时间提前半小时到达验 证入场）
	3月24日 20:00-21:30 3月25日 19:00-20:30			仅限当天下午 复试单元的考 生参加	
6			研招网操作	上研招网注册的截止时间为3月25日14:00 接受我院复试通知和接受我院待录取通知的截止时间 见研招网短信提示（预计3月27日左右）	

备注：

- （1）第3项的专业面试必须按照“材料审核—>身份认证—>专业面试”的顺序进行；
- （2）第3、4项无先后顺序，第4项的英语面试无需材料审核即可直接凭身份证参加，考生可根据现场和自身实际情况自行安排先参加哪一项面试；
- （3）材料审核必须在当天上午完成，安排在当天上午复试单元的考生优先进行；
- （4）专业面试和英语面试均为安排在本复试单元（以半天为一个复试单元）的考生优先进行。闲暇时方可安排其他复试单元的考生进行。