

武汉工程大学 计算机科学与工程学院

《面向对象程序设计》课程考试大纲

一、课程名称：面向对象程序设计

二、适用专业：2018 级计算机类各专业

三、开课学期：2019-2020 学年第 1 学期

四、考试方式：笔试、闭卷

五、组卷类型：题库抽题

六、考试时间：120 分钟

七、命题期望：72 分

八、考试题型：

- 1、判断题：共 10 分，10 个小题。每小题 1 分。
- 2、单选题：共 20 分，10 个小题。每小题 2 分。
- 3、程序分析：共 30 分，6 个小题。每小题 5 分。
- 4、程序设计：共 40 分，4 个小题。每小题 10 分。

九、知识点及考试要求：

第一章 C++语言基础

1、熟练掌握 C++语言的基础知识，例如 cin 输入/cout 输出，基本数据类型，常用运算及表达式求值，语句与控制结构等。

2、理解引用类型的特点。熟练掌握引用类型传递函数（输入和输出）参数。理解函数返回引用的用法。

3、理解函数原型、函数定义、函数调用等三个概念及其相互关系。

4、理解内联函数的概念及运行机制。熟练掌握函数参数默认值的设置。熟练定义重载函数。

5、会分析函数运行的结果。熟练定义函数描述算法过程。

6、理解数组的区间表示形式。掌握 vector 类的基本操作。会定义 vector 容器对象，会用 push_back 向容器中加入元素，并通过下标运算访问元素，通过 size 获取元素个数等。

7、掌握 string 类的基本操作，会定义字符串对象，通过 getline 读取字符串，通过下标运算访问字符元素，获取字符元素的个数，通过+连接字符串，通过关系运算比较字符串等。

第二章 类与对象的定义

8、理解面向对象方法的基本概念，例如对象、类、属性、方法、消息等。理解面向对象的三大特性：封装性、继承性、多态性。

9、熟练定义类，并设置恰当和正确的数据成员和成员函数。能够在类中提供操作实现简单的计算。

10、熟练定义构造函数、析构函数，以及默认构造函数、复制构造函数和赋值运算符函数。能够正确分析对象构造和析构的过程。

11、能够正确描述类之间的整体与部分的关系。理解成员初始化列表语法。能够运用类的复合关系解决简单的问题。

第三章 类的几个主题

12、理解 `this` 指针的用法。会通过 `this` 指针访问成员。会在赋值运算符函数中通过 `this` 返回作为整体的对象。

13、理解 `const` 对象的特点。理解 `const` 引用作函数参数的用法。重点掌握 `const` 成员函数的特点及其用法。

14、会运用 `new/delete` 运算符生成动态数组。会运用 `new/delete` 生成/释放堆对象，并注意区别堆对象与栈对象的构造析构过程的不同。

15、会定义 `friend` 函数作简单的计算。

第四章 运算符重载

16、理解运算符重载的概念、语法规则和两种形式。注意区别成员函数形式与友元函数形式的运算符函数。

17、会重载算术运算符、赋值运算符、关系运算符、流插入运算符。

第五章 模板、第六章 标准模板库 STL

18、会运用常用的 STL 算法，例如 `for_each`, `find`, `sort`, `count/count_if`, `max_element/min_element` 等。

19、熟练定义函数模板实现基本的数组算法。例如线性查找，二分查找，排序，最大最小值，计数，替换，逆转等。以及其它较为简单的算法过程。

第七章 继承与派生

20、理解继承机制的特点。会运用继承描述概念之间的层次关系。能够正确定义类之间的继承关系。

21、理解 `public` 继承时派生类的对象和成员函数对基类和派生类成员的访问规则。

22、熟练掌握派生类构造函数的定义。理解派生类对象构造/析构的过程。

第八章 虚函数与多态性

23、理解虚函数的概念以及定义语法。

24、理解虚函数运行过程，会分析虚函数的运行结果。

25、理解抽象基类与纯虚函数的概念以及定义语法。

26、能够定义虚函数实现多态性的计算。

第九章 C++的 I/O 流

27、理解 C++的 I/O 流库的基本概念、特点及类层次结构。

28、熟练运用标准 I/O 流对象的常用运算符(流插入运算符/流提取运算符)和成员函数 (get (getline) /put, read/write 等)。

29、掌握格式化 I/O 的常用格式标志位、成员函数和流操纵算子。

30、会运用文件 I/O 流的常用操作 (打开、读写、关闭) 实现简单的功能。

第十章 异常处理

本章内容不作考试要求。

十、重点内容及建议：

1、重点章节：第一章，第二章，第七章，第八章。

执笔人：张俊

教研室主任：吕涛

分管教学副院长：张俊

2019.8.12