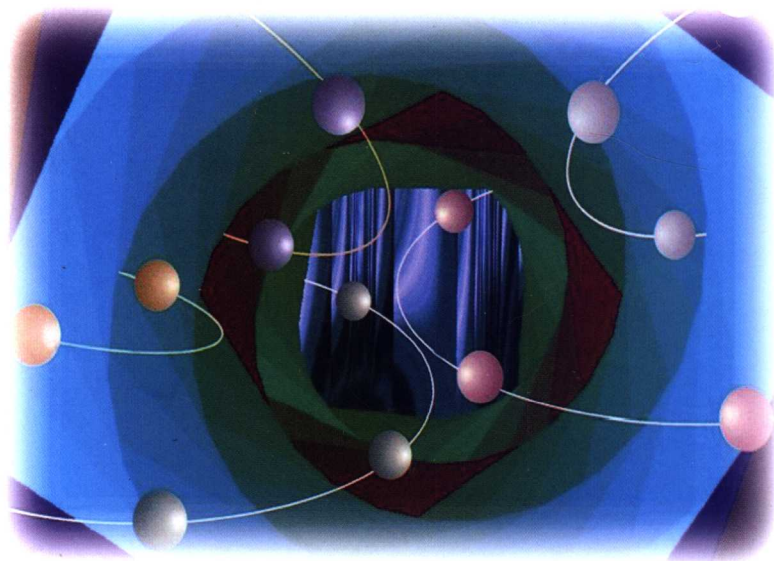




清华大学计算机基础教育课程系列教材

# C++ 语言程序设计 (第3版) 教师用书

郑 莉 张瑞丰 李 莉 编著



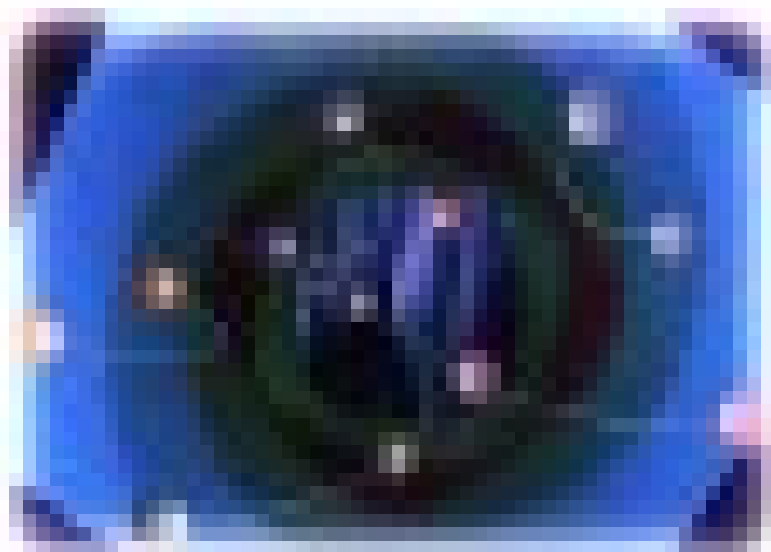
清华大学出版社

清华大学出版社“十二五”普通高等教育本科国家级规划教材

# C++ 语言程序设计

## (第3版) 教师用书

· · · · ·



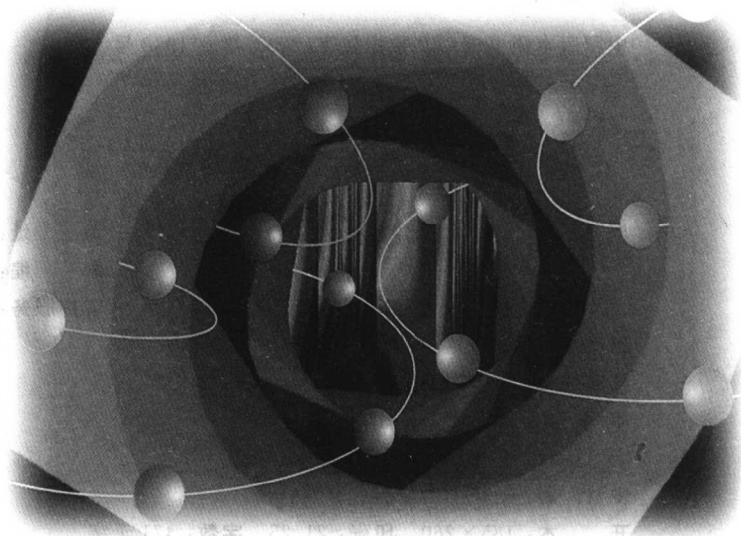
清华大学出版社



清华大学计算机基础教育课程系列教材

# C++语言程序设计 (第3版) 教师用书

郑莉 张瑞丰 李莉 编著



清华大学出版社  
北京

## 内 容 简 介

本书与《C++ 语言程序设计(第3版)》配套,内容包括每章教学要点、讲稿及备注(对每页PPT讲稿进行详细注释)、补充习题与解答。其中“教学要点”、“讲稿及备注”有助于新开课的教师尽快熟悉教学内容、掌握教学方法。由于配套的学生用书给出了全部习题解答,不利于老师通过作业考核平时成绩,因此本书提供了“补充习题与解答”,便于老师布置课后作业。

本书可作为从事大学本科计算机教学的教师使用,也可供其他计算机语言程序设计者参考。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

C++ 语言程序设计(第3版)教师用书/郑莉,张瑞丰,李莉编著. —北京:清华大学出版社,2005.5  
(清华大学计算机基础教育课程系列教材)

ISBN 7-302-10498-0

I. C… II. ①郑… ②张… ③李… III. C语言—程序设计—高等学校—教学参考资料  
IV. TP312

中国版本图书馆CIP数据核字(2005)第011988号

出版者:清华大学出版社

<http://www.tup.com.cn>

社总机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:张 龙

文稿编辑:霍志国

印刷者:北京密云胶印厂

装订者:北京市密云县京文制本装订厂

发行者:新华书店总店北京发行所

开 本:185×260 印张:21.25 字数:471千字

版 次:2005年5月第1版 2005年5月第1次印刷

书 号:ISBN 7-302-10498-0/TP·7124

印 数:1~2000

定 价:35.00元(含光盘)

# 序

计算机科学技术的发展不仅极大地促进了整个科学技术的发展,而且明显地加快了经济信息化和社会信息化的进程。因此,计算机教育在各国备受重视,计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

清华大学自 1990 年开始将计算机教学纳入基础课的范畴,作为校重点课程进行建设和管理,并按照“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的课程体系组织教学:

第一层次“计算机文化基础”的教学目的是培养学生掌握在未来信息化社会里更好地学习、工作和生活所必须具备的计算机基础知识和基本操作技能,并进行计算机文化道德规范教育。

第二层次“计算机技术基础”是讲授计算机软硬件的基础知识、基本技术与方法,从而为学生进一步学习计算机的后续课程,并利用计算机解决本专业及相关领域中的问题打下必要的基础。

第三层次“计算机应用基础”则是讲解计算机应用中带有基础性、普遍性的知识,讲解计算机应用与开发中的基本技术、工具与环境。

以上述课程体系为依据,设计了计算机基础教育系列课程。随着计算机技术的飞速发展,计算机教学的内容与方法也在不断更新。近几年来,清华大学不断丰富和完善教学内容,在有关课程中先后引入了面向对象技术、多媒体技术、Internet 与互联网技术等。与此同时,在教材与 CAI 课件建设、网络化的教学环境建设等方面也正在大力开展工作,并积极探索适应 21 世纪人才培养的教学模式。

为进一步加强计算机基础教学工作,适应高校正在开展的课程体系与教学内容的改革,及时反映清华大学计算机基础教学的成果,加强与兄弟院校的交流,清华大学在原有工作的基础上,重新规划了“清华大学计算机基础教育课程系列教材”。

该系列教材有如下几个特色:

1. 自成体系:该系列教材覆盖了计算机基础教学三个层次的教学内容。其中既包括所有大学生都必须掌握的计算机文化基础,也包括适用于各专业的软硬件基础知识;既包括基本概念、方法与规范,也包括计算机应用开发的工具与环境。

2. 内容先进:该系列教材注重将计算机技术的最新发展适当地引入教学中,保持教学内容的先进性。例如,系列教材中包括了面向对象与可视化编程、多媒体技术与应用、Internet 与互联网技术、大型数据库技术等。

3. 适应面广: 该系列教材照顾了理、工、文等各种类型专业的教学要求。

4. 立体配套: 为适应教学模式、教学方法和手段的改革, 该系列教材中多数都配有习题集和实验指导、多媒体电子教案, 有的还配有 CAI 课件以及相应的网络教学资源。

本系列教材源于清华大学计算机基础教育的教学实践, 凝聚了工作在第一线的任课教师的教学经验与科研成果。我希望本系列教材不断完善, 不断更新, 为我国高校计算机基础教育做出新的贡献。



# 前 言

目前,“C++ 语言程序设计”已经被许多大学列为程序设计基础课程,除了信息类专业外,很多其他工科专业也开设了这门课程。《C++ 语言程序设计》系列教材目前已经出版了三版。为了帮助使用本教材授课的教师,尤其是新教师,我们编写了这本教师用书。

本书与《C++ 语言程序设计(第3版)》配套,内容包括每章教学要点、讲稿及备注(对每页PPT讲稿进行详细注释)、补充习题与解答。其中“教学要点”、“讲稿及备注”有助于新开课的教师尽快熟悉教学内容、掌握教学方法。由于配套的学生用书给出了全部习题解答,不利于老师通过作业考核平时成绩,故本书提供了“补充习题与解答”,便于老师布置课后作业。

本书的配套光盘包括PPT格式的讲稿、补充习题源代码、例题源代码、实验参考程序。

本系列教材全部内容如下:

(1) 主教材:《C++ 语言程序设计(第3版)》。

(2) 《C++ 语言程序设计(第3版)学生用书》

包括学习指南、每章要点导读、实验指导和习题解答。

(3) 《C++ 语言程序设计(第3版)教师用书》

每章教学要点、讲稿注释、补充习题与解答。

(4) 配套资源

《C++ 语言程序设计(第3版)》电子教案

《C++ 语言程序设计(第3版)》网络视频课件

《C++ 语言程序设计(第3版)》音频课件

(5) 开放的教学网站 (<http://learn.tsinghua.edu.cn>)

网站对全球开放,有“在线讲堂”、“答疑教室”、“师生讨论”等栏目,在线演播配音课件,提供教学资源下载,组织教学活动。在“本学期课程”栏目中查找郑莉的“C++ 语言程序设计”(如果首页未列出,则单击“更多”链接),找到后单击课程名,然后以用户名“GUEST”、密码“guest”登录,即可阅读、下载教学资源。在作者不授课的学期,

读者可以查找以往课程,下载学习资源。

感谢您选择使用本书,欢迎您对本书内容提出意见和建议,我们将不胜感激。作者的电子邮件地址:zhengli@mail.tsinghua.edu.cn,来信标题请包含“C++ teacher”。

作 者

2005 年 3 月于清华大学



# 目 录

<b>第 1 章 绪论</b> .....	1
1.1 教学要点 .....	1
1.2 讲稿 .....	1
1.3 补充习题.....	14
<b>第 2 章 C++ 简单程序设计</b> .....	15
2.1 教学要点.....	15
2.2 讲稿.....	15
2.3 补充习题与解答.....	40
<b>第 3 章 函数</b> .....	50
3.1 教学要点.....	50
3.2 讲稿.....	51
3.3 补充习题与解答.....	68
<b>第 4 章 类与对象</b> .....	76
4.1 教学要点.....	76
4.2 讲稿.....	77
4.3 补充习题与解答.....	97
<b>第 5 章 C++ 程序的结构</b> .....	106
5.1 教学要点 .....	106
5.2 讲稿 .....	106
5.3 补充习题与解答 .....	123
<b>第 6 章 数组、指针与字符串</b> .....	131
6.1 教学要点 .....	131
6.2 讲稿 .....	132
6.3 补充习题与解答 .....	164

---

<b>第7章 继承与派生</b> .....	179
7.1 教学要点 .....	179
7.2 讲稿 .....	180
7.3 补充习题与解答 .....	201
<b>第8章 多态性</b> .....	209
8.1 教学要点 .....	209
8.2 讲稿 .....	210
8.3 补充习题与解答 .....	222
<b>第9章 群体类和群体数据的组织</b> .....	235
9.1 教学要点 .....	235
9.2 讲稿 .....	236
9.3 补充习题与解答 .....	261
<b>第10章 泛型程序设计与C++标准模板库</b> .....	268
10.1 教学要点.....	268
10.2 讲稿.....	269
10.3 补充习题与解答.....	279
<b>第11章 流类库与输入输出</b> .....	296
11.1 教学要点.....	296
11.2 讲稿.....	297
11.3 补充习题与解答.....	305
<b>第12章 异常处理</b> .....	312
12.1 教学要点.....	312
12.2 讲稿.....	313
12.3 补充习题与解答.....	318
<b>第13章 MFC库与Windows程序开发概述</b> .....	323
13.1 教学要点.....	323
13.2 讲稿.....	323
13.3 补充习题.....	327

# 第 1 章

## 绪论

### 1.1 教学要点

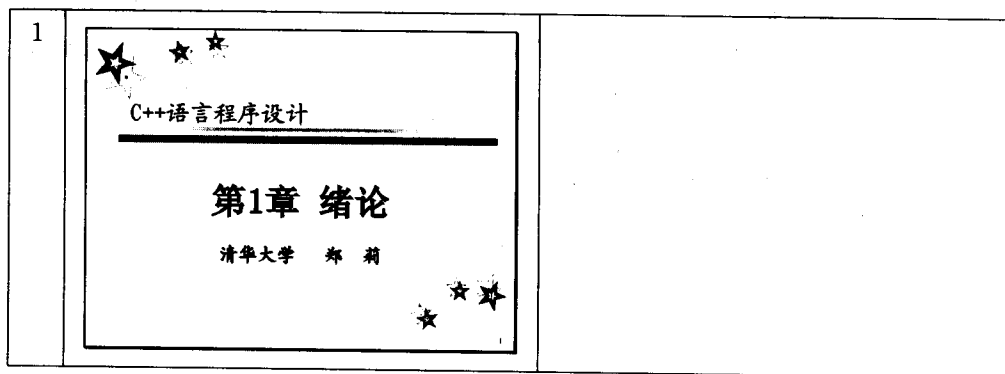
第 1 章简要介绍了面向对象的程序设计思想和程序设计基础知识。这部分内容对于全书的学习是非常重要的,在教学中不可忽略。教师应在大课上详细讲解,不能仅由学生自学,因为没有学过面向对象程序设计的学生自学本章内容会感到困难。

首先应该向学生阐明:语言是思维的工具,思维是通过语言来表述的。语法简单、词汇较少的语言,本身容易掌握,但是表现力比较差,用它表达思想就很难。语法相对复杂、词汇量大的语言,由于其表现手段多样,表现力也就比较强,用它表达思想就比较容易。因此复杂的语法和词汇不是学习中的障碍,而是表达思维的助手。面向过程的语言只能支持较简单的思维模式,而人类的思维非常复杂,因而人们采用面向过程的程序设计语言描述问题和解决问题的方法处理问题就感觉很困难。面向对象语言的表达能力相对较强,更接近人类的思维模式,因而采用面向对象的语言来描述问题和解决问题的方法处理问题就比较容易。关于计算机语言的发展和演变、计算机语言与人类语言的关系等方面内容,《面向对象的系统分析》一书讲得很透彻。

对于什么是对象,如何对同类型对象进行抽象,并将抽象的结果封装成为类,下面要进行简单通俗地介绍。

建议学时: 1 学时。

### 1.2 讲稿



2	<p>C++ 语言程序设计 清华大学出版社</p> <h3>本章主要内容</h3> <ul style="list-style-type: none"> <li>• 计算机程序设计语言的发展</li> <li>• 面向对象的方法</li> <li>• 面向对象的软件开发</li> <li>• 信息的表示与存储</li> <li>• 程序的开发过程</li> </ul>	介绍本章将要讲解的主要内容。
3	<p>C++ 语言程序设计 清华大学出版社</p> <h3>计算机程序</h3> <ul style="list-style-type: none"> <li>• 计算机的工作是用程序来控制的</li> <li>• 程序是指令的集合</li> <li>• 指令是计算机可以识别的命令</li> </ul>	计算机的工作是用程序来控制的。程序是指令的集合。指令就是计算机可以识别的命令。
4	<p>C++ 语言程序设计 清华大学出版社</p> <h3>机器语言与汇编语言</h3> <ul style="list-style-type: none"> <li>• 由计算机硬件系统可以识别的二进制指令组成的语言称为机器语言。 <small>计算机发展的初期,软件工程师们只能用机器语言来编写程序。这一阶段,在人类的自然语言和计算机编程语言之间存在巨大的鸿沟。</small></li> <li>• 汇编语言将机器指令映射为一些可以被人类读懂的助记符,如ADD, SUB等。 <small>此时编程语言与人类自然语言之间的鸿沟略有缩小,但仍与人类的思维相差甚远,因为它的抽象层次太低,程序员需要考虑大量的机器细节。</small></li> </ul>	由计算机硬件系统可以识别的二进制指令组成的语言称为机器语言。汇编语言是将机器指令映射为一些可以被人类读懂的助记符。
5	<p>C++ 语言程序设计 清华大学出版社</p> <h3>高级语言</h3> <p>高级语言屏蔽了机器的细节,提高了语言的抽象层次。程序中可以采用具有一定含义的数据命名和容易理解的执行语句,这使得在书写程序时可以联系到程序所描述的具体事物。</p>	高级语言的出现是计算机编程语言的一大进步。高级语言屏蔽了机器的细节,提高了语言的抽象层次。程序中可以采用具有一定含义的数据命名和容易理解的执行语句。

<p>6</p>	<p>C++ 面向对象设计</p> <h3>面向对象的语言</h3> <p><b>计算机语言的发展</b></p> <ul style="list-style-type: none"> <li>● 出发点:             <ul style="list-style-type: none"> <li>- 更直接地描述客观世界中存在的事物(对象)以及它们之间的关系。</li> </ul> </li> <li>● 特点:             <ul style="list-style-type: none"> <li>- 是高级语言。</li> <li>- 将客观事物看作具有属性和行为的对象。</li> <li>- 通过抽象找出同类对象的共同属性和行为,形成类。</li> <li>- 通过类的继承与多态实现代码重用。</li> </ul> </li> </ul>	<p>面向对象的编程语言与以往各种编程语言的根本不同点在于,它设计的出发点就是为了能更直接地描述客观世界中存在的事物(即对象)以及它们之间的关系。</p> <p>面向对象的编程语言将客观事物看作具有属性和行为(或称服务)的对象,通过抽象找出同类对象的共同属性(静态特征)和行为(动态特征)而形成类。</p>
<p>7</p>	<p>C++ 面向对象设计</p> <h3>面向对象的语言</h3> <p><b>计算机语言的发展</b></p> <ul style="list-style-type: none"> <li>● 优点:             <ul style="list-style-type: none"> <li>使程序能够比较直接地反映问题域的本来面目,软件开发人员能够利用人类认识事物所采用的一般思维方法进行软件开发。</li> </ul> </li> </ul>	<p>面向对象的语言优点:使程序能够比较直接地反映问题域的本来面目,软件开发人员能够利用人类认识事物所采用的一般思维方法进行软件开发。</p>
<p>8</p>	<p>C++ 面向对象设计</p> <h3>程序设计方法的发展历程</h3> <h4>——面向过程的程序设计方法</h4> <p><b>面向对象的方法</b></p> <ul style="list-style-type: none"> <li>● 最早的程序             <ul style="list-style-type: none"> <li>— 目的:用于数学计算。</li> <li>— 主要工作:设计求解问题的过程。</li> <li>— 缺点:对于庞大、复杂的程序难以开发和维护。</li> </ul> </li> </ul>	<p>面向过程的程序设计的目的是用于数学计算。主要工作是设计求解问题的过程。</p> <p>缺点是对于庞大、复杂的程序难以开发和维护。</p>
<p>9</p>	<p>C++ 面向对象设计</p> <h3>程序设计方法的发展历程</h3> <h4>——面向过程的结构化程序设计方法</h4> <p><b>面向对象的方法</b></p> <ul style="list-style-type: none"> <li>● 设计思路:             <ul style="list-style-type: none"> <li>- 自顶向下、逐步求精。采用模块分解与功能抽象,自顶向下、分而治之。</li> </ul> </li> <li>● 程序结构:             <ul style="list-style-type: none"> <li>- 按功能划分为若干个基本模块,形成一个树状结构。</li> <li>- 各模块间的关系尽可能简单,功能上相对独立;每一模块内部都是由顺序、选择和循环3种基本结构组成。</li> <li>- 其模块化实现的具体方法是使用子程序。</li> </ul> </li> </ul>	<p>结构化程序设计的思路:自顶向下、逐步求精;其程序结构是按功能划分为若干个基本模块,这些模块形成一个树状结构;各模块之间的关系尽可能简单,在功能上相对独立;每一模块内部均是由顺序、选择和循环3种基本结构组成;其模块化实现的具体方法是使用子程序。</p>

10	<p>C++ 语言程序设计 教师用书</p> <p><b>程序设计方法的发展历程</b> ——<b>面向过程的结构化程序设计方法</b></p> <p>面向对象的方法</p> <ul style="list-style-type: none"> <li>● 优点: <ul style="list-style-type: none"> <li>- 有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子任务,便于开发和维护。</li> </ul> </li> </ul> <p>10</p>	介绍面向过程的结构化程序设计方法的优点。
11	<p>C++ 语言程序设计 教师用书</p> <p><b>程序设计方法的发展历程</b> ——<b>面向过程的结构化程序设计方法</b></p> <p>面向对象的方法</p> <ul style="list-style-type: none"> <li>● 缺点: 可重用性差、数据安全性差、难以开发大型软件和图形界面的应用软件 <ul style="list-style-type: none"> <li>- 把数据和处理数据的过程分离为相互独立的实体。</li> <li>- 当数据结构改变时,所有相关的处理过程都要进行相应地修改。</li> <li>- 每种相对于老问题的新方法都要带来额外的开销。</li> <li>- 图形用户界面的应用程序,很难用过程来描述和实现,开发和维护也都很困难。</li> </ul> </li> </ul> <p>11</p>	介绍面向过程的结构化程序设计方法的缺点:可重用性差、数据安全性差、难以开发大型软件和图形界面的应用软件。
12	<p>C++ 语言程序设计 教师用书</p> <p><b>程序设计方法的发展历程</b> ——<b>面向对象的方法</b></p> <p>面向对象的方法</p> <ul style="list-style-type: none"> <li>● 将数据及对数据的操作方法封装在一起,作为一个相互依存、不可分离的整体——对象。</li> <li>● 对同类型对象抽象出其共性,形成类。</li> <li>● 类通过一个简单的外部接口与外界发生关系。</li> <li>● 对象与对象之间通过消息进行通信。</li> </ul> <p>12</p>	面向对象的方法是什么呢?首先,它将数据及对数据的操作方法放在一起,作为一个相互依存、不可分离的整体——对象。然后把同类型对象抽象出其共性而形成类。类中的大多数数据,只能用本类的方法进行处理。类通过一个简单的外部接口与外界发生关系,对象与对象之间通过消息进行通信。
13	<p>C++ 语言程序设计 教师用书</p> <p><b>程序设计方法的发展历程</b> ——<b>面向对象的方法</b></p> <p>面向对象的方法</p> <ul style="list-style-type: none"> <li>● 优点: <ul style="list-style-type: none"> <li>- 程序模块间的关系更为简单,程序模块的独立性、数据的安全性有了良好的保障。</li> <li>- 通过继承与多态性,可以大大提高程序的可重用性,使得软件的开发和维护都更为方便。</li> </ul> </li> </ul> <p>13</p>	介绍面向对象的方法的优点。

<p>14.</p>	<p><b>面向对象的基本概念</b> ——对象</p> <p><b>面向对象的方法</b></p> <ul style="list-style-type: none"> <li>● 一般意义上的对象：                     <ul style="list-style-type: none"> <li>- 是现实世界中一个实际存在的事物。</li> <li>- 可以有形的（比如一辆汽车），也可以是无形的（比如一项计划）。</li> <li>- 是构成世界的一个独立单位，具有如下特征：                             <ul style="list-style-type: none"> <li>● 静态特征：可以用某种数据来描述；</li> <li>● 动态特征：对象所表现的行为或具有的功能。</li> </ul> </li> </ul> </li> </ul>	<p>从一般意义上讲，对象是现实世界中一个实际存在的事物，它可以是有形的，也可以是无形的。对象是构成世界的一个独立单位，它具有自己的静态特征和动态特征。</p>
<p>15</p>	<p><b>面向对象的基本概念</b> ——对象</p> <p><b>面向对象的方法</b></p> <ul style="list-style-type: none"> <li>● 面向对象方法中的对象：                     <ul style="list-style-type: none"> <li>- 是系统中用来描述客观事物的一个实体，它是用来构成系统的一个基本单位。对象由一组属性和一组行为构成。</li> <li>- 属性：用来描述对象静态特征的数据项。</li> <li>- 行为：用来描述对象动态特征的操作序列。</li> </ul> </li> </ul>	<p>面向对象方法中的对象，是系统中用来描述客观事物的一个实体，它是用来构成系统的一个基本单位。对象由一组属性和一组行为构成。属性是用来描述对象静态特征的数据项，行为是用来描述对象动态特征的操作序列。</p>
<p>16</p>	<p><b>面向对象的基本概念</b> ——类</p> <p><b>面向对象的方法</b></p> <ul style="list-style-type: none"> <li>● 分类——人类通常的思维方法</li> <li>● 分类所依据的原则——抽象                     <ul style="list-style-type: none"> <li>- 忽略事物的非本质特征，只注意那些与当前目标有关的本质特征，从而找出事物的共性，把具有共同性质的事物划分为一类，得出一个抽象的概念。</li> <li>- 例如，石头、树木、汽车、房屋等都是人们在长期的生产和生活实践中抽象出的概念。</li> </ul> </li> </ul>	<p>分类所依据的原则是抽象，即忽略事物的非本质特征，只注意那些与当前目标有关的本质特征，从而找出事物的共性，把具有共同性质的事物划分为一类，得出一个抽象的概念。</p>
<p>17</p>	<p><b>面向对象的基本概念</b> ——类</p> <p><b>面向对象的方法</b></p> <ul style="list-style-type: none"> <li>● 面向对象方法中的“类”                     <ul style="list-style-type: none"> <li>- 具有相同属性和服务的一组对象的集合。</li> <li>- 为属于该类的全部对象提供了抽象的描述，包括属性和行为两个主要部分。</li> <li>- 类与对象的关系：犹如模具与铸件之间的关系，一个属于某类的对象称为该类的一个实例。</li> </ul> </li> </ul>	<p>面向对象方法中的“类”是具有相同属性和服务的一组对象的集合。它为属于该类的全部对象提供了抽象的描述，其内部包括属性和行为两个主要部分。</p> <p>类与对象的关系犹如模具与铸件之间的关系，一个属于某类的对象称为该类的一个实例。</p>

<p>18</p>	<p>C++ 语言程序设计</p> <h3>面向对象的基本概念</h3> <h4>封装</h4> <p>面向对象的方法</p> <ul style="list-style-type: none"> <li>• 把对象的属性和服务结合成一个独立的系统单元。</li> <li>• 尽可能隐蔽对象的内部细节。对外形成一个边界(或者说一道屏障),只保留有限的对外接口使之与外部发生联系。</li> </ul> <p>18</p>	<p>封装是面向对象方法的一个重要原则,就是把对象的属性和服务结合成一个独立的系统单位,并尽可能隐蔽对象的内部细节。</p> <p>这包括两个含义:第1个含义是,把对象的全部属性和全部服务结合在一起,形成一个不可分割的独立单位;第2个含义也称作“信息隐蔽”,即尽可能隐蔽对象的内部细节,对外形成一个边界(或者说一道屏障),只保留有限的对外接口用来与外部发生联系。</p>
<p>19</p>	<p>C++ 语言程序设计</p> <h3>面向对象的基本概念</h3> <h4>继承</h4> <p>面向对象的方法</p> <ul style="list-style-type: none"> <li>• 继承对于代码复用有着重要意义,是面向对象技术能够提高软件开发效率的重要原因之一。</li> <li>• 定义:特殊类的对象拥有其一般类的全部属性与服务,称作特殊类对一般类的继承。</li> <li>• 例如:将轮船作为一个一般类,客轮便是一个特殊类。</li> </ul> <p>19</p>	<p>继承是面向对象技术能够提高软件开发效率的重要原因之一。继承的定义:特殊类的对象拥有其一般类的全部属性与服务,称作特殊类对一般类的继承。</p>
<p>20</p>	<p>C++ 语言程序设计</p> <h3>面向对象的基本概念</h3> <h4>多态性</h4> <p>面向对象的方法</p> <ul style="list-style-type: none"> <li>• 多态是指在一般类中定义的属性或行为,被特殊类继承之后,可以具有不同的数据类型或表现出不同的行为。这使得同一个属性或行为在一般类及其各个特殊类中具有不同的语义。</li> <li>• 例如:       <ul style="list-style-type: none"> <li>- 数的加法-&gt;实数的加法</li> <li>- 复数的加法</li> </ul> </li> </ul> <p>20</p>	<p>多态性是指在一般类中定义的属性或行为,被特殊类继承之后,可以具有不同的数据类型或表现出不同的行为。这使得同一个属性或行为在一般类及其各个特殊类中具有不同的语义。</p>
<p>21</p>	<p>C++ 语言程序设计</p> <h3>面向对象的软件工程</h3> <p>面向对象的软件开发</p> <ul style="list-style-type: none"> <li>• 面向对象的软件工程是面向对象方法在软件工程领域的全面应用。其中包括:       <ul style="list-style-type: none"> <li>- 面向对象的分析(OOA)</li> <li>- 面向对象的设计(OOD)</li> <li>- 面向对象的编程(OOP)</li> <li>- 面向对象的测试(OOT)</li> <li>- 面向对象的软件维护(OOSM)</li> </ul> </li> </ul> <p>21</p>	<p>学习面向对象的程序设计,首先应该对软件开发和维护的全过程有一个初步了解。什么是面向对象的软件工程?面向对象的软件工程是面向对象方法在软件工程领域的全面应用。它包括面向对象的分析(OOA)、面向对象的设计(OOD)、面向对象的编程(OOP)、面向对象的测试(OOT)和面向对象的软件维护(OOSM)等主要内容。</p>



22	<p>C++ 面向对象编程</p> <h3>系统分析</h3> <p>面向对象软件开发</p> <ul style="list-style-type: none"> <li>● 系统分析阶段应该扼要精确地抽象出系统必须做什么, 但是不关心如何去实现。</li> <li>● 面向对象的系统分析, 直接用问题域中客观存在的事物建立模型中的对象, 对单个事物及事物之间的关系, 都保留它们的原貌, 不做转换, 也不打破原有界限而重新组合, 因此能够很好地映射客观事物。</li> </ul> <p>22</p>	<p>面向对象的系统分析, 直接用问题域中客观存在的事物建立模型中的对象, 无论是对单个事物还是对事物之间的关系, 都保留它们的原貌, 不做转换, 也不打破原有界限而重新组合, 因此能够很好地映射客观事物。</p>
23	<p>C++ 面向对象编程</p> <h3>设计</h3> <p>面向对象软件开发</p> <ul style="list-style-type: none"> <li>● 针对系统的一个具体实现运用面向对象的方法。其中包括两方面的工作:             <ul style="list-style-type: none"> <li>- 把OOA模型直接搬到OOD, 作为OOD的一部分。</li> <li>- 针对具体实现中的人机界面、数据存储、任务管理等因素补充一些与实现有关的部分。</li> </ul> </li> </ul> <p>23</p>	<p>设计阶段是针对系统的一个具体实现运用面向对象的方法。其中包括两方面的工作, 一是把 OOA 模型直接搬到 OOD, 作为 OOD 的一部分; 二是针对具体实现中的人机界面、数据存储、任务管理等因素补充一些与实现有关的部分。</p>
24	<p>C++ 面向对象编程</p> <h3>编程</h3> <p>面向对象软件开发</p> <p>OOP工作就是用一种面向对象的编程语言把OOD模型中的每个成分书写出来, 是面向对象的软件开发最终落实的重要阶段。</p> <p>24</p>	<p>编程是面向对象的软件开发最终落实的重要阶段。在 OOA 和 OOD 理论出现之前, 程序员要写一个好的面向对象的程序, 首先要学会运用面向对象的方法来认识问题域, 所以 OOP 被看作一门比较高深的技术。现在, OOP 的工作比较简单, 认识问题域与设计系统成分的工作也将在 OOA 和 OOD 阶段完成, OOP 工作就是用一种面向对象的编程语言把 OOD 模型中的每个成分书写出来。</p>
25	<p>C++ 面向对象编程</p> <h3>测试</h3> <p>面向对象软件开发</p> <ul style="list-style-type: none"> <li>● 测试的任务是发现软件中的错误。</li> <li>● 在面向对象的软件测试中继续运用面向对象的概念与原则来组织测试, 以对象的类作为基本测试单位, 可以更准确地发现程序错误并提高测试效率。</li> </ul> <p>25</p>	<p>测试的任务是发现软件中的错误, 任何一个软件产品在交付使用之前都要经过严格的测试。在面向对象的软件测试中继续运用面向对象的概念与原则来组织测试, 以对象的类作为基本测试单位, 可以更准确地发现程序错误并提高测试效率。</p>