<<计算机专业英语>>

图书基本信息

书名：<<计算机专业英语>>

13位ISBN编号：9787302223597

10位ISBN编号：7302223599

出版时间：2010-8

出版时间：清华大学出版社

作者：卜艳萍，周伟 编著

页数：391

<<计算机专业英语>>

前言

“计算机专业英语”是一门内容丰富、集计算机专业知识和英语运用能力为一体的应用型课程，其综合性强、内容广泛。
通过对本教材的学习，学生能掌握大量的科技和专业词汇，基本能看懂计算机专业硬件、软件、应用等方面的专业资料，能比较顺利地阅读计算机及其零部件的说明书，能基本理解各种软件的操作提示和帮助信息。
其主要任务是为计算机科学与技术专业学生和工程技术人员奠定必要的计算机专业英语知识基础，培养阅读、理解和翻译计算机专业文献的能力。
本课程为毕业设计时阅读英文专业资料打下良好的基础，并为将来工作中顺利解决涉及专业英语知识的问题提供必要的知识保证。

本书共分28课，讲述了计算机硬件基础、计算机系统结构、计算机软件思想、编程语言、计算机网络结构、电子商务应用、计算机图像与动画以及计算机领域的新技术等内容。
每课内容相对独立，如数据结构、操作系统、无线网络、远程教育、人工智能、电子支付等。
每课后均附有重点词汇、课文难点注释、练习、两篇与课文内容相关的阅读材料以及计算机专业英语中相关的语法知识讲述或专业术语介绍。
书末附有课文及第一篇阅读材料的参考译文。
第二篇阅读材料不提供参考译文，以便于在课堂进行循序渐进的阅读训练，考查学生阅读能力的提高情况。

本教材有以下几个方面的特点：　①计算机专业知识丰富，涵盖了大部分计算机专业基础课和专业课的内容；　②介绍了必要的语法知识及专业文章的翻译方法及技巧；　③介绍了大量计算机专业常用术语；　④注意与计算机技术专业课的协调性；　⑤注重实践性和实用性；　⑥课文长度及选材难度适当；　⑦每课配有关键词、注释及大量习题；　⑧每课配有两篇阅读材料供学生阅读，以提高学生阅读相关知识及掌握词汇的能力。

<<计算机专业英语>>

内容概要

本书包含了大量与计算机专业基础、专业技术及专业前沿知识相关的英文资料，并根据知识点分成28课，分别讲述了计算机硬件基础、计算机系统结构、计算机软件思想、编程语言、计算机网络结构、电子商务应用、计算机图像与动画以及计算机领域的新技术等内容。
每课内容相对独立，如数据结构、操作系统、无线网络、远程教育、人工智能、电子支付等。
每课后均附有重点词汇、课文难点注释、练习、两篇与课文内容相关的阅读材料以及计算机专业英语中的语法知识讲述或专业术语介绍。
书末附有课文及第一篇阅读材料的参考译文。
　　本教材可作为计算机科学与技术专业及相关电类、信息类专业本科学生的教学用书，也可作为相关领域专业技术人员的参考书。

<<计算机专业英语>>

书籍目录

章节摘录

The second phase is design. Design concentrates on how the system will accomplish the goals. It is here that the structure of the software system is established. The input to this phase is a（debugged and validated ）requirements document：the output is a design expressed in some appropriate form（for example ，pseudo-code）.Validation of a design is important. Each requirement in the requirements document must have a corresponding design fragment to meet it. Formal verification，while possible to a limited extent，can be exceedingly difficult. More informal rev lve the entire design team，management，and even the client. It is a well-established principle that the best structure for a large software system is a modular one. Indeed，it is by means of this modular decomposition that the implementation of large systems becomes a possibility. Without such a breakdown，the technical details required in the implementation of a large system would exceed a human ’s comprehensive powers. With a modular design，how eve Lonly the details pertaining to the module under consideration need be mastered. This same modular design is also conductive to furore maintenance because it allows changes to be made on a modular basis. The third phase，implementation，is the actual coding of the design developed in the second phase. The design must be translated into a machine-readable form. The coding step performs this task. If design is performed in a detailed manner, coding can be accomplished mechanistically. The lure of this phase is strong，and many a fool hardy programmer has been drawn to it before adequately laying the groundwork in the first two phases. As a result，requirements are incompletely understood and the design is flawed. The implementation proceeds blindly, and many problems arise as a result. The forth phase is software testing. Testing is closely associated with implementation，because each module of the system is normally tested as it is implemented. In the development of a large system, testing involves several stages. First，each program module is tested as a single program，usua ly isolated from the other programs in the system. Such testing is defined as module testing or unit testing. Unit testing is done in a controlled environment whenever possible so that the test team Call feed a predetermined set of data to the module being tested and observe what output data are produced. H In addition，the test team checks the internal data structures，the logic，and the boundary conditions for the in put and out put data. When collections of modules have been unit-tested. the next step is to insure that the interfaces among the modules are defined and handled properly. Integration testing is the process of verifying that the components of a system work together as described in the program design and system design specifications. Once we are sure that information is passed among modules according to the design prescriptions ，we test the system to assure that it has the desired functionality. A function test evaluates the system to determine if the functions described by the requirements specification are actually performed by the integrated system.

Unfortunately, the testing and debugging of a system is extremely difficult to perform successfully. Experience has shown that large software systems can contain numerous errors. Even after significant testing. Many of these errors may go undetected for the life of the system，but others may cause major malfunctions. The elimination of such errors is one of the goals of software engineering.

<<计算机专业英语>>

编辑推荐

《计算机专业英语》的教学目标明确，注重理论与实践的结合。
教学方法灵活，培养学生自主学习的能力。
教学内容先进，强调计算机在各专业中的应用。
教学模式完善，提供配套的教学资源解决方案。

<<计算机专业英语>>

版权说明