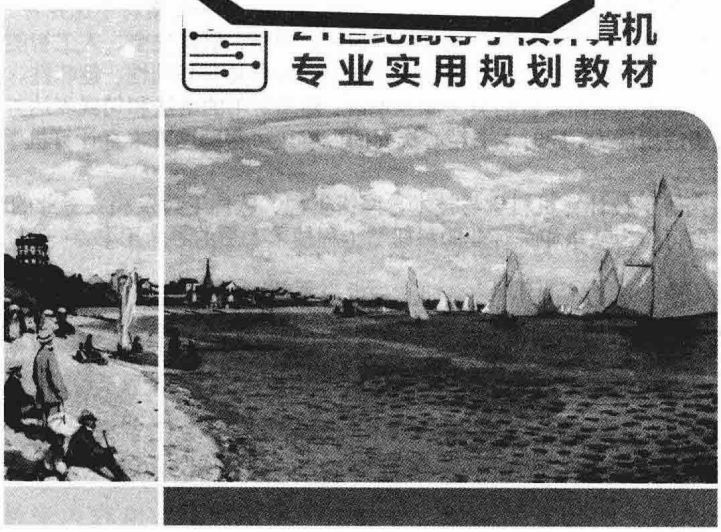
21世纪高等学校计算机
专业实用规划教材

计算机专业英语

◎ 戚文静 李晓峰 刘学 李国文 编著

清華大學出版社



专业实用规划教材

计算机专业英语

◎ 戚文静 李晓峰 刘学 李国文 编著

清华大学出版社
北京

内 容 简 介

本书从计算机科学各个领域的最新教材、专著、论文、百科等英文素材中选择各分支学科的基本概念、方法及最新发展等内容，涵盖计算机硬件、软件、数据库、网络、信息安全、人工智能、机器学习、人机交互、量子计算、大数据、云计算等领域的基础理论和应用，内容具有基础性、趣味性、广泛性和知识性。读者通过阅读和学习本书，能够掌握大量的英文专业术语、熟悉常用的语法和句式表达方法，提高读者在计算机科学相关领域的英语应用能力。同时本书精心选择的内容还可帮助读者丰富计算机学科知识、拓展科学研究视野。

本书可作为普通高等学校计算机科学与技术、网络工程、软件工程及相关专业的英语教材或计算机导论课程的双语教材，也可作为从事相关行业的科研与工程技术人员的参考用书。

图书在版编目（CIP）数据

计算机专业英语/戚文静等编著. —北京：清华大学出版社，2019
（21 世纪高等学校计算机专业实用规划教材）
ISBN 978-7-302-51129-8

Ⅰ. ①计…　Ⅱ. ①戚…　Ⅲ. ①电子计算机 – 英语 – 高等学校 – 教材　Ⅳ. ①TP3

中国版本图书馆 CIP 数据核字（2018）第 202518 号

责任编辑：黄　芝　薛　阳
封面设计：刘　键
责任校对：梁　毅
责任印制：宋　林

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市君旺印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张：**22　　**字　　数：**534 千字
版　　次：2019 年 6 月第 1 版　　**印　　次：**2019 年 6 月第 1 次印刷
印　　数：1～1500
定　　价：59.00 元

产品编号：054041-01

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程（简称‘质量工程’）”，通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程（群）加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机专业课程领域，以专业基础课为主、专业课为辅，横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

（1）反映计算机学科的最新发展，总结近年来计算机专业教学的最新成果。内容先进，充分吸收国外先进成果和理念。

（2）反映教学需要，促进教学发展。教材要适应多样化的教学需要，正确把握教学内容和课程体系的改革方向，融合先进的教学思想、方法和手段，体现科学性、先进性和系统性，强调对学生实践能力的培养，为学生知识、能力、素质协调发展创造条件。

（3）实施精品战略，突出重点，保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上；特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现教学质量和教学改革成果的教材。

（4）主张一纲多本，合理配套。专业基础课和专业课教材配套，同一门课程有针对不同层次、面向不同应用的多本具有各自内容特点的教材。处理好教材统一性与多样化，基本教材与辅助教材、教学参考书，文字教材与软件教材的关系，实现教材系列资源配套。

（5）依靠专家，择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主题。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机专业实用规划教材
联系人：黄芝 huangzh@tup.tsinghua.edu.cn

前言

“计算机英语”是计算机及相关专业的一门专业基础课程。由于计算机相关核心技术大部分源于英语国家，很多的技术文档资料、编程工具原版都以英语为主，因此较好的英文水平有利于专业和职业的长远发展；另外，计算机技术更新速度极快，如果不掌握一定的专业词汇，必然会影响对新技术的理解和消化。为了提高学生在计算机相关领域的英语应用能力，使学生在有限的学时中比较全面地掌握专业词汇、能顺利地阅读和理解专业文献，同时保证学生的学习兴趣、拓展专业知识面、避免陷于深奥的专业知识，本教材在内容的编排和选择上坚持以下三个原则。

基础性和趣味性原则：每一部分内容介绍计算机科学一个学科领域的科普性内容，避免出现较深奥和晦涩的概念及理论，易于读者阅读和理解。

广泛性和知识性原则：力求覆盖计算机科学相关领域内容，高度概括每个领域的发展历史、研究内容和发展情况，使不同层次的读者通过阅读和学习可以对该领域有全面、客观的了解。

可靠性和实效性原则：素材取自国内外最近几年计算机科学各个领域的最新教材、专著、论文、学者访谈、百科等可靠来源，使读者可以了解计算机科学与技术的最新进展。

本教材每章内容相对独立，教师可根据实际教学需要确定重点授课内容。本书第 1 章概括计算机科学的发展历史、学科分支及未来发展方向；第 2 章集合了计算机体系结构、硬件、互联网、无线网方面的基础知识；第 3 章介绍操作系统的常识，包括操作系统的定义、任务及常用操作系统的案例及对比；第 4 章介绍算法、数据结构和软件工程方面的知识；第 5 章集合了关系数据库、查询语言、信息检索和网页检索的基础概念和常识；第 6 章为人工智能相关领域的知识，包括图灵测试、知识表达和推理、机器人、计算机视觉、人工智能可能存在的风险等；第 7 章为计算图形学领域的分支和应用，包括计算机辅助设计、3D 建模、虚拟现实、数据可视化等；第 8 章为人机交互技术方面的相关知识，包含人机交互的发展、人机界面的设计原理和规则等；第 9 章为计算机安全相关知识，包括基本概念、安全措施、密码学及网络战争等内容；第 10 章介绍计算机科学与技术领域的一些最新进展，包括量子计算、深度学习、云计算和大数据方面的内容。本书不仅是一本计算机专业英语的教材，也是对计算机专业知识的一个概览，具有很强的科学性、知识性、趣味性。

本书的第 1～4 章主要由刘学副教授编写；第 5 章和第 6 章主要由李晓峰教授编写；第 7、8、10 章及附录词汇部分主要由戚文静教授编写和整理；第 9 章和第 10 章由李国文博士编写；最后由戚文静完成对本书的统稿；另外，刘宇祺、王亦佳等同学参与了本书稿的录入和校对工作，赵敬教授对本书的编写提出了很多宝贵意见。

另外，在清华大学出版社编辑的大力支持和协助下，本书得以在较短时间内出版和发行，在此对他们的工作表示诚挚的感谢。同时，感谢戚文静家人对我的理解、支持和关爱，使得本书稿能够顺利完成。

教材中难免有不足和疏漏之处，恳请广大读者提出宝贵意见和建议，以便进一步完善本教材。

戚文静

2018 年夏于映雪湖

Contents

Chapter 1 Introduction of Computer Science

1.1 History of Computer Science

The start of the modern science that we call "Computer Science" can be traced back to a long ago age. In Asia, the Chinese were becoming very involved in commerce with the Japanese, Indians, and Koreans. Businessmen needed a way to tally accounts and bills. Somehow, out of this need, the abacus was born. The abacus is the first true precursor to the adding machines and computers which would follow. For over a thousand years after the Chinese invented the abacus, not much progress was made to automate counting and mathematics. The Greeks came up with numerous mathematical formulae and theorems, but all of the newly discovered math had to be worked out by hand. Most of the tables of **integrals**, **logarithms**, and **trigonometric** values were worked out this way, their accuracy unchecked until machines could generate the tables in far less time and with more accuracy than a team of humans could ever hope to achieve.

Blaise Pascal, noted mathematician, thinker, and scientist, built the first mechanical adding machine in 1642 based on a design described by Hero of Alexandria to add up the distance a carriage travelled. The basic principle of his calculator is still used today in water meters and modern-day **odometers**. This first mechanical calculator, called the Pascaline, as shown in Fig.1-1, had several disadvantages. Although it did offer a substantial improvement over manual calculations, only Pascal himself could repair the device and it cost more than the people it replaced! In addition, the first signs of **technophobia** emerged with mathematicians fearing the loss of their jobs due to the progress.

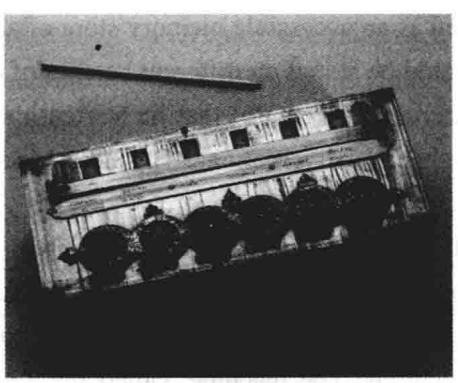


Fig.1-1 Pascal calculating machine

Fig.1-2 Arithmometer

The **Arithmometer**, as shown in Fig.1-2, was the first mechanical calculator strong enough and reliable enough to be used daily in an office environment. This calculator could add and subtract two numbers directly and could perform long multiplications and divisions effectively by using a movable **accumulator** for the result. Patented in France by Thomas de Colmar in 1820 and manufactured from 1851 to 1915, it became the first commercially successful mechanical calculator.

While Thomas de Colmar was developing the first successful commercial calculator, Charles Babbage realized as early as 1812 that many long computations consisted of operations that were regularly repeated. He theorized that it must be possible to design a calculating machine which could do these operations automatically. He produced a **prototype** of this "**difference engine**" by 1822 and with the help of the British government and started working on the full machine in 1823. It was intended to be **steam-powered**; fully automatic, even to the printing of the resulting tables; and commanded by a fixed **instruction program**. This machine used the decimal number system and was powered by cranking a handle. The British government was interested, since producing tables was time consuming and expensive and they hoped the difference engine would make the task more economical[3].

In 1833, Babbage ceased working on the difference engine because he had a better idea. His new idea was to build an "**analytical engine**." The analytical engine was a real parallel decimal computer which would operate on words of 50 decimals and was able to store 1000 such numbers. The machine would include a number of built-in operations such as **conditional control**, which allowed the instructions for the machine to be executed in a specific order rather than in numerical order. The instructions for the machine were to be stored on **punched cards**, similar to those used on a Jacquard loom.

A step toward automated computation was the introduction of punched cards, which were first successfully used in connection with computing in 1890 by Herman Hollerith working for the US Census Bureau. He developed a device which could automatically read census information which had been punched onto card. Surprisingly, he did not get the idea from the work of Babbage, but rather from watching a train conductor punch tickets. As a result of his invention, reading errors were consequently greatly reduced, work flow was increased, and, more important, stacks of punched cards could be used as an accessible memory store of almost unlimited capacity; furthermore, different problems could be stored on different batches of cards and worked on as needed. Hollerith's **tabulator** became so successful that he started his own firm to market the device; this company eventually became International Business Machines (IBM).

Hollerith's machine though had limitations. It was strictly limited to tabulation. The punched cards could not be used to direct more complex computations. In 1941, Konrad Zuse, a German who had developed a number of calculating machines, released the first programmable computer designed to solve complex engineering equations. The machine, called the Z3, was controlled by **perforated strips** of discarded movie film. As well as being controllable by these

celluloid strips, it was also the first machine to work on the **binary system**, as opposed to the more familiar **decimal system**. Binary representation was proven to be important in the future design of computers which took advantage of a multitude of two-state devices such card readers, electric circuits which could be on or off, and vacuum tubes.

By the late 1930s, punched-card machine techniques had become so well established and reliable that a large automatic digital computer, called the Harvard Mark I, was constructed, which could handle 23-decimal-place numbers and perform all four **arithmetic operations**; moreover, it had special built-in programs, or **subroutines**, to handle logarithms and trigonometric functions. Meanwhile, the British mathematician Alan Turing wrote a paper in 1936 entitled On Computable Numbers in which he described a hypothetical device, a **Turing machine**, which presaged programmable computers. The Turing machine was designed to perform logical operations and could read, write, or erase symbols written on squares of an infinite paper tape. This kind of machine came to be known as a **finite state machine** because at each step in a computation, the machine's next action was matched against a finite instruction list of possible states.

Back in America, John W. Mauchly and J. Presper Eckert at the University of Pennsylvania built giant ENIAC machine. ENIAC contained 17,468 vacuum tubes, 7,200 crystal diodes, 1,500 relays, 70,000 resistors, 10,000 capacitors and around 5 million hand-soldered joints. It weighed more than 30 short tons (27 t), was roughly 8 by 3 by 100 feet (2.4m × 0.9m × 30m), took up 1800 square feet ($167m^2$), and consumed 150kW of power[4]. This led to the rumor that whenever the computer was switched on, lights in Philadelphia dimmed. ENIAC is generally acknowledged to be the first successful high-speed electronic digital computer, it was efficient in handling the particular programs for which it had been designed and was productively used from 1946 to 1955.

In 1945, mathematician John von Neumann contributed a new understanding of how practical fast computers should be organized and built; these ideas, often referred to as the **stored-program technique**, became fundamental for future generations of high-speed digital computers and were universally adopted. The primary advance was the provision of a special type of machine instruction called conditional control transfer, which permitted the program sequence to be interrupted and reinitiated at any point, and by storing all instruction programs, instructions could be arithmetically modified in the same way as data. As a result, frequently used subroutines did not have to be reprogrammed for each new problem but could be kept intact in "libraries" and read into memory when needed. The computer control served as an errand runner for the overall process. The first-generation stored-program computers required considerable maintenance, attained perhaps 70% to 80% reliable operation, and were used for 8 to 12 years. Typically, they were programmed directly in **machine language**, although by the mid-1950s progress had been made in several aspects of advanced programming. This group of machines included EDVAC and UNIVAC, the first commercially available computers.

BASIC (Beginners All-purpose Symbolic Instruction Code) had originally been developed in 1963 by Thomas Kurtz and John Kemeny, it was designed to provide an interactive, easy method for upcoming computer scientists to program computers. By this time, a number of other specialized and general-purpose languages had been developed. A surprising number of today's popular languages have actually been around since the 1950s. FORTRAN, developed by a team of IBM programmers, was one of the first high level languages, in which the programmer does not have to deal with the machine code of 0s and 1s. It was designed to express scientific and mathematical formulas. COBOL was developed in 1960 by a joint committee. It was designed to produce applications for the business world and had the novice approach of separating the data descriptions from the actual program. In the late 1960s, a Swiss computer scientist, Niklaus Wirth, released Pascal, which forced programmers to program in a structured, logical fashion and pay close attention to the different types of data in use.

Operating systems are the **interface** between the user and the computer. Windows is one of the numerous **graphical user interfaces** around that allows the user to manipulate their environment using a mouse and icons. Other examples of Graphical User Interfaces (GUIs) include X-Windows, which runs on UNIX® machines, or Mac OS X, which is the operating system of the Macintosh. (1-1) An **application** is any program that a computer runs that enables you to get things done. This includes things like word processors for creating text, graphics packages for drawing pictures, and communication packages for moving data around the globe.

The Web was developed at CERN (European Organization for Nuclear Research) in Switzerland during 1980s. As a new form of communicating text and graphics across the Internet, it makes use of the **hypertext markup language** (HTML) as a way to describe the attributes of the text and the placement of graphics, sounds, or even movie clips. Since it was first introduced, the number of users has blossomed and the number of sites containing information and searchable archives has been growing at an unprecedented rate.

1.2 Areas of Computer Science

Computer science can be divided into a variety of theoretical and practical disciplines. Some fields, such as **computational complexity** theory, are highly abstract, whilst fields such as computer graphics emphasize real-world applications.

1.2.1 Theoretical Computer Science

1. Theory of Computation

According to Peter J. Denning, the fundamental question underlying computer science is, "What can be efficiently automated?" The study of the theory of computation is focused on answering fundamental questions about what can be computed and what amount of resources are