◎ 李长庚　主编

大学英语准专业英语阅读

电脑·电信类

College English Sub-Specialized English Reading

the Computer & Telecommunication

安徽科学技术出版社

大学英语准专业英语阅读

电脑·电信类

主　编　李长庚

编　者（按姓氏笔画为序）

邓　英　李长庚　沈逢桂

黄　川　黄　贵

安徽科学技术出版社

图书在版编目(CIP)数据

大学英语准专业英语阅读.电脑、电信类/李长庚主编.
—合肥:安徽科学技术出版社,2003.11
ISBN 7-5337-2806-8

Ⅰ.大… Ⅱ.李… Ⅲ.①英语-阅读教学-高等学校-自学参考资料②电子计算机-英语-阅读教学-高等学校-自学参考资料③电信-英语-阅读教学-高等学校-自学参考资料 Ⅳ.H319.4

中国版本图书馆CIP数据核字(2002)第000088号

*

安徽科学技术出版社出版
(合肥市跃进路1号新闻出版大厦)
邮政编码:230063
电话号码:(0551)2825419
新华书店经销 合肥义兴印务有限责任公司印刷

*

开本:787×1092 1/16 印张:12 字数:307千
2003年11月第1版 2003年11月第1次印刷
定价:14.50元

前　言

本书为电脑、电信类准专业英语阅读教材。

《出版说明》中明确指出“准专业英语教材”的特点是：语言难度和课文长度略高于四级要求，专业内容力求做到面广易懂，练习设计编排实现由四级英语式向专业英语式转移。我们在编写这本教材时，注意遵循了上述原则。文章选材尽量做到内容新，涉及面广，既涵盖有关专业内容，又避免专业性太强，力求熔专业性、知识性、时效性、可读性为一体。每课的 Part A、Part B 和 Part C 为一相同或相近话题，互为补充，互相映衬。内容主要涉及计算机、电信、电视、网络、办公自动化、计算机辅助教学、电子游戏、电器等方面。练习设计编排既考虑到方便教学，又真正体现由四级英语式向专业英语式转移的自然过渡和衔接。

本书由李长庚主编。参与编写的人员有：李长庚（Unit 1，Unit 8 和 Glossary）、黄贵(Unit 2，Unit 4 的 Part B 和 Part C，Unit 7 的 Part B)、沈逢桂（Unit 3，Unit 4 的 Part A，Unit 6 的 Part B 和 Part C)、黄川（Unit 5 的 Part A 和 Part C，Unit 9 的 Part B 和 Part C，Unit 7 的 Part A 和 Part C)、邓英（Unit 5 的 Part B，Unit 6 的 Part A，Unit 9 的 Part A，Unit 10)。

本书内容详实，可读性强，主要供处于四级英语式向专业英语式过渡阶段的在校大学生使用，也可供具有中级英语水平并对电脑、电信感兴趣的读者使用。

编　者

出 版 说 明

世纪之交，国家教育部正式颁布实施了新修订的《大学英语教学大纲》。这是一部既总结过去又瞻望未来的新大纲。它凝结着教育部有关领导和专家的心血与汗水，汇集着大学英语教师十几年教书育人的经验和智慧，是指导全国大学英语界同仁迎接21世纪挑战、促进大学英语教学上新台阶的行动纲领。

具有鲜明时代特征的新大纲，以实施素质教育的思想为指导，对大学英语教学的各方面提出了定性、定量的新标准及具体的安排。它将大学英语教学分为两个阶段，即基础阶段（1～6级，第1～4学期）和应用提高阶段（第5～8学期，该阶段主要学习专业英语和高级英语），明确提出大学英语教学要四年不断线。根据新大纲这一要求，并结合编者多年开展的四级后教学试验，我们拟将大学英语的全程教育分为基础阶段（第1～4学期）、准专业英语阶段（第5学期）、专业英语阶段（第6～7学期），即“二大一小”3个阶段。这种“4（学期）+1（学期）+2（学期）”教学安排模式一是基于多年的教学实践，二是符合“循序渐进”、“巩固提高”的认知规律和英语语言学习规律。为配合“412”模式教学，尤其为了使准专业英语阶段的教学起到过渡性和自然衔接的作用，我们特别成立了由资深副教授以上职称的大学英语教师组成的准专业英语阅读编写委员会。具体分工为：李长庚：电脑、电信类；闫学迅：金融、财经类；沈传海：人文、管理类；何建梅：生化、材料、环境类；俞进：土木、建筑类；童传富：机械、仪器、仪表类。这6类准专业英语阅读材料的特点在于：语言难度和课文长度略高于四级要求，专业内容力求做到面广易懂，练习设计编排实现由四级英语式向专业英语式转移。根本目的是有助于这一阶段大学英语教师开展教学，真正使学生获益，达到预期的效果。

大学英语准专业英语阅读编写委员会

大学英语准专业英语阅读

编写委员会

主　任：顾成华

副主任：沈逢桂

委　员：（按姓氏笔画为序）

闫学迅　李长庚　沈传海

何建梅　俞　进　童传富

Contents

Unit 1

Part A

Computer Science

Introduction

Computer Science is the study of the theory, experimentation, and engineering that form the basis for the design and use of computers—devices that automatically process information. Computer science traces its roots to work done by English mathematician Charles Babbage[1], who first proposed a programmable mechanical calculator in 1837. Until the advent of electronic digital computers in the 1940s, computer science was not generally distinguished as being separate from mathematics and engineering.[2] Since then it has sprouted numerous branches of research that are unique to the discipline.

The Development of Computer Science

Early work in the field of computer science during the late 1940s and early 1950s focused on automating the process of making calculations for use in science and engineering. Scientists and engineers developed theoretical models of computation that enabled them to analyze how efficient different approaches were in performing various calculations. Computer science overlapped considerably during this time with the branch of mathematics known as numerical analysis, which examines the accuracy and precision of calculations.

As the use of computers expanded between the 1950s and the 1970s, the focus of computer science broadened to include simplifying the use of computers through programming languages[3]—artificial languages used to program computers, and operating systems—computer programs that provide a useful interface between a computer and a user. During this time, computer scientists were also experimenting with new applications and computer designs, creating the first computer networks, and exploring relationships between computation and thought.

In the 1970s, computer chip manufacturers began to mass produce microprocessors—the electronic circuitry that serves as the main information processing center in a computer. This new technology revolutionized the computer industry by dramatically reducing the cost of building computers and greatly increasing their processing speed. The microprocessor made possible the advent of the personal computer, which resulted in an explosion in the use of computer applications.[4] Between the early 1970s and 1980s, computer science rapidly expanded in an effort to develop new applications for personal computers and to drive the technological advances in the computing industry. Much of the earlier research that had been done began to reach the public through personal computers, which derived most of their early software from existing concepts and systems.

Computer scientists continue to expand the frontiers of computer and information systems by pioneering the designs of more complex, reliable, and powerful computers; enabling networks of computers to efficiently exchange vast amounts of information; and seeking ways to make computers behave intelligently. As computers become an increasingly integral part of modern society, computer scientists strive to solve new problems and invent better methods of solving current problems.

The goals of computer science range from finding ways to better educate people in the use of existing computers to highly speculative research into technologies and approaches that may not be viable for decades. Underlying all of these specific goals is the desire to better the human condition today and in the future through the improved use of information.[5]

Theory and Experiment

Computer science is a combination of theory, engineering, and experimentation. In some cases, a computer scientist develops a theory, then engineers a combination of computer hardware and software based on that theory, and experimentally tests it. An example of such a theory-driven approach is the development of new software engineering tools that are then evaluated in actual use. In other cases, experimentation may result in new theory, such as the discovery that an artificial neural network exhibits behavior similar to neurons in the brain, leading to a new theory in neurophysiology.

It might seem that the predictable nature of computers makes experimentation unnecessary because the outcome of experiments should be known in advance. But when computer systems and their interactions with the natural world become sufficiently complex, unforeseen behaviors can result.[6] Experimentation and the traditional scientific method are thus key parts of computer science.

Major Branches of Computer Science

Computer science can be divided into four main fields: software development, computer architecture (hardware), human-computer interfacing (the design of the most efficient ways for humans to use computers), and artificial intelligence (the attempt to make computers behave intelligently). Software development is concerned with creating computer programs that perform efficiently. Computer architecture is concerned with developing optimal hardware for specific computational needs. The areas of artificial intelligence (AI) and human-computer interfacing often involve the development of both software and hardware to solve specific problems.

Software Development

In developing computer software, computer scientists and engineers study various areas and techniques of software design, such as the best types of programming languages and algorithms (see below) to use in specific programs, how to efficiently store and retrieve information, and the computational limits of certain software-computer combinations. Software designers must consider many factors when developing a program. Often, program performance in one area must be sacrificed for the sake of the general performance of the software. For instance, since computers have only a limited amount of memory, software designers must limit the number of features they include in a program so that it will not require more memory than the system it is designed for can supply.

Software engineering is an area of software development in which computer scientists and engineers study methods and tools that facilitate the efficient development of correct, reliable, and robust computer programs. Research in this branch of computer science considers all the phases of the software life cycle, which begins with a formal problem specification, and progresses to the design of a solution, its implementation as a program, testing of the program, and program maintenance. Software engineers develop software tools and collections of tools called programming environments to improve the development process. For example, tools can help to manage the many components of a large program that is being written by a team of programmers.

Algorithms and data structures are the building blocks of computer programs. An algorithm is a precise step-by-step procedure for solving a problem within a finite time and using a finite amount of memory. Common algorithms include searching a collection of data, sorting data, and numerical operations such as matrix multiplication. Data structures are patterns for organizing information, and often represent relationships between data values. Some common data structures are called lists, arrays, records, stacks, queues, and trees.

Computer scientists continue to develop new algorithms and data structures to solve new problems and improve the efficiency of existing programs. One area of theoretical research is called algorithmic complexity. Computer scientists in this field seek to develop techniques for determining the inherent efficiency of algorithms with respect to one another. Another area of theoretical research called computability theory seeks to identify the inherent limits of computation.

Software engineers use programming languages to communicate algorithms to a computer. Natural languages such as English are ambiguous—meaning that their grammatical structure and vocabulary can be interpreted in multiple ways—so they are not suited for programming. Instead, simple and unambiguous artificial languages are used. Computer scientists study ways of making programming languages more expressive, thereby simplifying programming and reducing errors.[7] A program written in a programming language must be translated into machine language (the actual instructions that the computer follows). Computer scientists also develop better translation algorithms that produce more efficient machine language programs.

Databases and information retrieval are related fields of research. A database is an organized collection of information stored in a computer, such as a company's customer account data. Computer scientists attempt to make it easier for users to access databases, prevent access by unauthorized users, and improve access speed. They are also interested in developing techniques to compress the data, so that more can be stored in the same amount of memory. Databases are sometimes distributed over multiple computers that update the data simultaneously, which can lead to inconsistency in the stored information. To address this problem, computer scientists also study ways of preventing inconsistency without reducing access speed.

Information retrieval is concerned with locating data in collections that are not clearly organized, such as a file of newspaper articles. Computer scientists develop algorithms for creating indexes of the data. Once the information is indexed, techniques developed for databases can be used to organize it. Data mining is a closely related field in which a large body of information is analyzed to identify patterns. For example, mining the sales records from a grocery store could identify shopping patterns to help guide the store in stocking its shelves more effectively.

Operating systems are programs that control the overall functioning of a computer. They provide the user with interface, place programs into the computer's memory and cause it to execute them, control the computer's input and output devices, manage the computer's resources such as its disk space, protect the computer from unauthorized use, and keep stored data secure. Computer scientists are interested in making operating systems easier to use, more secure, and more efficient by developing new user interface designs, designing new mechanisms that allow data to be shared while preventing access to sensitive data, and developing algorithms that make more effective use of the computer's time and memory.

The study of numerical computation involves the development of algorithms for calculations, often on large sets of data or with high precision. Because many of these computations may take days or months to execute, computer scientists are interested in making the calculations as efficient as possible. They also explore ways to increase the numerical precision of computations, which can have such effects as improving the accuracy of a weather forecast. The goals of improving efficiency and precision often conflict, with greater efficiency being obtained at the cost of precision and vice versa.

Symbolic computation involves programs that manipulate nonnumeric symbols, such as characters, words, drawings, algebraic expressions, encrypted data (data coded to prevent unauthorized access), and the parts of data structures that represent relationships between values. One unifying property of symbolic programs is that they often lack the regular patterns of processing found in many numerical computations. Such irregularities present computer scientists with special challenges in creating theoretical models of a program's efficiency, in translating it into an efficient machine language program, and in specifying and testing its correct behavior.

Computer Architecture

Computer architecture is the design and analysis of new computer systems. Computer architects study ways of improving computers by increasing their speed, storage capacity, and reliability, and by reducing their cost and power consumption. Computer architects develop both software and hardware models to analyze the performance of existing and proposed computer designs, then use this analysis to guide development of new computers. They are often involved with the engineering of a new computer because the accuracy of their models depends on the design of the computer's circuitry. Many computer architects are interested in developing computers that are specialized for particular applications such as image processing, signal processing, or the control of mechanical systems. The optimization of computer architecture to specific tasks often yields higher performance, lower cost, or both.

Artificial Intelligence

Artificial intelligence (AI) research seeks to enable computers and machines to mimic human intelligence and sensory processing ability, and models human behavior with computers to improve our understanding of intelligence. The many branches of AI research include machine learning, inference, cognition, knowledge representation, problem solving, case-based reasoning, natural language understanding, speech recognition, computer vision, and artificial neural networks.

A key technique developed in the study of artificial intelligence is to specify a problem as a set of states, some of which are solutions, and then search for solution states. For example, in chess, each move creates a new state. If a computer searched the states resulting from all possible sequences of moves, it could identify those that win the game. However, the number of states associated with many problems (such as the possible number of moves needed to win a chess game) is so vast that exhaustively searching them is impractical. The search process can be improved through the use of heuristics—rules that are specific to a given problem and can therefore help guide the search. For example, a chess heuristic might indicate that when a move results in checkmate, there is no point in examining alternate moves.[8]

Robotics

Another area of computer science that has found wide practical use is robotics—the design and development of computer-controlled mechanical devices. Robots range in complexity from toys to automated factory assembly lines, and relieve humans from tedious, repetitive, or dangerous tasks. Robots are also employed where requirements of speed, precision, consistency, or cleanliness exceed what humans can accomplish. Roboticists—scientists involved in the field of robotics—study the many aspects of controlling robots. These aspects include modeling the robot's physical properties, modeling its environment, planning its actions, directing its mechanisms efficiently, using sensors to provide feedback to the controlling program, and ensuring the safety of its behavior. They also study ways of simplifying the creation of controlling programs. One area of research seeks to provide robots with more of the dexterity and adaptability of humans, and is closely associated with AI.

Human-Computer Interfacing

Human-computer interfaces provide the means for people to use computers. An example of a human-computer interface is the keyboard, which lets humans enter commands into a computer and enter text into a specific application. The diversity of research into human-computer interfacing corresponds to the diversity of computer users and applications. However, a unifying theme is the development of better interfaces and experimental evaluation of their effectiveness. Examples include improving computer access for people with disabilities, simplifying program use, developing three-dimensional input and output devices for virtual reality, improving handwriting and speech recognition, and developing heads-up displays for aircraft instruments in which critical information such as speed, altitude, and heading are displayed on a screen in front of the pilot's window. One area of research, called visualization, is concerned with graphically presenting large amounts of data so that people can comprehend its key properties.

Connection of Computer Science to Other Disciplines

Because computer science grew out of mathematics and electrical engineering, it retains many close connections to those disciplines. Theoretical computer science draws many of its approaches from mathematics and logic. Research in numerical computation overlaps with mathematics research in numerical analysis. Computer architects work closely with the electrical engineers who design the circuits of a computer.

Beyond these historical connections, there are strong ties between AI research and psychology, neurophysiology, and linguistics.[9] Human-computer interface research also has connections with psychology. Roboticists work with both mechanical engineers and physiologists in designing new robots.

Computer science also has indirect relationships with virtually all disciplines that use computers. Applications developed in other fields often involve collaboration with computer scientists, who contribute their knowledge of algorithms, data structures, software engineering, and existing technology. In return, the computer scientists have the opportunity to observe novel applications of computers, from which they gain a deeper insight into their use.[10] These relationships make computer science a highly interdisciplinary field of study.

New Words & Expressions

trace one's roots to 追根……到,追溯……到

advent /ˈædvənt/ ***n*.** (尤指不寻常的人或事)出现,到来

sprout /spraut/ ***v*.** 使萌芽
***n*.** 新芽

overlap /ˌəuvəˈlæp/ ***v*.** (与……)交迭,(与……)重叠

interface /ˈintəfeis/ ***n*.** 界面

mass produce 大规模生产

circuitry /ˈsəːkitri/ ***n*.** 电路,线路

revolutionize /ˌrevəˈluːʃənaiz/ ***vt*.** 使革命化,引起革命,引起大变革

microprocessor /ˈmaikrəuˌprəusesə/ ***n*.** [计]微处理器

integral /ˈintigrəl/ ***a*.** 完整的,整体的,构成整体所需要的;[数学]积分的
***n*.** 完整,[数学]积分

speculative /ˈspekjulətiv/ ***a*.** 投机的

viable /ˈvaiəbl/ ***a*.** 能养活的,能生育的;可行的

underlie /ˌʌndəˈlai/ ***vt*.** 位于……之下,成为……的基础

neural /ˈnjuərəl/ ***a*.** 神经系统的,神经中枢的,背的

neuron /ˈnjuərən/ ***n*.** [解]神经细胞,神经元

neurophysiology /ˌnjuərəuˌfiziˈɔlədʒi/ ***n*.** 神经生理学

optimal /ˈɔptiməl/ ***a*.** 最佳的,最理想的

algorithm /ˈælgəriðəm/ ***n*.** [数]运算法则

retrieve /riˈtriːv/ ***v*.** 重新得到,找回

facilitate /fəˈsiliteit/ ***vt*.** 使容易,使便利;推动,帮助,促进

robust /rəuˈbʌst/ ***a*.** 强壮的,坚强的,精力充沛的

implementation /ˌimplimenˈteiʃən/ ***n*.** 执行

finite /ˈfainait/ ***a*.** 有限的,[数]有穷的,限定的

matrix /ˈmætriks/ ***n*.** 矩阵

multiplication /ˌmʌltipliˈkeiʃən/ ***n*.** [数]乘法;增加;(动、植物的)繁殖,增殖

array /əˈrei/ ***n*.** 排列,编队,军队,衣服,大批
***vt*.** 部署,穿着,排列

stack /stæk/ ***n*.** 堆,一堆
***v*.** 堆叠

inherent /inˈhiərənt/ ***a*.** 固有的,内在的,与生俱来的

with respect to 关于……一事,就……事而言

ambiguous /æmˈbigjuəs/ ***a*.** 暧昧的,不明确的

database /ˈdeitəˌbeis/ ***n*.** [计]数据库,资料库

inconsistency /ˌinkənˈsistənsi/ ***n.*** 矛盾
vice versa 反之亦然
manipulate /məˈnipjuleit/ ***vt.*** (熟练地)操作,使用(机器等)
algebraic /ˌældʒiˈbreiik/ ***a.*** 代数的,关于代数学的
encrypt /inˈkript/ ***vt.*** [计]加密,将……译成密码
optimization /ˌɔptimaiˈzeiʃən/ ***n.*** 最佳化,最优化
mimic /ˈmimik/ ***a.*** 模仿的,假装的,[生]拟态的
n. 效颦者,模仿者;小丑;仿制品
vt. 模仿,模拟
sensory /ˈsensəri/ ***a.*** 感觉的,知觉的
cognition /kɔgˈniʃən/ ***n.*** 认识,认知
heuristics /hjuəˈristiks/ ***n.*** 启发式的教学方法
robotics /rəuˈbɔtiks/ ***n.*** 机器人学,机器人技术
dexterity /deksˈteriti/ ***n.*** 灵巧,巧妙;机敏;熟练
visualization /ˌvizjuəlaiˈzeiʃən/ ***n.*** 视觉化,形象化
linguistics /liŋˈgwistiks/ ***n.*** 语言学
collaboration /kəˌlæbəˈreiʃən/ ***n.*** 协作,合作
in return 作为报答
interdisciplinary /ˌintəˈdisipliˌnəri/ ***a.*** 各学科相互间的,科技整合的

Notes

1. **Charles Babbage(1792～1871)**, British mathematician and inventor, who designed and built mechanical computing machines on principles that anticipated the modern electronic computer. Babbage was born in Teignmouth, Devonshire, and was educated at the University of Cambridge. He became a fellow of the Royal Society in 1816 and was active in the founding of the Analytical, the Royal Astronomical, and the Statistical societies. In the 1820s Babbage began developing his Difference Engine, a mechanical device that can perform simple mathematical calculations. Babbage started to build his Difference Engine, but was unable to complete it because of a lack of funding. However, in 1991 British scientists, following Babbage's detailed drawings and specifications, constructed the Difference Engine. The machine worked flawlessly, calculating up to a precision of 31 digits, proving that Babbage's design was sound. In the 1830s Babbage began developing his Analytical Engine, which was designed to carry out more complicated calculations, but this device was never built. Babbage's book *Economy of Machines and Manufactures* (1832) initiated the field of study known today as operational research.
2. **Until the advent of electronic digital computers in the 1940s, computer science was not generally distinguished as being separate from mathematics and engineering.** 直到20世纪40年代电子数字计算机的出现,计算机科学才被公认为从数学和工程学中分离出来。
3. **...to include simplifying the use of computers through programming languages...**
   动词 include 后接动名词,而不可接不定式,本文中多处出现。
4. **The microprocessor made possible the advent of the personal computer, which resulted in an explosion in the use of computer applications.**
   possible 在句中作宾语补足语,由于宾语 the advent of the personal computer 过长,而其后又跟一个非限制性定语从句,故将 possible 前置。
5. **Underlying all of these specific goals is the desire to better the human condition today and in the future through the improved use of information.** 所有这些具体目标的根本,就是通过利用信息来改善今天和将来人类状况的欲望。
   此句是由现在分词 underlying 引导的完全倒装句,表示强调,正常语序应为:The desire to better the human condition today and in the future through the improved use of information is underlying all of these specific goals.
6. **But when computer systems and their interactions with the natural world become sufficiently complex, unforeseen behaviors can result.**
   注意 result 一词在此处的用法,意为"结果(发生……的结果)"。又如:
   If we announce the news, a panic will result. 这条消息一经发布,定会造成恐慌。

7. Computer scientists study ways of making programming languages more expressive, thereby simplifying programming and reducing errors.

thereby一词为正式用语,作"by that, by means of that"解,意为"借以,因此"。此处与分词 simplifying 连用,作结果状语。又如:

(1) Our bodies can sweat, thereby losing heat by evaporation.我们的身体可以出汗,借蒸发失去热量。

(2) A firm might sometimes sell at a loss to drive a competitor out of business, and thereby increases its market power.一个公司也许有时亏本销售以使竞争者破产,借以提高其市场能力。

8. For example, a chess heuristic might indicate that when a move results in checkmate, there is no point in examining alternate moves.

There is no (or not much) point in doing sth.意为"做某事是没有什么意义的",介词 in 有时可省略。又如:

There's not much point in arguing any further.再议论下去也没有多大意义。

9. Beyond these historical connections, there are strong ties between AI research and psychology, neurophysiology, and linguistics.

beyond 作"apart from"解,意为"除……之外"。又如:

He appears to have no personal staff, beyond a secretary who can't make coffee.除了一个不会泡咖啡的秘书外,他似乎没有私人职员。

10. In return, the computer scientists have the opportunity to observe novel applications of computers, from which they gain a deeper insight into their use.

in retrun (for) 意为"作为……的报酬,作为回报"。又如:

(1) I presented him with a picture, and he gave me his photograph in return.我送给他一幅图画,他回赠我一张个人照片。

(2) He didn't expect anything in return for his help.他帮助别人并不期待任何回报。

Exercises

Ⅰ. ***Answer the following questions according to the text.***

1. What is computer science concerned with?
2. Discuss briefly the development of computer science.
3. What are the goals of computer science?
4. What are the key parts of computer science?
5. How many fields does computer science fall into? And what are they?
6. What are the building blocks of a computer program?
7. Why are natural languages such as English not suited for programming?
8. What is meant by a database?
9. What is, by definition, computer architecture?
10. What are the connections of computer science to other disciplines?

Ⅱ. ***Translate the following paragraphs into English.***

目前大多数计算机销售商将他们的计算机归入第四代计算机,少数则属于第五代。前三代计算机是以电子学技术的重大突破为标志,即依次使用电子管、晶体管和集成电路器件。有些人主张把推出大规模集成电路(单元空间上有更多的电路)的 1971 年作为第四代计算机的起点,但是另外一些计算机的设计者则对此有争议,他们认为如果承认这一前提,那么在 1971 年以后就应有第五代、第六代,甚至第七代计算机。

当代计算机的基本技术仍然是集成电路,但这并不是说这二十年来没有任何重大革新。事实上,计算机工业在电路上的进一步小型化,数据通信、计算机硬件和软件的设计以及输入、输出设备等方面都经历了连续的令人眼花缭乱的进展。

微处理机是对第四代计算机的问世的最重要的贡献之一。一块硅片便可包容的微处理器是一种微型化电子电路产品。第一个完全能运行的微处理器,有时也称单片机,是 1971 年问世的。现在全世界拥有的微处理器的数量已超过人口数。

Part B

Computer (1)

Introduction

Computer is a machine that performs tasks, such as mathematical calculations or electronic communications, under the control of a set of instructions called a program. Programs usually reside within the computer and are retrieved and processed by the computer's electronics, and the program results are stored or routed to output devices, such as video display monitors or printers. [1]Computers are used to perform a wide variety of activities with reliability, accuracy, and speed.

Uses of Computers

People use computers in a wide variety of ways. In business, computers track inventories with bar codes[2] and scanners, check the credit status[3] of customers, and transfer funds electronically. In homes, tiny computers embedded in the electronic circuitry of most appliances control the indoor temperature, operate home security systems, tell the time, and turn videocassette recorders on and off. Computers in automobiles regulate the flow of fuel, thereby increasing gas mileage. Computers also entertain, creating digitized sound on stereo systems or computer-animated features from a digitally encoded laser disc. Computer programs, or applications, exist to aid every level of education, from programs that teach simple addition or sentence construction to advanced calculus. Educators use computers to track grades and prepare notes; with computer-controlled projection units, they can add graphics, sound, and animation to their lectures (See Computer-Aided Instruction). Computers are used extensively in scientific research to solve mathematical problems, display complicated data, or model systems that are too costly or impractical to build, such as testing the airflow around the next generation of space shuttles[4]. The military employs computers in sophisticated communications to encode and unscramble messages, and to keep track of personnel and supplies. [5]

How Computers Work

The physical computer and its components are known as hardware. Computer hardware includes the memory that stores data and instructions; the central processing unit (CPU)[6] that carries out instructions; the bus that connects the various computer components; the input devices, such as a keyboard or mouse, that allow the user to communicate with the computer; and the output devices, such as printers and video display monitors, that enable the computer to present information to the user. The programs that run the computer are called software. Software generally is designed to perform a particular type of task—for example, to control the arm of a robot to weld a