

专门用途英语(ESP)规划教材系列一
ESP Textbook Series I

计算机英语

Computer English

● 张国安 主编 ●

- 中国大学英语教学改革的方向
- 中国社会经济发展的必然要求



苏州大学出版社
Soochow University Press

专门用途英语(ESP)规划教材系列一
ESP Textbook Series I

计算机英语

Computer English

张国安 主编

- 中国大学英语教学改革的方向
- 中国社会经济发展的必然要求



苏州大学出版社
Soochow University Press

图书在版编目(CIP)数据

计算机英语=Computer English/张国安主编. —苏州:
苏州大学出版社, 2009. 7
(专门用途英语(ESP)规划教材系列一)
ISBN 978-7-81137-283-0

I. 计… II. 张… III. 电子计算机—英语—高等学校—
教材 IV. TP3

中国版本图书馆 CIP 数据核字(2009)第 129956 号

计算机英语 Computer English 张国安 主编 责任编辑 汤定军

苏州大学出版社出版发行
(地址: 苏州市干将东路 200 号 邮编: 215021)
丹阳市教育印刷厂印装
(地址: 丹阳市西门外 邮编: 212300)

开本 787 mm×960 mm 1/16 印张 18.75 字数 344 千
2009 年 7 月第 1 版 2009 年 7 月第 1 次印刷
ISBN 978-7-81137-283-0 定价: 32.00 元

苏州大学版图书若有印装错误, 本社负责调换
苏州大学出版社营销部 电话: 0512-67258835
苏州大学出版社网址 <http://www.sudapress.com>

《计算机英语》编委会

主 编：张国安

副主编：任丽鸿

编 委：王明秀 吴秀升 薛醒思

策 划：汤定军

出版者的话

Publisher's Words

随着社会的发展,交流领域的进一步扩大,人们对英语的需求范围也在不断增大。在市场经济环境下,传统的英语教学与社会需求之间存在着越来越大的差距,其内容、理念等不断受到冲击和挑战,学了很多年英语的大学毕业生无法在工作实践中学以致用。因此,高校英语教学必须做出较大的改革,以适应社会需求的不断变化,在此背景下专门用途英语(ESP)的出现成为必然。可喜的是,专门用途英语教学虽然起步较晚,但发展很快,现在全国已有不少高校开设了专门用途英语课程。

1. 专门用途英语是英语教学发展的必然

英语教学和社会的发展密不可分。英语教学是一项社会活动,随着社会的发展,教学活动的内容和方法也在不断地改进。而专门用途英语的出现是社会发展的需要,也是英语教学改革的前提和目标。在中国,英语教学经历了不同的阶段,从最初的以阅读为基础的语法翻译法,到以口语教学为主的交际法,尔后人们渐渐明白在各个领域中英语的用法存在着较大的差异,随之英语教学便转向专门用途英语。

2. 专门用途英语是语言学理论发展的产物

回顾语言学的历史,我们发现,最初人们是在语言的范畴内探讨语言现象,而没有把语言与社会联系起来。随着社会的发展,人们突破了这一樊篱,语言的共时研究和历时研究为社会语言学的应运而生打下了基础,而社会语言学的兴起为专门用途英语的产生和发展提供了理论基础。索绪尔认为,研究语言就要研

究语言的交际性和功能性。专门用途英语是英语的一种功能变体,是特定的社会文化群体使用的专门语言。显然,专门用途英语的产生和发展得益于社会语言学的理论观照。

3. 专门用途英语的存在有其教学依据

1994年,原国家教委制定了高等院校面向21世纪教学内容和课程体系改革计划。该计划对21世纪的中国外语人才提出了新要求:基本功扎实,知识面宽广,有一定的相关专业知识,有较强的能力和较高的素质,成为“外语+专业”的复合型外语人才。最近几年,国家教育部门对专门用途英语的发展更加关注,出台了许多措施,鼓励各级高校有针对性地开设社会需求比较大的专门用途英语课程。

专门用途英语教学是社会发展的必然,是普通用途英语教学的扩展和延续,是英语语言文学教学贴近经济社会、培养实用型英语人才的一大变革。

基于以上情况,苏州大学出版社适时地推出《专门用途英语(ESP)规划教材系列》,以便更好地满足社会、经济发展的需要,更好地服务于中国的英语教学改革,为英语学习者以及英语爱好者提供更好的专门用途英语教材。苏州大学出版社希望借此能为推动中国的英语教学改革和发展作出自己的贡献。

《专门用途英语(ESP)规划教材系列一》包括《法律英语》、《旅游英语》、《体育英语》、《计算机英语》、《经济英语》和《机电英语》。该套教材的编写者是全国专门用途英语教学的一线老师或相关学者,他们具有渊博的学识和丰富的教学经验,在各自的领域里都是佼佼者。该套教材不仅适用于全国普通高等院校、高职高专院校的大学生,也适用于开设相关专业英语的研究生,而对于那些对专门用途英语感兴趣的人士也是很好的参考资料。

《专门用途英语(ESP)规划教材系列一》具有以下特点:

1. 专业知识与案例分析相结合

该套教材把专业知识与实践紧密地结合起来,先让学生对某一领域的专业知识有系统的了解,再通过案例分析等情景模式让学生把专业知识更好地融入现实,从而有针对性地开展英语交流,达到既巩固专业知识又提高英语表达能力的目的。这种结合可以克服纯粹专业知识学习的枯燥,调动学生用英语交流的积极性,从而实现英语教学的交际性和功能性。

2. 严谨性与趣味性相结合

教材编写的宗旨是让读者掌握一定的知识、传达一定的信息,因而教材的内容和结构必须严谨。该套教材的编写秉承这一原则,不仅前后结构安排严密、逻辑性强,同时又融入了趣味性和现代气息,将笑话、漫画等表达手段很好地穿插其中。

3. 工整与活泼相结合

该套教材的编写理念从策划开始就强调借鉴国外原版教材的活泼与丰满,但又不放弃中国传统的工整、对称等审美情趣,因而整体看来实现了既活泼又工整的效果。教材中的英文采用了国际上的流行字体,中文采用了扁宋和楷体;每一个图标既体现了章节的变化有度,又能很好地表现各章节的特点。

在该套专门用途英语教材即将问世之际,我们希望它们能够很好地服务于中国英语教学改革、推动中国英语教学的发展。各位读者如有什么疑问或建议,请联系我们(tangdingjun@suda.edu.cn),对于你们的指导我们倍加感谢。

苏州大学出版社

2009. 2. 2



前言

Preface

本书力图向读者介绍计算机科学的全貌,即在计算机环境下如何解决问题和有效地完成任务。由于计算机知识的多样性,本书无法只介绍其中某一方面的问题,而是尽可能地描述计算机科学的发展,读者从中可以看清计算机发展的脉络。

本书覆盖了离散数学、算法、编程方法学、面向对象程序设计、操作系统、科学计算这样一些传统意义上的计算机科学理论,也涉及到计算机图形学、多媒体、人工智能、计算机网络这些新议题,更加进了述录系统、语音技术、网络安全、应用程序、微控制器这样一些应用实例。这样的编排是为了拓展读者的视野,从中发现自己感兴趣的问题。

与其他计算机英语教科书不同的是,本书是按照简单的全反应式模型来编写的,与其说这是一本英语书,不如说这是用英语编写的介绍计算机科学的引论。本书不再介绍英语语法,但教师还是应当对复杂的语法现象做深入分析,这既是理解上下文的需要,也是提高英语阅读能力所必须的。从学习者的角度来看,这使解释更清晰,读者把握内容更全面。从教学的角度看,这样的做法为教师提供了一种自然的循序渐进的教学方法,从而讲授那些难于掌握的材料。

本书分成 20 个单元,每个单元包含 3 篇文章,Text A 和 Text B 给出了一些关键词汇和短语,对较为深入的问题作了注解,并安排少量问题。读者除了用英文完成那些问题外,还应自主查找相关资料。这些问题有些涉及计算机的专门知识,有些是为了提高英语应用水平。使用本书并不需要计算机科学基本知识以外的其他预先知识。当然,已经学了一些计算机基础知识会有助于理解本书中一些实际应

用方面的材料。

全书由福建工程学院张国安、东营职业学院任丽鸿编写,福建工程学院王明秀、吴秀升、薛醒思老师提供了宝贵意见,在此表示由衷的谢意。读者如需要练习参考答案等信息,请联系我们(jvtczga@yahoo.com.cn 或 tangdingjun@suda.edu.cn)。

张国安 任丽鸿

2009 年 6 月

Contents

Unit 1	Computer Science	[1]
Unit 2	Discrete Mathematics for Computer Science	[13]
Unit 3	Algorithms in the “Real World”	[29]
Unit 4	Dictation Systems	[42]
Unit 5	Programming Methodology	[53]
Unit 6	Coding Guidelines	[69]
Unit 7	Object -Oriented Systems Design	[96]
Unit 8	Operating Systems and Systems Programming	[110]
Unit 9	Introduction to Scientific Computing	[127]
Unit 10	Introduction to Computer System	[146]
Unit 11	Speech Technology Research, Development, and Deployment	[157]
Unit 12	Computer Language and Programming	[176]
Unit 13	Computer Graphics and Image	[186]
Unit 14	Multimedia Technology	[197]
Unit 15	Research Project in Artificial Intelligence	[207]

Unit 16	Software That Enables You to Author Intelligent Tutor Behavior	[225]
Unit 17	Computer Network Technology	[242]
Unit 18	Computer and Network Security	[252]
Unit 19	Computer Application	[263]
Unit 20	Foundation and Application of Microcontroller	[273]

Unit 1

Computer Science

Text A

Research in Computer Science

Computer Science at MIT (Massachusetts Institute of Technology) is a bit different from programs offered at most other universities. Many potential students really don't understand what it's all about here. It's not about "applications programming", or "data processing" or "web design". "Systems analyst" is not the sort of job to which our students aspire after they graduate^[1] (by sophomore year it's easy to get on-campus jobs keeping systems running). At MIT one starts out by learning about dealing with complexity, modelling and abstraction, and goes on to study computer architecture (how to design computer systems), artificial intelligence, modelling, and theory. There is quite a bit of advanced mathematics. Computer Science studies how to make computers faster, more efficient, and more intelligent. Graduates of Computer Science have the ability to perform a wide range of jobs. The breadth of their studies allows them to learn quickly the variety of languages and machines used in industry. The depth of study gives a better understanding of the problem and how to solve it.^[2] The curriculum is as intensive as electrical engineering. You can't get a big jump on the major by studying programming in high school. Our introductory CS subject is taught in PYTHON^[3], and the software engineering lab uses a version of Java^[4]; C++ can also be useful in UROPs (Undergraduate Research Opportunities Programs).

Research in Computer Science at MIT covers many topics. Here are a few:

- Robots design and control;
- Making machines hear, see, move, and learn;
- The use of intelligent machines as resources for making decisions;
- The use of computers in education;
- Using computers to design chips and other things;
- Making computers easier to use;
- Determining if a problem is solvable;
- Making better programming languages;
- Automating offices and other functions;
- Getting computers to talk to each other;
- Applied probability;
- Medical decision-making;
- Computer graphics.

Graduates in Computer Science get jobs developing systems for science, business or fun. Some join large companies like Google, Microsoft, Oracle, or IBM; others join or start small start-up companies to be “in on the ground floor”. Some design video games, others work on supercomputers or robots. You will find MIT Computer Science graduates behind the design and development of most computers on the market. Many graduates continue on to get advanced degrees, not only technical but also medical, legal, business, etc. ^[5]

What Is UROP?

The UROP cultivates and supports research partnerships between MIT undergraduates and faculty. One of the earliest programs of its kind in the United States, MIT's UROP invites undergraduates to participate in research as the junior colleagues of Institute faculty. The late Margaret L. A. MacVicar, Professor of Physical Science and Dean for Undergraduate Education, created MIT's UROP in 1969, inspired by Edwin H. Land. Land, the inventor of instant photography, believed in the power of learning by doing.

UROP offers the chance to work on cutting-edge research—whether you join established research projects or pursue your own ideas. As UROPers, undergraduates participate in each phase of standard research activity: developing research plans, writing proposals, conducting research, analyzing data and

presenting research results in oral and written form.^[6] UROP projects take place during the academic year, as well as over the summer, and research can be done in any academic department or interdisciplinary laboratory. Projects can last for an entire semester, and many continue for a year or more. UROP students receive academic credit, pay or work on a voluntary basis. MIT students use their UROP experiences to become familiar with the faculty, learn about potential majors, and investigate areas of interest. UROPers gain practical skills and knowledge they eventually apply to careers after graduation or as graduate students. Most importantly, they become involved in exciting research!

What Kind of Computer Should I Buy to Bring with Me?

Unlike lots of other colleges, MIT does not require that its students buy or own a personal computer. There are lots of big, powerful computers here (many thousands of networked computers in all areas of the campus). Computer time for word processing and play is totally available. LINUX is the operating system of choice for many students while others use Microsoft and Apple systems.^[7] If you still want to purchase one, you should know that MIT has made arrangements for discounts for the entire MIT community with various companies. Don't worry too much about compatibility with MIT computers and software; make your best deal.

(From: <http://www.mit.edu>)



Words and expressions

aspire	vi.	热望, 向往; 怀有大志
sophomore	adj.	(大学、高中) 二年级的, 二年级学生的
architecture	n.	结构, 构造
curriculum	n.	(学校等的) 全部课程
robot	n.	机器人; 自动控制装置; 遥控装置
cultivate	vt.	培养; 陶冶; 鼓励; 建立
faculty	n.	(大学的) 系, 科, 院; 【主美】(大学或院系的) 全体教职员
pursue	v.	追求; 进行; 从事; 继续

interdisciplinary	adj.	各学科间的,跨学科的
career	n.	(终身的)职业
start out		着手
participate in		参与,参加
become familiar with		熟悉
become involved in		涉猎;参与

Notes

1. "Systems analyst" is not the sort of job to which our students aspire after they graduate ... 学生毕业后并不向往系统分析员这类职位……
aspire to 是“向往”的意思,本句中中介词 to 前置于定语从句连接词 which 前面。
2. The breadth of their studies allows them to learn quickly the variety of languages and machines used in industry. The depth of study gives a better understanding of the problem and how to solve it. 学习的广度让学生可以快速掌握计算机行业中使用的各种语言和机器,学习的深度让学生更深入地理解各种问题及其解决办法。
3. PYTHON 一种程序设计语言,见 Text B。
4. Java 典型的面向对象程序设计语言,见 Unit 7。
5. Many graduates continue on to get advanced degrees, not only technical but also medical, legal, business, etc. 许多毕业生继续深造,不仅在计算机方面,也在医学、法律、商业等方面获取更高学位。
6. As UROPers, undergraduates participate in each phase of standard research activity: developing research plans, writing proposals, conducting research, analyzing data and presenting research results in oral and written form. 作为 UROP 项目的参与者,本科生参加到常规的研究活动中,包括提出研究计划、写建议书、开展研究、分析数据、以口头或书面形式提交研究成果。
7. LINUX is the operating system of choice for many students while others use Microsoft and Apple systems. 很多学生选择 LINUX(一种多用户、多任务环境的 UNIX 型操作系统)作为操作系统,也有一些学生使用微软或苹果公司的操作系统。



Exercises

1. According to Text A, write a paragraph in 100 words to give your opinion about the following topics:
 - ◇ In the 13 fields described in Text A, is there any one interesting to you?
 - ◇ How do you think about UROP?
 - ◇ Have you bought a computer? What is the OS running on it? What will you do by using it?
2. Select appropriate prepositions to fill in the following spaces:
 - ◇ He started _____ to write a novel.
 - ◇ This book deals _____ an important issue.
 - ◇ How many people participated _____ the opening ceremony?
 - ◇ Jim believes _____ fresh air and morning exercises.
 - ◇ He was involved _____ writing his doctoral dissertation.
 - ◇ The chairman was quite familiar _____ the procedure for conducting a meeting.
 - ◇ We have already made arrangements _____ our vacation.
 - ◇ Don't worry _____ him; he is a cheerful and elastic man.

Text B

PYTHON

Python is an easy-to-learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python website,

<http://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third-party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

For a description of standard objects and modules, see the Python Library Reference document. The Python Reference Manual gives a more formal definition of the language. To write extensions in C or C++, read Extending and Embedding the Python Interpreter and Python/C API Reference. There are also several books covering Python in depth.

This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style.^[1] After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in the Python Library Reference.

If you do much work on computers, eventually you find that there's some task you'd like to automate. For example, you may wish to perform a search-and-replace over a large number of text files, or rename and rearrange a bunch of photo files in a complicated way. Perhaps you'd like to write a small custom database, or a specialized GUI (Graphic User Interface) application, or a simple game.

If you're a professional software developer, you may have to work with several C/C++/Java libraries but find the usual write/compile/test/re-compile cycle is too slow. Perhaps you're writing a test suite for such a library and find writing the testing code a tedious task. Or maybe you've written a program that could use an extension language, and you don't want to design and implement a whole new language for your application.