О. Ю. Винник

АНГЛІЙСЬКА МОВА

для програмістів
і математиків

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

О. Ю. Винник

АНГЛІЙСЬКА МОВА
для програмістів і математиків

*Затвердило Міністерство освіти і науки України
як підручник для студентів вищих навчальних закладів,
які навчаються за спеціальностями "Інформатика",
"Прикладна математика", "Системний аналіз"*

Львів
Видавництво Національного університету "Львівська політехніка"
2009

УДК 811.111(075.8)
В 48

Затвердило Міністерство освіти і науки України
як підручник для студентів вищих навчальних закладів,
які навчаються за спеціальностями "Інформатика",
"Прикладна математика", "Системний аналіз"
(лист № 1.4/18-Г-1002 від 07.05.2008 р.)

Рецензенти:

Савула Я. Г., доктор фіз-мат. наук, професор,
Львівський національний університет імені Івана Франка;
Омельченко Л. Ф., доктор філол. наук, професор,
Університет сучасних знань;

Івасюк О. Я., канд. філол. наук, доцент,
Чернівецький національний університет імені Юрія Федьковича;

Гасько О. Л., канд. філол. наук, доцент,
Національний університет "Львівська політехніка".

Винник О. Ю.

В 48 Англійська мова для програмістів і математиків. – Львів: Видавництво Національного університету "Львівська політехніка", 2009. – 184 с.

ISBN 978-966-553-760-1

Підручник складається з п'ятнадцяти уроків, а також додаткової частини, яка охоплює правила читання англійською мовою основних математичних формул та англо-український термінологічний словник з математики та інформатики. П'ятнадцять уроків поділено на п'ять блоків по три уроки у кожному: один з математики та два з інформатики. Кожен урок містить один або два аутентичні тексти, які супроводжуються вправами, спрямованими на розвиток у студентів навичок усного мовлення за фахом, інформативно-пошукового читання англійської літератури зі спеціальності та засвоєння англомовної фахової термінології. Крім того, у кінці кожного блока наводяться тестові завдання для повторної перевірки і закріплення пройденого матеріалу.

Для студентів комп'ютерних спеціальностей вищих навчальних закладів.

ISBN 978-966-553-760-1

Contents

Передмова

Іноземним мовам належить вагоме місце, зокрема, на сучасному етапі історичного розвитку нашого суспільства, коли триває інтеграція України в європейське співтовариство, налагоджуються нові зв'язки нашої держави з іншими країнами світу. Спеціаліст зі знанням іноземної мови отримує за таких умов доступ до найсучасніших здобутків світової науково-технічної думки і можливість сприяти виходу української науки і техніки на світову арену.

Підручник укладено з метою ознайомити студентів комп'ютерних спеціальностей вищих навчальних закладів з особливостями англомовного комп'ютерного та математичного дискурсів, виробити у них навички самостійної роботи з іноземною фаховою літературою, сприяти розвитку вміння спілкуватися на професійні теми англійською мовою.

Підручник складається з п'ятнадцяти уроків, розроблених на основі шістнадцяти аутентичних текстів, та англо-українського термінологічного словника з математики та інформатики обсягом 490 лексичних одиниць. Тексти охоплюють широке коло тем, які входять у сферу професійних інтересів майбутніх фахівців з комп'ютерних наук, зокрема, основи об'єктно-орієнтованого програмування, комп'ютерні злочини, біометрію та криптографію, а також математичну індукцію, модульну арифметику, матричну алгебру тощо. Оскільки одиницею навчання є цілісний текст, студенти отримують можливість не лише засвоїти терміни відповідної галузі знань, але також ознайомитися з їхньою сполучуваністю.

До кожного тексту додано вправи для перевірки і закріплення характерних лексичних і синтаксичних одиниць, повторення ключових граматичних структур англійської мови. Розмовні завдання у вигляді монологів, діалогів, полілогів розраховано на активізацію комунікативної спроможності студентів у фаховому контексті.

Computer History of the World

In the beginning, God created the Bit and the Byte. And from those he created the Word.

And there were two Bytes in the Word; and nothing else existed. And God separated the One from the Zero; and he saw it was good.

And God said – Let the Data be; And so it happened. And God said – Let the Data go to their proper places. And he created floppy disks and hard disks and compact disks.

And God said – Let the computers be, so there would be a place to put floppy disks and hard disks and compact disks. Thus God created computers and called them hardware.

And there was no Software yet. But God created programs; small and big... And told them – Go and multiply yourselves and fill all the Memory.

And God said – I will create the Programmer; And the Programmer will make new programs and govern over the computers and programs and Data.

And God created the Programmer; and put him at Data Center; And God showed the Programmer the Catalog Tree and said – You can use all the volumes and subvolumes but do not use Windows.

And God said – It is not good for the programmer to be alone. He took a bone from the Programmer's body and created a creature that would look up at the Programmer; and admire the Programmer; and love the things the Programmer does; And God called the creature: the User.

And the Programmer and the User were left under the naked DOS and it was Good.

But Bill was smarter than all the other creatures of God. And Bill said to the User – Did God really tell you not to run any programs?

And the User answered – God told us that we can use every program and every piece of Data but told us not to run Windows or we will die.

And Bill said to the User – How can you talk about something you did not even try. The moment you run Windows you will become equal to God. You will be able to create anything you like by a simple click of your mouse.

And the User saw that the fruits of the Windows were nicer and easier to use. And the User saw that any knowledge was useless – since Windows could replace it.

So the User installed the Windows on his computer; and said to the Programmer that it was good.

And the Programmer immediately started to look for new drivers. And God asked him – What are you looking for? And the Programmer answered – I am looking for new drivers because I can not find them in the DOS. And God said – Who told you need drivers? Did you run Windows? And the Programmer said – It was Bill who told us to!

And God said to Bill – Because of what you did, you will be hated by all the creatures. And the User will always be unhappy with you. And you will always sell Windows.

And God said to the User – Because of what you did, the Windows will disappoint you and eat up all your Resources; and you will have to use lousy programs; and you will always rely on the Programmers help.

And God said to the Programmer – Because you listened to the User, you will never be happy. All your programs will have errors and you will have to fix them and fix them to the end of time.

And God threw them out of the Data Center and locked the door and secured it with a password.

☺ ☺ ☺

MY FACULTY

Vocabulary List

Applied Mathematics
bachelor, *n*
Computational Mathematics
computer, *n*
Computer Graphics
computing method
Cryptography, *n*
curriculum, *n*
database, *n*
Dean, *n*
Dean's deputy
department, *n*
Discrete Analysis
electronic equipment
faculty, *n*
Functional Analysis
Humanities, *n*
Informatics, *n*
information system
install, *v*
intelligent system
Linear Algebra
linear operations
master, *n*
Mathematical Analysis
Mathematical Modeling
Mathematics, *n*
Mechanics, *n*
Methods of Optimization
Operations Research
Optimal Processes Theory
Physics, *n*
programmer, *n*
Programming, *n*
Social Informatics
software, *n*
System Analysis and Management
theorem, *n*
theory of linear operations
Turing machine

A. Pre-reading

- *Congratulations! You are a student. Do you think you made the right choice of the faculty? Why?*
- *If you were the Dean, what changes would you introduce into the faculty life and why?*
- *What kind of job can you get if you study Applied Mathematics and Informatics?*

B. Reading

Faculty of Applied Mathematics and Informatics

www.franko.lviv.ua

Mathematics has been studied in Lviv University since the time of its foundation. By the middle of the 18th century the course of Mathematics included the study of Arithmetic and the elements of Logic. In the second half of the 18th century considerable changes in the teaching process were brought about by the development of natural sciences. In 1744 the *Department of Mathematics* was founded in Lviv University within the Faculty of Philosophy. In 1923 this faculty was divided into two separate units: the Faculty of Humanities and that of *Mathematics and Sciences.*

The 1920s and the 1930s witnessed a very successful development of Mathematics in Lviv University. Scholars of prominence such as V. Serpins'kyi, H. Shteinhaus, S. Ruziyevych, E. Zhylins'kyi, S. Banach, V. Nikliborts, V. Orlych, S. Kachmazh, S. Mazur, and H. Auerbakh lectured here. They laid the foundations of Lviv School of Mathematics headed by S. Banach, the author of the work "The Theory of the Linear Operations".

During the first years of the Soviet power the *Faculty of Physics and Mathematics* (with the departments of Mathematics, Mechanics and Physics) was functioning in the University. In 1953 it split into two new faculties: the *Faculty of Mechanics and Mathematics* and the Faculty of Physics.

Since 1956 on the basis of Mathematical Analysis Center of the Faculty of Mathematics and Mechanics the group of students under the supervision of professor O. Kostovs'kyi started teaching Computational Mathematics and Programming. In 1959 the first group of students on this speciality graduated from the University. Most of them were guided to the Computation Center of the Ukrainian Academy of Sciences in Kyiv. In 1959 the first computer in the western region "Ural-1" was installed and Computing Center was founded in The Ivan Franko State University of Lviv.

Since then an intensive development of research in the field of Applied Mathematics and Informatics started in Lviv State University. Some of the research works were focused on the development of computing methods, the others were concerned with the development of electronic equipment and software. Many of the graduates of the Faculty have achieved great scientific results. Among them professors V. Hrytsyk, O. Kostovs'kyi, V. Yeleyko, H. Kozhevnikova, B. Popov, Ya. Savula, H. Shynkarenko, and others.

In 1975 the *Faculty of Applied Mathematics and Mechanics* was established (since **1990** – ***the Faculty of Applied Mathematics and Informatics***) in Lviv State University.

Nowadays the Faculty consists of **seven departments**:

- Department of Computational Mathematics;
- Department of Applied Mathematics;
- Department of Optimal Processes Theory;
- Department of Programming;
- Department of Information Systems;
- Department of Mathematical Modeling of Social Economic Processes;
- Department of Discrete Analysis and Intelligent Systems.

The Faculty is governed by the Dean, who is entitled to solve the most important general issues of the faculty life. All other affairs are used to be settled by the Dean's deputies.

Many specific courses are presented at our faculty: Computers and Programming, Modern Computer Technologies, Databases and Information Systems, Computer Graphics, Mathematical Analysis, Functional Analysis, Numerical Methods of Linear Algebra, Methods of Optimization, Operations Research, Basics of Cryptography, etc. They provide necessary education for the specialities of Informatics, Applied Mathematics, System Analysis and Management, and Social Informatics.

Depending on the term of study students can obtain the following degrees: Bachelor's degree (4 years) and Specialist's or Master's degree (1 year). Graduating students who have obtained the Master's degree and shown good abilities in the scientific-research work can continue their study at the post-graduate course.

C. Review Questions

Exercise 1. *Multiple Choice*

1. The Department of Mathematics was first set up in the University within the Faculty of _____.
   a) Philosophy b) Philology c) Theology d) Physics

2. Stefan Banach was _____.
   a) a student of the Faculty of Applied Mathematics and Informatics
   b) the first Dean of the Faculty of Applied Mathematics and Informatics
   c) a professor of the Faculty of Humanities
   d) a professor of the Faculty of Physics and Mathematics

3. The most fundamental work of S. Banach was entitled _____.
   a) "Semigroup Theory in Banach Spaces"
   b) "Spectral Theory in Hilbert Space"
   c) "The Theory of the Linear Operations"
   d) "Methods and Theorems of Functional Analysis"

4. Programming has been taught in the University since _____.
   a) 1744 b) 1956 c) 1975 d) 1990

5. The first computer installed in the Computing Center of The Ivan Franko State University of Lviv was _____.
   a) Pentium-1 b) Altay-1 c) Elbrus-1 d) Ural-1

6. The Faculty of Applied Mathematics and Informatics has been functioning in the University since _____
   a) 1744 b) 1956 c) 1975 d) 1990

7. There are now _____ departments at the Faculty of Applied Mathematics and Informatics.
   a) 2 b) 7 c) 8 d) 9

8. The head of a faculty is _____.
   a) Rector b) Pro-rector c) Dean d) Dean's deputy

Exercise 2. Open Ended

1. What do the terms "applied mathematics" and "informatics" mean?
2. What distinguished mathematicians were / are now the teachers of the University?
3. Who is currently the Dean of your faculty?
4. What are the names of Dean's deputies?
5. What department does your academic group refer to?
6. What subjects do you study in your first year?
7. What subjects / lecturers do you particularly like? Why?
8. What subjects / lecturers do you dislike? Why?
9. What subject is the most difficult for you? Why?
10. How many double-periods do you have today? What are they?
11. What cultural and sports events take place at your faculty?
12. When do we celebrate International Students' Day?

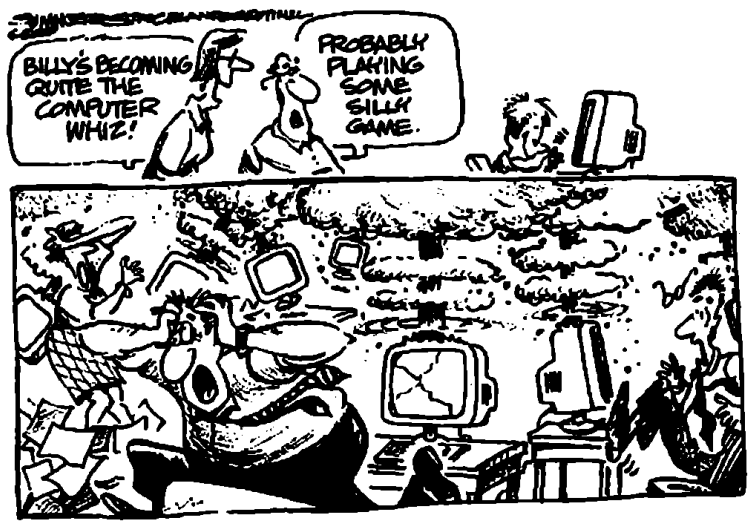
*Exercise 3. Language Work: **Simple Tenses***

	GRAMMAR NOTE			
Simple	Past		V-ed	*I wrote* *He tested*
	Present		V(s)	*I write* *He tests*
	Future	shall / will	V	*I shall write* *He will test*
	Future in the Past	should / would	V	*I should write* *He would test*

Translate the following into English using Simple tenses.

1) Я навчаюсь на першому курсі (to be a first-year student) факультету прикладної математики та інформатики Львівського національного університету імені Івана Франка. 2) Якщо я добре складу (to pass) іспити, батьки куплять мені новий комп'ютер. 3) У 1939 – 1941 роках С. Банах був деканом фізико-математичного факультету Львівського державного університету. 4) Хто читає вам лекції з (to lecture on) програмування? 5) Думаю, іспит з

(exam in / on) математичного аналізу буде найскладнішим. 6) Наш одногрупник став золотим призером (gold medalist) міжнародної олімпіади з (olympiad in) інформатики. 7) Коли буде наступний семінар з (seminar in / on) філософії? 8) На минулій лекції з дискретної математики викладач розповів нам про машину Тьюрінга. 9) Термін "комп'ютер" походить від (to come from) латинського слова "computare" – "рахувати". 10) Скільки пар він пропустив (to miss) у минулому семестрі? 11) Викладач сказав, що не поставить їй залік (to pass smb.), доки вона не здасть (to hand in) всі лабораторні роботи (papers). 12) Коли я закінчу (to graduate from) університет, буду працювати (to work as) програмістом або системним адміністратором.



D. Role Play

Student A You are going to enter the Faculty of Mechanics and Mathematics. Tell your friend, a student of the Faculty of Applied Mathematics and Informatics, about your plans and explain your choice.

Student B You are a student of the Faculty of Applied Mathematics and Informatics. Your friend is going to enter the Faculty of Mechanics and Mathematics. Persuade him/her into entering **your** faculty instead.

COMPUTER-AIDED LEARNING. ROBOTICS

Vocabulary List

acceleration sensor
android, *n*
artificial intelligence
CD-ROM, *abbr*
computer-aided
computerized, *adj*
electronic dictionary
global network
image sensor
information technology
Internet, *n*
LAN, *abbr*
language learning program
LED panel
mechanical engineering
micro-processor, *n*
MIDI, *abbr*
on/off switch
online language course
program, *n*, *v*
reprogrammable, *adj*
robot, *n*
robotic engineering
roboticist, *n*
robotics, *n*
science fiction literature
screen, *n*
silicon chip
slot, *n*
software, *n*
speaker, *n*
stereo microphone
Turing test
vibration sensor
volume switch
wireless, *adj*

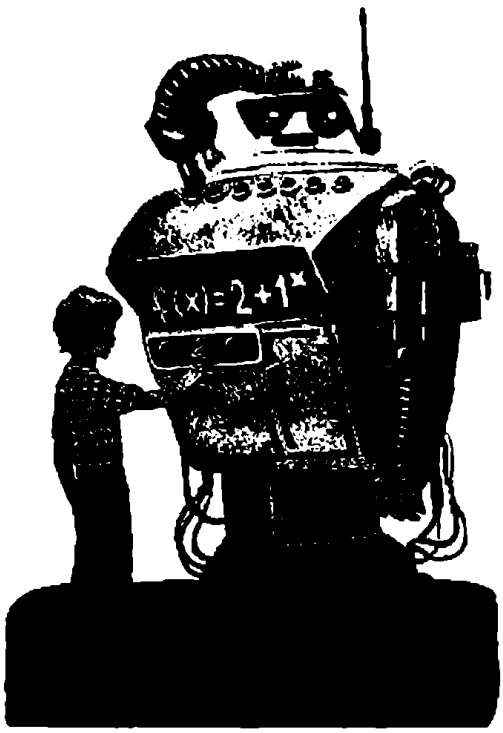
Pre-reading

Where from do you learn most: university, parents, modern media, friends or reading books? Give reasons.
How do you imagine the schooling of the distant future?
Have you read any books by Isaac Asimov or seen any films suggested by his books?

Margie was scornful. "School? What's there to write about school? I hate school."

Margie always hated school, but now she hated it more than ever. The mechanical teacher had been giving her test after test in geography and she had been doing worse and worse until her mother had shaken her head sorrowfully and sent for the County Inspector.

He was a round little man with a red face and a whole box of tools with dials and wires. He smiled at Margie and gave her an apple, then took the teacher apart. Margie had hoped he wouldn't know how to put it together again, but he knew how all right, and, after an hour or so, there it was again, large and black and ugly, with a big screen on which all the lessons were shown and the questions were asked. That wasn't so bad. The part Margie hated most was the slot where she had to put homework and test papers. She always had to write them out in a punch code they made her learn when she was six years old, and the mechanical teacher calculated the mark in no time.



The Inspector had smiled after he was finished and patted Margie's head. He said to her mother, "It's not the little girl's fault, Mrs. Jones. I think the geography sector was geared a little too quick. Those things happen sometimes. I've slowed it up to an average ten-year level. Actually, the over-all pattern of her progress is quite satisfactory." And he patted Margie's head again.

Margie was disappointed. She had been hoping they would take the teacher away altogether. They had once taken Tommy's teacher away for nearly a month because the history sector had blanked out completely.

B. Reading 1

The Fun They Had

by Isaac Asimov

Isaac Asimov (1920 – 1992) was born in Russia and grew up in the USA. His fantastic career as a science fiction writer began in 1939 with the appearance of a short story "Marooned Off Vesta" in "Amazing Stories". In his five decades as an author, he wrote more than four hundred books, won every award his readers and colleagues could contrive to give him, and provided pleasure and insight to millions.

"The Fun They Had" was written by him in 1951. It depicts the world in the year 2157 when each child has his/her own machine teacher.

Margie even wrote about it that night in her diary. On the page headed May 17, 2157, she wrote, "Today, Tommy found a real book!"

It was a very old book. Margie's grandfather once said that when he was a little boy his grandfather told him that there was a time when all stories were printed on paper.

They turned the pages, which were yellow and crinkly, and it was awfully funny to read words that stood still instead of moving the way they were supposed to – on a screen, you know. And then, when they turned back to the page before, it had the same words on it that it had had when they read it the first time.

"Gee," said Tommy, "what a waste. When you're through with the book, you just throw it away, I guess. Our television screen must have had a million books on it and it's good for plenty more. I wouldn't throw it away."

"Same with mine," said Margie. She was eleven and hadn't seen as many telebooks as Tommy had. He was thirteen. She said, "Where did you find it?"

"In my house." He pointed without looking, because he was busy reading. "In the attic." "What's it about?" "School."

So she said to Tommy, "Why would anyone write about school?"

Tommy looked at her with very superior eyes. "Because it's not our kind of school, stupid. This is the old kind of school that they had hundreds and hundreds of years ago." He added loftily, pronouncing the word carefully, "*Centuries* ago."

Margie was hurt. "Well, I don't know what kind of school they had all that time ago." She read the book over his shoulder for a while, then said, "Anyway, they had a teacher."

"Sure they had a teacher, but it wasn't a regular teacher. It was a man." "A man? How could a man be a teacher?" "Well, he just told the boys and girls things and gave them homework and asked them questions." "A man isn't smart enough." "Sure he is. My father knows as much as my teacher." "He can't. A man can't know as much as a teacher." "He knows almost as much, I betcha."

Margie wasn't prepared to dispute that. She said, "1 wouldn't want a strange man in my house to teach me."

Tommy screamed with laughter. "You don't know much, Margie. The teachers didn't live in the house. They had a special building and all the kids went there." "And all the kids learned the same thing?" "Sure, if they were the same age."

"But my mother says a teacher has to be adjusted to fit the mind of each boy and girl it teaches and that each kid has to be taught differently."

"Just the same they didn't do it that way then. If you don't like it, you don't have to read the book."

"I didn't say I didn't like it," Margie said quickly. She wanted to read about those funny schools.

They weren't even half-finished when Margie's mother called, "Margie! School!" Margie looked up. "Not yet, Mamma."

"Now!" said Mrs. Jones. "And it's probably time for Tommy, too."

Margie said to Tommy, "Can I read the book some more with you after school?"

"Maybe," he said nonchalantly. He walked away whistling, the dusty old book tucked beneath his arm.

Margie went into the schoolroom. It was right next to her bedroom, and the mechanical teacher was on and waiting for her. It was always on at the same time every day except Saturday and Sunday, because her mother said little girls learned better if they learned at regular hours.

The screen was lit up, and it said: "Today's arithmetic lesson is on the addition of proper fractions. Please insert yesterday's homework in the proper slot."

Margie did so with a sigh. She was thinking about the old schools they had when her grandfather's grandfather was a little boy. All the kids from the whole neighborhood came, laughing and shouting in the schoolyard, sitting together in the schoolroom, going home together at the end of the day. They learned the same things, so they could help one another on the homework and talk about it. And the teachers were people...

The mechanical teacher was flashing on the screen: "When we add the fractions 1/2 and 1/4..."

Margie was thinking about how the kids must have loved it in the old days. She was thinking about the fun they had.

C. Review Questions

Exercise 1. True/False

1. Tommy and Margie live in the 22nd century.
2. Tommy asked Margie to throw the old book away, when she was through with it.
3. Mechanical teachers never break down.
4. Margie's mechanical teacher had no time to calculate the mark for her homework and test papers.
5. In the Inspector's opinion, Margie's knowledge of geography deserves satisfactory mark.
6. Margie does not study at the weekends.
7. Margie's arithmetic lesson that day was on the addition of decimal fractions.
8. Margie enjoyed the old kind of school.

Exercise 2. Open Ended

1. What did Margie and Tommy find unusual about the old book?
2. What is school like in the year 2157 and how far is it different from today's schools?
3. Would you like to have your personal mechanical teacher? Why?
4. What part of her mechanical teacher does Margie dislike most of all?
5. Do you think homework should be made optional or even cancelled at all? Give reasons.
6. Do you think that CD-ROMs and the Internet will one day lead to the type of school described in the story?
7. *Computer-Aided Language Learning* (*CALL*) refers to any software which is designed and used for languages-related purposes, including electronic dictionaries, translation tools, online language courses, language learning materials available on CD-ROMs, etc. Which of these tools have you already used for the purposes of foreign language learning? Which one(s) would you like to use and why?
8. Read the advertisement below. Would you sign up for the program it extols? Why?

THE FASTEST WAY TO LEARN A LANGUAGE.
GUARANTEED

Imagine learning a new language.
Now imagine it being easy!

Finally! A breakthrough language learning program that delivers fast, effective instruction that rewards effort with success. Rosetta Stone offers CD-ROM and Online solutions for every use. Individuals at home, at work or on the road. Teachers. Trainers. Institutions with global networks. Millions of users in over 100 countries simply say, "It works!"

❑ *Wow! I'm amazed at the ease with which I am learning German! The pictures, text, and pronunciation combine to make this the easiest language-learning experience I've had with any course!*
Steve, Salt Lake City, UT

❑ *I've scoured the earth for the best way to learn Arabic and this is it... a terrific breakthrough product!*
Dr. Bob Arnot, Chief Foreign Correspondent

Exercise 3. *Language Work:* ***Comparison of Adjectives***

Fill in the following table using adjectives from the advertisement (p. 19) and others that have been added for your practice:

Positive	Comparative	Superlative
fast	*faster*	*fastest*
new		
easy		
effective		
good		
terrific		
big		
many		

D. Role Play

Student A You live in the 22nd century. Your homework is to write an essay about the schooling of the past century. The time machine transports you to your coeval who lives in the 21st century. Ask him/her questions about his/her study. Describe your system of education.

Student B You are receiving a visitor from the 22nd century. Answer his/her questions about your study. Set a stress on the advantages of your learning system. Ask him/her questions about his/her study.

E. Pre-reading

- *What is a robot?*
- *What image comes first to your mind at the mention of this word?*
- *What do you know about the latest developments in the field of robotics?*
- *What advantages and disadvantages of robots can you think of?*

F. Reading 2

Laws of Robotics: Implications for Information Technology

by Roger Clarke

Robotics, a branch of engineering, is a popular source of inspiration in science fiction literature; indeed, the term originated in that field. It was first used by the Czech playwright Karel Çapek in 1918 in a short story and again in his 1921 play *R. U. R.*, which stood for *Rossum's Universal Robots*. The term *robot* derives from the Czech word *robota*, meaning forced work or compulsory service, or *robotnik*, meaning *serf*. Rossum, a fictional Englishman, used biological methods to invent and mass-produce "men" to serve humans. Eventually they rebelled, became the dominant race, and wiped out humanity. The play was soon well known in the English-speaking countries.

Many authors have written about robot behaviour and their interaction with humans, but in this company Isaac Asimov stands supreme. To cope with the potential for robots to harm people, Asimov, in 1940 formulated the Laws of Robotics. The writer subjected all of his fictional robots to these laws by having them incorporated within the architecture of their (fictional) "platinum-iridium positronic brains".

Laws of Robotics

- **First Law:** A robot may not injure a human being, or, through inaction, allow a human being to come to harm.
- **Second Law:** A robot must obey orders given it by human beings, except where such orders would conflict with the First Law.
- **Third Law:** A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

The laws quickly attracted – and have since retained – the attention of readers. Asimov's fiction even influenced the origins of robotic engineering. Engelberger, who built the first industrial robot,

called Unimate, in 1958, attributes his long-standing fascination with robots to his reading of Asimov's "I, Robot" when he was a teenager.

Undeterred by its somewhat chilling origins (or perhaps ignorant of them), technologists of the 1950s appropriated the term *robot* to refer to machines controlled by programs. A robot is a reprogrammable multifunctional device designed to manipulate and/or transport material through variable programmed motions for the performance of a variety of tasks. The term *robotics*, which Asimov claims he coined in 1942 refers to a science or art involving both artificial intelligence (to reason) and mechanical engineering (to perform physical acts suggested by reason).

Robotics offers benefits such as high reliability, accuracy, and speed of operation. Low long-term costs of computerized machines may result in significantly higher productivity, particularly in work involving variability within a general pattern. Humans can be relieved of mundane work and exposure to dangerous workplaces. Their capabilities can be extended into hostile environments involving high pressure (deep water), low pressure (space), high temperatures (furnaces), low temperatures (ice caps and cryogenics), and high-radiation areas (near nuclear materials or occurring naturally in space).

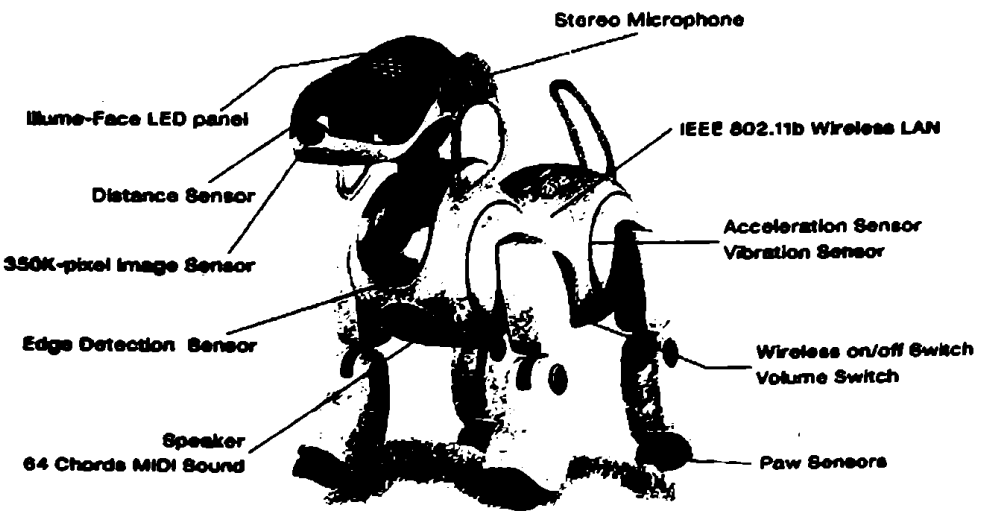


FIGURE 2.1 *The world's first entertainment robot* ***AIBO*** *resembles a dog*

On the other hand, deleterious consequences are possible. Robots might directly or indirectly harm humans or their property; or the damage may be economic or immaterial (for example, to a person's reputation). The harm could be accidental or result from human instructions. Indirect harm may occur to workers, since the application of robots generally results in job redefinition and sometimes in outright job displacement. Moreover, the replacement of humans by machines may undermine the self-respect of those affected, and perhaps of people generally.

G. Review Questions

Exercise 1. Open-Ended

1. How did the word "robot" appear?
2. When was the first industrial robot built?
3. What makes the difference between a robot and a simple computer?
4. Who coined the term "robotics"? What does it mean?
5. What is artificial intelligence?
6. What can you discover by using the Turing test?
7. What jobs traditionally done by people can be equally done by robots today?

H. Discussion Questions

1. Can artificial intelligence be ever created? Give reasons.
2. How necessary is it to design machines with the human form?
3. How do you imagine the future of the mankind when mass production of humanoid robots and androids becomes possible? Do you think they could be dangerous for people?
4. What are the implications of science fiction for real roboticists and information technologists?
5. Would you wish your home to be run as described in the newspaper extract given on the following page?

THE ROBOT AGE has begun – thanks to the silicon chip which can do the work of a massive computer bank. Already the cheap brain-power of these quarter inch chips – called micro-processors – has put a calculator in almost everyone's pocket and created a £25 million industry for computer games.

IMAGINE your home being run by an electronic Micro Mother. A push-button brain that organizes the shopping and the cooking, pays the bills, and even remembers your birthday!

And imagine all the household chores being handled by a robot Mrs Mop that washes the floor, cleans the carpet and even mows the lawn!

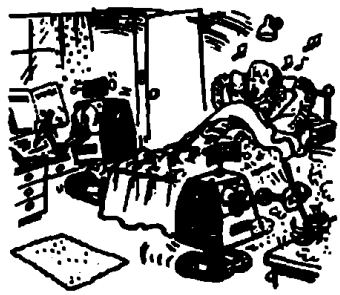
No, it's not just a futuristic dream – tomorrow's world is already here...

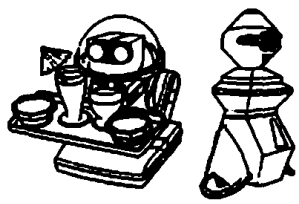
Micro Mum and Mrs Mop are the forerunners of the first generation of computerized home robots created by the silicon chip.

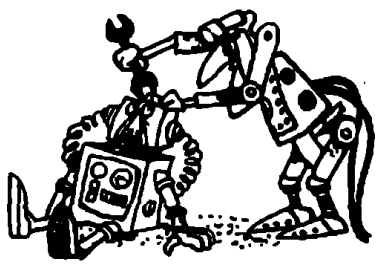
Just press a button and Micro Mum will wake you up,
make the tea, read the news,
pay the bills and
cook the bacon
just the way you like it...

I. Writing

Try yourself as a science fiction writer and make up a story or a dialogue about one of the pictures below:







MATHEMATICAL GENIUSES

Vocabulary List

abstract set
algebra, *n*
algorithm, *n*
analogy, *n*
Analysis, *n*
artificial intelligence
Banach space
bell curve
Cartesian geometry
computer, *n*
computer science
functional analysis
Gauss-Jordan elimination
Gaussian error curve
general topology
genius, *n*
geometry, *n*
group theory
hypotenuse, *n*
integral equation
I.Q., *abbr*
least squares method
Lebesgue measure
linear operation
linear space
mathematician, *n*
mathematics, *n*
matrix equation
non-Euclidean geometry
partial differential equation
problem, *n*
proof, *n*
right angled triangle
set theory
square, *n*
sum, *n*, *v*
theorem, *n*
theory of measure and integration
thesis, *n*
vector space

A. Pre-reading

- *What names of world-famous mathematicians do you know?*
- *What have you already learned about Stefan Banach?*
- *Read the quote by Banach given on the following page. What type of mathematician – according to Banach's typology – do you think you are now? What type do you expect to become in your fifth year at the university?*

B. Reading

Stefan Banach

by J. J. O'Connor and E. F. Robertson

A mathematician is a person who can find analogies between theorems; a better mathematician is one who can see analogies between proofs, and the best mathematician can notice analogies between theories. One can imagine that the ultimate mathematician is one who can see analogies between analogies.

S. Banach

Banach was born on 30 March 1892 in Kraków, Austria-Hungary (now Poland). His father Stefan Greczek worked as a tax official. He was not married to Banach's mother Katarzyna Banach who vanished after Stefan was baptised, when he was only four days old, and nothing more is known of her.

Banach attended primary school in Kraków, then the Henryk Sienkiewicz Gymnasium No 4. During his first few years at the Gymnasium the boy achieved first class grades with mathematics and natural sciences being his best subjects. However, the excellent grades of his early years gave way to poorer grades as he approached his final school examination. A

On leaving school Banach wanted to study mathematics, but felt that nothing new could be discovered in this field so he chose to study engineering. Banach's father had never given his son much support, but now once he left school he quite openly told Stefan that he was now on his own. B

Stefan Banach was largely self-taught in mathematics; his genius was accidentally discovered by Hugo Shteinhaus. Waiting to take up a post at The Jan Kazimierz University in Lviv, Shteinhaus would walk through the streets of Kraków in the evenings and during

one such walk he overheard the words "Lebesgue measure". Shteinhaus approached the park bench and introduced himself to the two young apprentices of mathematics. The youngsters were Stefan Banach and Otto Nikodym. From then on they would meet on a regular basis.

Once Shteinhaus told Stefan of a problem which he was working on without success. After a few days Banach had the main idea for the required counterexample and they both wrote a paper, Banach's first, which appeared in the *Bulletin of the Kraków Academy* in 1918. Since then Banach started to write important mathematics papers very rapidly. C

In 1920 Banach became a lecturer in mathematics at Lviv Polytechnic School and submitted a dissertation for his doctorate under Lomnits'kyi's supervision. This was, of course, not the standard route to a doctorate, for Banach had no university mathematics qualifications. However, an exception was made to allow him to submit the thesis *On Operations on Abstract Sets and their Application to Integral Equations* (1922). D

In 1924 Banach was promoted to full professor of The Kazimierz University in Lviv (now renamed The Ivan Franko University). The mathematical life was very intense in Lviv at that time. Some of the mathematicians met practically every day, informally in small groups, at all times of the day to discuss problems of common interest, communicating to each other the latest work and results. Apart from the more official meetings of the local sections of the Mathematical Society (which took place Saturday evenings, almost every week!), there were frequent informal discussions mostly held in "The Scottish Coffee House" (the original Szkocka Café in the former Academichna Str.). Problems were written in a large notebook kept by the headwaiter, who, upon demand,

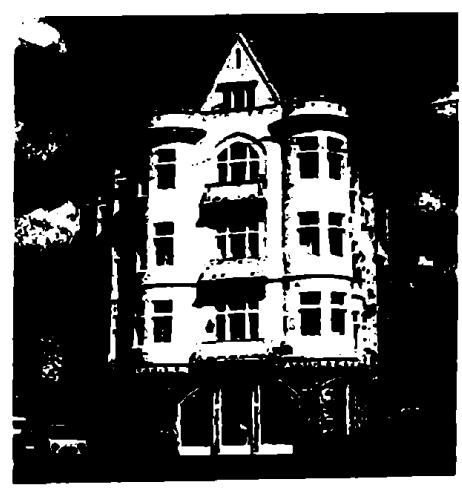


FIGURE 3.1 *Szkocka Café*

would bring it out of some, return it to its secret secure hiding place,and after the guests departed location. Often prizes were offered for the best solutions. E

Stefan Banach also spent most of his days in cafés, not only in the company of others but by himself as well. He liked the noise and the music. They did not prevent him from concentrating and thinking. There were cases when, after the cafés closed for the night, he would walk over to the railway station where the cafeteria was open around the clock. F

In 1939, just before the start of World War II, Banach was elected President of the Polish Mathematical Society. At the beginning of the war Soviet troops occupied Lviv. Banach had been on good terms with the Soviet mathematicians before the war started, visiting Moscow several times, and was treated well by the new Soviet administration. He was allowed to continue to hold his chair at Lviv State University and became the Dean of the Faculty of Physics and Mathematics. Life at this stage was little changed for Banach who continued his research, his textbook writing, lecturing and sessions in the cafés. He was in Kyiv when Germany invaded the Soviet Union and he returned immediately to his family in Lviv.

The Nazi occupation of Lviv in June 1941 meant that Banach lived under very difficult conditions. He was arrested under suspicion of trafficking in German currency but released after a few weeks. He survived a period when Polish academics were murdered, his doctoral supervisor Lomnits'kyi dying on the tragic night of 3 July 1941. Banach survived, but the only way he could work for a living was by feeding lice with his blood in professor Rudolf Weigl Institute where typhoid fever research was conducted. G

Stefan Banach laid the foundations of modern functional analysis, which he continued to work at throughout his life. He also made fundamental contributions to general topology, set theory, the theory of measure and integration, and the general theory of linear spaces, or vector spaces, e.g., *The Theory of the Linear Operations* (1931, 1932, 1948). He introduced and developed the concept of complete normed linear spaces, now called Banach spaces.

C. Review Questions

Exercise 1. Fill in the Blank

Seven sentences have been removed from the text. Choose from the sentences below the one which fits each gap (A-G). There is an extra sentence which you do not need to use.

1. This thesis is sometimes said to mark the birth of functional analysis.
2. Banach was not physically fit for army service, having poor vision in his left eye.
3. There, over a glass of beer, he would think about his problems.
4. Stefan passed this examination in 1910 but he failed to achieve a pass with distinction, an honor which went to about one quarter of the students.
5. A collection of these problems appeared later as the *Scottish Book.*
6. Banach planned to go to Kraków after the war to take up the chair of mathematics at The Jagiellonian University but his health undercut during the occupation and he died in Lviv in 1945 of lung cancer.
7. It was also through Shteinhaus that Banach met his future wife Lucja Braus.
8. Banach went to Lviv where he enrolled in the Faculty of Engineering at Lviv Polytechnic School (now Lviv Polytechnic National University).

Exercise 2. Open Ended

1. When and where was S. Banach born?
2. Did Stefan's parents pay much attention to his upbringing and education? Give reasons.
3. What kind of pupil was he?
4. When did he become interested in mathematics?
5. Why did Banach decide to study engineering when he left school?
6. How did Banach, a young apprentice of mathematics, excel professor Shteinhaus?

7. When and where did Banach publish his first mathematics paper?
8. What was the title of his dissertation?
9. What was remarkable and unusual about Banach's way to doctorate?
10. When did he take up the post of full professor of The Kazimierz University in Lviv?
11. What working conditions promoted Banach's scientific thinking?
12. Where did the Scottish Book get its name from?
13. When was Banach elected President of the Polish Mathematical Society?
14. What faculty of Lviv State University was Banach the dean of?
15. How did he manage to survive during the Nazi occupation of Lviv?
16. What branch of mathematics is Banach considered the founder of?
17. What is the most famous work of Banach?
18. What mathematical concept is named after him?

*Exercise 3. Matching: **Mathematical Geniuses***

1. Pythagoras of Samos
2. Euclid of Alexandria
3. Muhammed ibn Musa al-Khwarizmi
4. René Descartes
5. Carl Friedrich Gauss
6. Baron Augustin-Louis Cauchi
7. Nikolay Lobachevsky
8. Charles Babbage
9. Ada King, Countess of Lovelace
10. Sofia Kovalevskaya
11. John Venn
12. Alan Mathison Turing

a) British mathematician who designed a type of calculating machine which modern computers are based on.

b) the most prominent mathematician of antiquity best known for his treatise on mathematics *The Elements*.

c) the first pure mathematician in the history of mathematics. Today we particularly remember him for his famous geometry theorem – for a right angled triangle the square on the hypotenuse is equal to the sum of the squares on the other two sides.

d) English mathematician and logician, wartime codebreaker, the founder of research in artificial intelligence, the father of computer science.

e) Russian woman mathematician, the author of the work *Partial differential equations*.

f) well-known Arabic mathematician of the 9th century, the author of *The Compendious Book on Calculation by Completion and Balancing* (*Al-Kitāb al-mukhtasar fī hisāb al-jabr wa'l-muqābala*). Latin corruption of the book's title resulted in the word *algebra* (*al-jabr*); a corruption of the author's own name resulted in the term *algorithm*.

g) English logician best known for his diagrammatic way of representing sets, their unions, and intersections.

h) German mathematician, who developed the fundamental theorem of algebra, the least squares method, Gauss-Jordan elimination (for solving matrix equations), and the bell curve, or Gaussian error curve.

i) Russian mathematician who is considered the founder of non-Euclidean geometry.

j) French mathematician, pioneer of Analysis and group theory.

k) English mathematician called the first computer programmer. The programming language *Ada* is named for her.

l) French mathematician and philosopher whose work *La géométrie* includes his application of algebra to geometry from which we now have Cartesian geometry.

Did You Know?

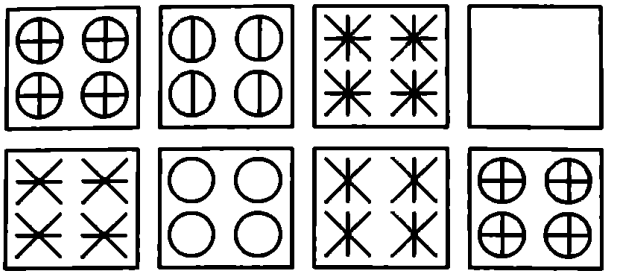
SIX Nobel Prizes are awarded each year, one in each of the following categories: literature, physics, chemistry, peace, economics, and physiology and medicine. Notably absent from this list is an award for Mathematics.

One of the most common - and unfounded - reasons as to why Nobel decided against a Nobel prize in math is that his mistress cheated him with a famous mathematician Gosta Mittag-Leffler.

Exercise 4. Quiz: ARE YOU A GENIUS?

*There is an international organization, **Mensa**, whose only requirement for membership is an I.Q. in the "genius" range. Try this test and see if you are eligible.*

1. Which of the lower boxes best completes the series on the top?



(a) (b) (c) (d)

2. I am a man. If Larry's son is my son's father, what relationship am I to Larry?
   a) his grandfather c) his son e) I am Larry
   b) his father d) his grandson f) his uncle
3. Which word does not belong in the following group?
   a) knife b) swan c) smile d) feather e) lovely f) thought
4. Which two shapes below represent mirror images of the same shape?

(a) (b) (c) (d) (e)

5. What number comes next in the series?
   9, 16, 25, 36... .
6. Complete this analogy with a five-letter word ending with letter "H". *High is to low as sky is to* _ _ _ _ H.
7. In the box below, a rule of arithmetic applies across and down the box so that two of the numbers in a line produce the third. What is the missing number?

6	2	4
2	?	0
4	0	4

8. Complete this analogy with a seven-letter word ending with the letter "T". *Potential is to actual as future is to* _____T.
9. In the group below, find the two words whose meanings do not belong with the others.
   a) glue b) sieve c) buzz saw d) nail e) string f) paper clip
10. Mountain is to land as whirlpool is to:
    a) forest b) wet c) sea d) sky e) shower
11. Find the number that logically completes the series.
    2, 3, 5, 9, 17... .
12. Two of the shapes below represent mirror images of the same shape. Which are they?

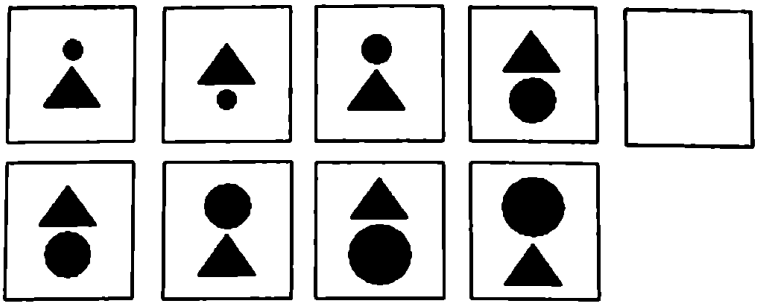
(a) (b) (c) (d)

13. Statistics indicate that men drivers are involved in more accidents than women drivers. The only conclusion that can certainly be drawn is that:
    a) male chauvinists are wrong, as usual, about women's abilities
    b) men are actually better drivers but drive more frequently
    c) men and women drive equally well, but men log more total mileage
    d) most truck drivers are men
    e) there is not enough information to justify a conclusion
14. In the box below, a rule of arithmetic applies across and down the box so that two of the numbers in a line produce the third. What is the missing number?

6	2	12
4	5	20
24	10	?

15. If A×B=24, C×D=32, B×D=48 and B×C=24 what does A×B×C×D equal?
    a) 480 b) 576 c) 744 d) 768 e) 824

16. Which of the four lower selections best completes the series on the top?



(a) (b) (c) (d)

17. Which word does not belong in this group?
    a) microscope c) microphone e) telegraph
    b) magnifying glass d) telescope

18. Find the two words nearest in the meaning to each other.
    a) beam b) lump c) giggle d) ray e) collection

19. If Jim turns right or left at the stop sign he will run out of gas before he reaches a service station. He has already gone too far past a service station to return before he runs out of gas. He does not see a service station ahead. Only one of the following statements can be positively deduced:
    a) he may run out of gas
    b) he will run out of gas
    c) he shouldn't have taken this route
    d) he is lost
    e) he should turn right at the stop sign
    f) he should turn left at the stop sign

20. Complete the following analogy

as + – 0 are to:

a) + – 0 b) 0 + – c) – + 0 d) 0 – + e) + + 0

How to score

Give yourself one point for each correct answer. If you scored:

20-25 points: You are extremely intelligent – a perfect candidate for *Mensa*.

15-19 points: This should put you in the higher percentiles of the population – definitely a *Mensa* candidate.

10-14 points: Nothing to be ashamed of – a most respectable score. You should probably try the complete, standard *Mensa* test.

Fewer than 10 points: Forget about joining *Mensa*, but don't stew about it. You may just be having a bad day. Some of the most successful writers, businessmen, artists and other famous people don't have exceptionally high I.Q.s either.

D. Discussion Questions

1. The nature versus nurture debate: do we inherit our I.Q. or is it the result of our environment, studying, and upbringing?
2. Have I.Q. and intelligence got anything to do with creativity and genius?
3. Does I.Q. vary according to sex?
4. Does being intelligent matter or are other things more important?
5. Are puzzles, intelligence tests and so on a waste of time?
6. What do you know about the left and right brain hemispheres?

Check Your Progress (1)
Units 1 – 3

Fill in the blanks in the sentences below using information from units 1 – 3.

1. By the middle of the 18th century the course of Mathematics included the study of _____ and elements of _____.
2. In _____ the Department of Mathematics was founded in Lviv University within the Faculty of Philosophy.
3. They laid the foundations of Lviv School of Mathematics headed by S. Banach, the author of the work "The Theory of the _____ _____"
4. Since 1956 the group of students under the supervision of professor O. Kostovs'kyi started teaching Computational Mathematics and _____.
5. The Faculty of Applied Mathematics and Informatics was established in Lviv State University in _____.
6. The part Margie hated most was the slot where she had to put homework and _____ _____.
7. The mechanical teacher was flashing on the _____: "When we add the _____ 1/2 and 1/4..."
8. To cope with the potential for robots to harm people, Asimov, in 1940 formulated the _____of _____.
9. Engelberger built the first industrial robot, called Unimate, in _____.
10. On leaving school Banach wanted to study mathematics, but felt that nothing new could be discovered in this field so he chose to study _____.
11. Stefan Banach laid the foundations of modern _____ _____.
12. S. Banach introduced and developed the concept of complete normed linear spaces, now called _____ _____.

SPOKEN NUMBERS AND MEASUREMENTS (1)

1. *The sentences below are written as they would be spoken. Rewrite them as they would normally be written using numbers and abbreviated forms.*

E.g. I take a size fourteen and a half shirt.
I take a size 14½ shirt.

a) Forty minus fifteen plus six is thirty-one.
b) One hundred and fifty divided by ten is fifteen.
c) Three multiplied by six is eighteen.
d) Two to the power of four equals sixteen.
e) Seven squared equals forty-nine.
f) *a* second is greater than *a* third.
g) Bracket *a* minus five *b* close the bracket.
h) Phone me any time on four double one four eight five oh.
i) "Which bus goes to the High street?" – "One-five-two."
j) The unemployment rate is five point one seven percent.
k) It cost me six euros seventy-five.

2. *Write out the following sentences exactly as they would be spoken, i.e. as in the exercise above.*

a) $73 + 20 - 43 = 50$.
b) $129 \div 3 = 43$.
c) $4 \times 21 = 84$.
d) $3^k = 9$.
e) $4^2 = 16$.
f) $b_1 > b_2$.
g) $(2x - y)$.
h) Our new phone number is 307 2201.
i) "What is your house number?" – "238".
j) It is 17.38% gold.
k) The chocolate is priced at €3.25.

MATHEMATICAL INDUCTION

Vocabulary List

add, *v*	illustration, *n*	set, *v*
arbitrary, *adj*	mathematical induction	statement, *n*
assume, *v*	obtain, *v*	sum, *n, v*
data, *n pl*	polynomial, *n, adj*	term, *n*
denote, *v*	positive integer	theorem, *n*
divide, *v*	proof, *n*	therefore, *adv*
equation, *n*	proposition, *n*	thus, *adv*
express, *v*	prove, *v*	true, *adj*
expression, *n*	reasoning, *n*	valid, *adj*
factor, *n, v*	series, *n*	value, *n*

A. Pre-reading

– *How can you benefit from the knowledge of the English language of Mathematics and Computer Science in your studies and professional career?*

B. Reading

Mathematical Induction

by Max Kurtz

Where a proposition is believed to be true but a direct proof is difficult to secure, an indirect proof is sometimes obtainable by a process of reasoning called *mathematical induction.* To illustrate the process, we shall prove the following:

$$1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \cdots + n(n+1) = \frac{n(n+1)(n+2)}{3} \qquad (d)$$

Let S_n denote the sum of the terms at the left, and set $n = 1$. The left and right sides of Eq. (*d*) yield the following values, respectively:

$$S_1 = 1 \cdot 2 = 2 \qquad S_1 = \frac{1 \cdot 2 \cdot 3}{3} = 2$$

Thus, Eq. (*d*) is valid for $n = 1$. Now assume it is valid for $n = k$, where k is an arbitrary positive integer. Adding the $(k + 1)$th term in the series and expressing it as $3(k + 1)(k + 2)/3$, we obtain

$$S_{k+1} = \frac{k(k+1)(k+2)}{3} + \frac{3(k+1)(k+2)}{3} = \frac{(k+1)(k+2)}{3}(k+3) =$$
$$= \frac{(k+1)(k+2)(k+3)}{3}$$

Comparing the expression for S_{k+1} with the expression in Eq. (*d*), we arrive at this conclusion: If Eq. (*d*) is valid for $n = k$, it is also valid for $n = k + 1$. We have already demonstrated that Eq. (*d*) is valid for $n = 1$. Therefore, it is valid for $n = 2$. Since it is valid for $n = 2$, it is also valid for $n = 3$, etc. It follows that Eq. (*d*) is valid for all positive integral values of n.

As a second illustration, we shall demonstrate that $x - y$ is a factor of the expression $x^n - y^n$ for all positive integral values of n. For this purpose, we write

$$x^{k+1} - y^{k+1} = x(x^k - y^k) + y^k(x - y)$$

Therefore, if the proposition is true for $n = k$, it is also true for $n = k + 1$. The proposition is true for $n = 1$, and the proof is now complete.

C. Review Questions

Exercise 1. Text-based Translation

Якщо твердження вважають істинним, але пряме доведення є складним; процес міркувань; позначимо через S_n суму членів;

покладемо n рівним 1; ліва і права частини рівності (d) задають такі значення; відповідно; довільне додатне ціле число; рівність (d) виконується при $n = k$; тощо; звідси випливає; дільник виразу $x^n - y^n$; що і треба було довести.

Exercise 2. Anagrams

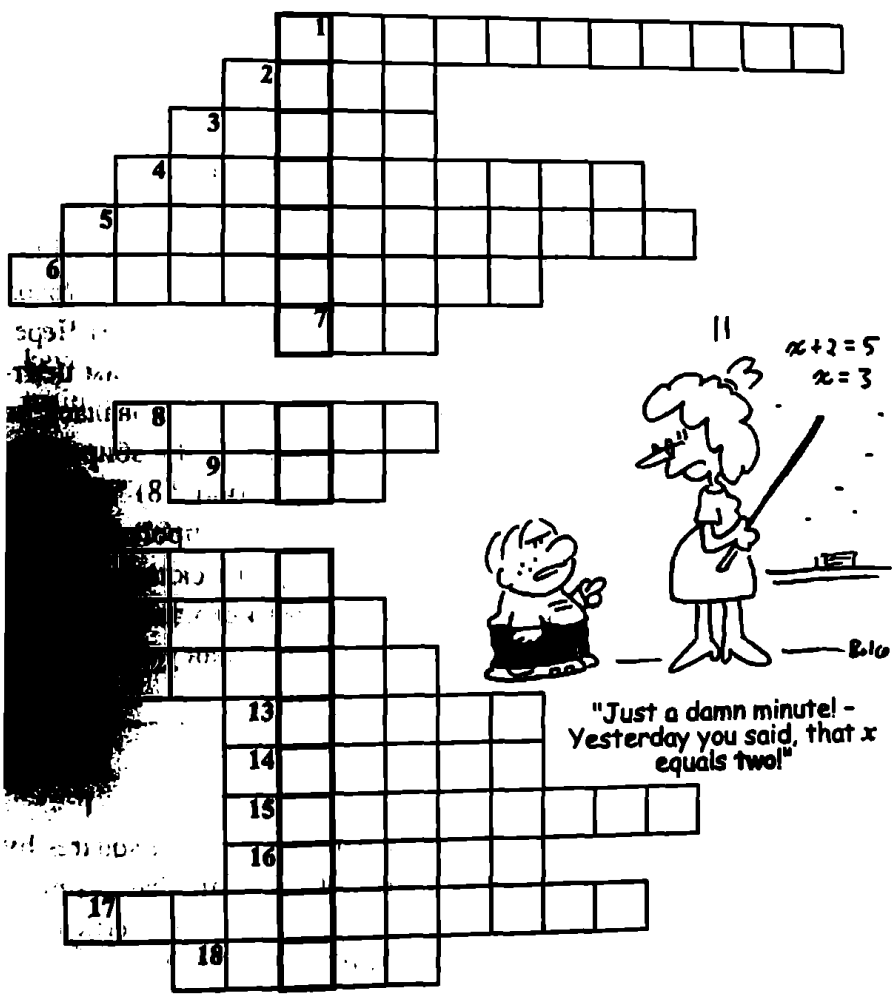
Solve the anagrams by reading the clues and putting the letters of the words in order. Enter the solutions in the table on page 41 to find the ***mystery clue.***

1. Something that must be proved ..... POSPOORINIT
2. Based on facts and not imagined or invented..... UTRE
3. Reasons that show a theorem (=statement) to be true..... FPORO
4. Something you decide after considering some data..... NCSCUOIONL
5. Connected with or using mathematics ..... EIMCMAALTHTA
6. Any mathematical form expressed symbolically.. ISEESOXNPR
7. The result of an addition..... UMS
8. To be a sign of something..... ENTOED
9. Opposite of "right"..... FETL
10. A number or polynomial that divides a given number or polynomial exactly..... RATFCO
11. A whole number..... GIRTNEE
12. To get..... ATONBI
13. To think that something is true, although you have no proof of it ..... MSUASE
14. To come directly after an event or as a result of it... WLOFOL
15. A method of proving that each of an infinite sequence of mathematical statements is true if the first statement is proved to be true..... ITNONIDCU
16. Straightforward..... RCETDI
17. To show or prove something clearly..... SDERENAMOTT
18. Opposite of "left"..... GIRTH

☺ ☺ ☺

"Do you love your math more than me?"
"Of course not, dear - I love you much more."
"Then prove it!"
"OK... Let R be the set of all lovable objects..."

☺ ☺ ☺



Exercise 3. Language Work: Continuous Tenses

	Past	was / were	V-ing	*I was writing* *We were testing*
	Present	am / is / are		*I am writing* *He is testing*
	Future	shall / will be		*I shall be writing* *He will be testing*
	Future in the Past	should / would be		*I should be writing* *He would be testing*

Translate the following into English using Continuous tenses.

1) Зараз студенти пишуть тест з лінійної алгебри. 2) Ти будеш користуватись комп'ютером сьогодні увечері? 3) (*Мама синові*) Замість того, щоб готуватись до (to study for) іспитів, ти постійно сидиш в (to surf) Інтернеті! 4). Вони сказали, що упродовж трьох наступних днів будуть аналізувати ці статистичні дані. 5) Через вірус ми тимчасово не користуємось нашим комп'ютерним центром. 6) Інформація про їхню знахідку швидко поширювалась по (to spread over) глобальній комп'ютерній мережі. 7) Ти збираєшся брати участь в олімпіаді з (olympiad in) інформатики? 8) "Над чим зараз працює ваша команда?"– "Ми розробляємо програму вивчення мови за допомогою комп'ютера". 9) Студент складає іспит з (to take exam in / on) програмування. Викладач його запитує: "Що таке *поліморфізм*?" 10) Система зависла, коли я грав у цю гру.

C. Problem Solving

Suppose that you begin with a chocolate bar made up of n squares by k squares. At each step, you choose a piece of chocolate that has more than two squares and snap it in two along any line, vertical or horizontal. Eventually, it will be reduced to single squares. Show by induction that the number of snaps required to reduce it to single squares is $nk - 1$.

BUYING A COMPUTER

Vocabulary List

CD-ROM, *abbr*
central processing unit (CPU)
configuration, *n*
data, *n pl*
data storage
display, *n, v*
DVD, *abbr*
graphic(s) processor
hard disk
hard drive
hardware, *n*
input/output device
keyboard, *n*
main memory
microprocessor chip
monitor, *n*
motherboard, *n*
mouse, *n*
peripherals, *n pl*
process, *v*
program, *n, v*
Random Access Memory (RAM)
Read Only Memory (ROM)
software, *n*
speaker, *n*
storage device
system unit
three-dimensional, *adj*

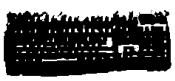
A. Pre-reading

– *Match the pictures with the names:*


1


2


3


4


5


6

___ monitor
___ keyboard
___ CD-ROM
___ motherboard
___ system unit
___ hard drive

B. Reading

What Is a Computer System?

by Santiago Remacha Esteras

A computer system consists of two parts: the software and the hardware. The software is the information in the form of data and program instructions. The hardware components are the electronic and mechanical parts of the system. The basic structure of a computer system is made up of three main hardware sections: the Central Processing Unit (CPU), the main memory, and the peripherals.

The CPU is a microprocessor chip which executes program instructions and coordinates the activities of all the other components. In a way, it is the "brain" of the computer. Larger computers may have two or more CPUs, in which case they are simply called "processors" because each is no longer a "central" unit. Increasingly, personal computers contain specialized graphic processors, with dedicated memory, for handling the computations needed to display complex graphics, such as for three-dimensional simulations and games.

The main memory holds the instructions and data which are currently being processed by the CPU. The internal memory of a microcomputer is usually composed of two sections: RAM (Random Access Memory) and ROM (Read Only Memory).

The peripherals are the physical units attached to the computer. They include input/output devices as well as storage devices. Input devices enable us to present information to the computer; for example, the keyboard and the mouse. Output devices allow us to extract the results from the computer; for example, we can see the output on the monitor or in printed form. Auxiliary data storage is usually provided by an internal hard disk and may be supplemented by other media such as CD-ROMs or DVDs.

These are the main physical units of a computer system, generally known as the configuration.

Did You Know?

IN its history, the computer has experienced 5 generations:

I generation – ***1940 -1956***: **Vacuum Tubes**

II generation – ***1956 -1963***: **Transistors**

III generation – ***1964 -1971***: **Integrated Circuits**

IV generation – ***1971 - present***: **Microprocessors**

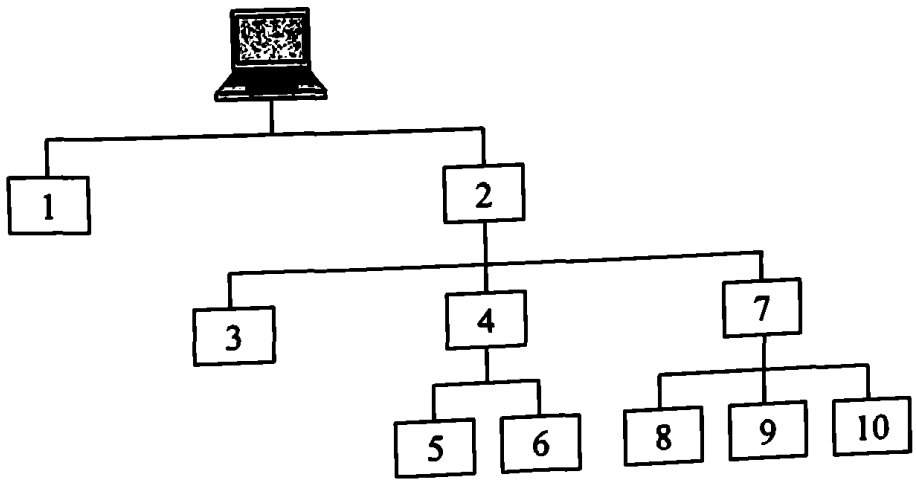
V generation – ***present and beyond***: **Artificial Intelligence**

C. Review Questions

Exercise 1. Matching

Write the numbers from the chart next to the elements of computer system.

output devices	peripherals
software	input devices
main memory	storage devices
ROM	hardware
CPU	RAM



Exercise 2. Open Ended

1. Describe the features of the computer system that you
   a. have today;
   b. would like to have today;
   c. would like to have in another year or two

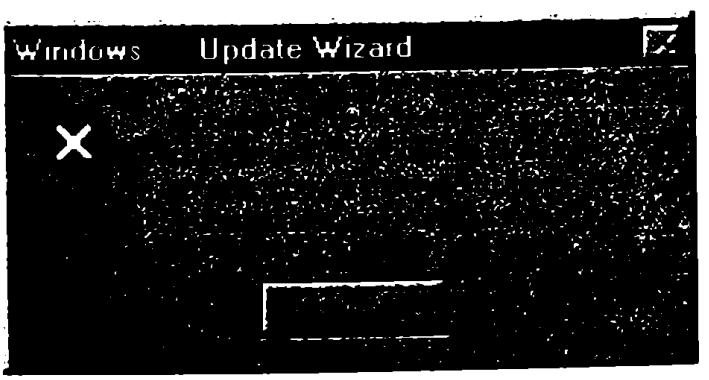
along the following parameters:

- CPU (specify type and speed)_____
- Memory (specify amount)_____
- Hard disk (specify capacity)_____
- Disk drives (specify type, speed, and capacity)_____
- Monitor (specify type and size)_____
- System software_____
- Application software_____
- Other – modem, speakers, web camera, printer (specify)_____

Useful phrases:

It has got...
It's very fast. It runs at...
The hard disk can hold...
As for the disk drive,...
I need 22" monitor, because...
The standard RAM memory is... and it is expandable...
I'd like to add a graphics processor in order to...

2. If you already have your own computer, what piece of its equipment would you like to upgrade first and why?



Exercise 3. Ordering

Put the lines in this telephone conversation in the right order and then choose a title for it. The first line has been marked for you as an example.

Conversation title:_____

___ **B:** Yes, it's Asus P5QC.
___ **B:** That sounds alright. How much is it?
___ **B:** One module – 2 gigabytes.
___ **B:** Um, I'm afraid, I can't afford it for $81. Have you got anything cheaper?
___ **A:** Uh-huh. And how much memory have you got at the moment?
___ **A:** Well, we have 1Gb RAM for only $44.99, but it is not currently in stock.
_1_ **A:** *PU Computer Parts*. Can I help you?
___ **A:** Well, there are different memory modules available. Do you know the model of your motherboard?
___ **A:** $80.99.Would you like to order it now?
___ **A:** I see... Then, perhaps, you'd be interested in this one – DDR3 2Gb PQI 1333.
___ **B:** What a shame... OK, thank you. Good-bye.
___ **B:** Yes. I bought one of your machines a month ago and now it looks as though I need more memory.

Exercise 4. Short Response

*As you know there are three types of memory used by computers: **RAM, ROM, and storage memory**. Look through this list of features and decide which type of memory they refer to.*

1. Any section of the main memory can be read with equal speed and ease.
2. It is available in magnetic, optical and video disks.
3. A certain amount of this memory can be designated as "cache" memory to store information in applications that are used very frequently.

4. It stores basic operating instructions, needed by the CPU to function correctly.
5. Memory which can be expanded by adding SIMMs of 1MB, 2MB, 4MB or other major increments.
6. Information is permanent and cannot be deleted.
7. You can save and store your documents and applications.

D. Role Play

Student A You want to buy a personal computer for up to $1000.

Student B You are a sales assistant in a computer shop. Do your best to sell the most expensive equipment from your pricelist.

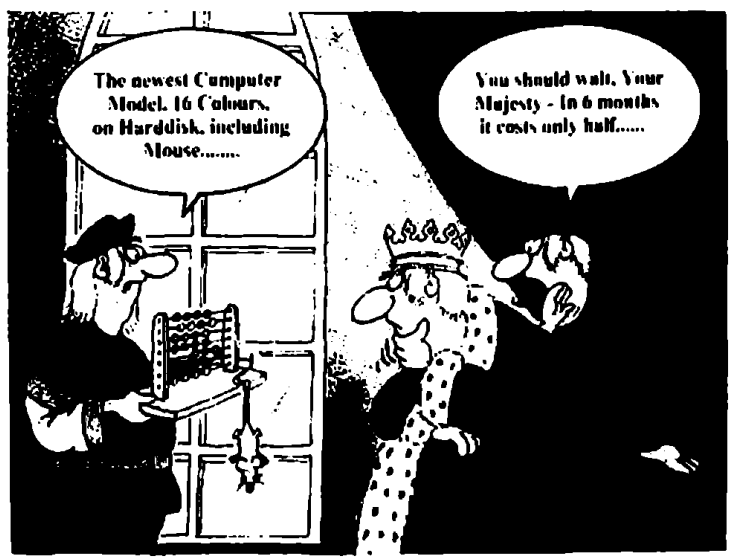
Note: *you can either make your own pricelists (e.g., see the table below) or use the original ones.*

Products available	CPU speed	Min/Max RAM	Hard disk	Disk drives	Monitor	Price
...	...	...	...	...	...	...

Begin your conversation as follows:

Assistant: Hello. Do you need any help?

Buyer: Yes, I'm looking for a personal computer...



GETTING READY TO PROGRAM

Vocabulary List

algorithm, *n*
bug, *n*
C++ compiler
central processing unit (CPU)
code, *n*, *v*
compilation, *n*
compile, *v*
data, *n pl*
debug, *v*
debugger, *n*
debugging, *n*
digital electronic machine
disk drive
edit, *v*
error, *n*
executable file
execute, *v*
file, *n*, *v*
flowchart, *n*
input/output device
keyboard, *n*
linker, *n*
memory, *n*
object code
object file
operating system
preprocessor, *n*
printer, *n*
program, *n*, *v*
programmable, *adj*
programmer, *n*
programming, *n*
run, *n*, *v*
screen, *n*
source code
source file
store, *v*
tape drive
test, *n*, *v*
testable, *adj*
tester, *n*
testing, *n*

A. Pre-reading

- *Describe how you use computers in your study and free time.*
- *What programming language(s) do you already know / are you studying now / will you study at university later?*
- *When do we celebrate Programmer's day?*
- *Remember your first program. Tell your group mates about it.*

B. Reading

Getting Ready to Program

by Ira Pohl

Programs are written to instruct machines to carry out specific tasks or to solve specific problems. A step-by-step procedure that accomplishes a desired task is called an *algorithm*. Thus, *programming* is the activity of communicating algorithms to computers. We have all given instructions to someone in English and then had that person carry out the instructions. The programming process is analogous, except that machines have no tolerance for ambiguity and must have all steps specified in a precise language and in tedious detail.

The Programming Process

1. Specify the task.
2. Discover an algorithm for its solution.
3. Code the algorithm in C++.
4. Test the code.

A computer is a digital electronic machine composed of three main components: processor, memory, and input/output devices. The processor is also called the *central processing unit*, or *CPU*. The processor carries out instructions that are stored in the memory. Along with the instructions, data also is stored in memory. The processor typically is instructed to manipulate the data in some desired fashion. *Input/output devices* take information from agents external to the machine and provide information to those agents. Input devices are typically terminal keyboards, disk drives, and tape drives. Output devices are typically terminal screens, printers, disk drives, and tape drives. The physical makeup of a machine can be quite complicated, but the user need not be concerned with the details.

The *operating system* consists of a collection of special programs and has two main purposes. First, the operating system oversees and coordinates the resources of the machine as a whole. For example, when a file is created on a disk, the operating system takes

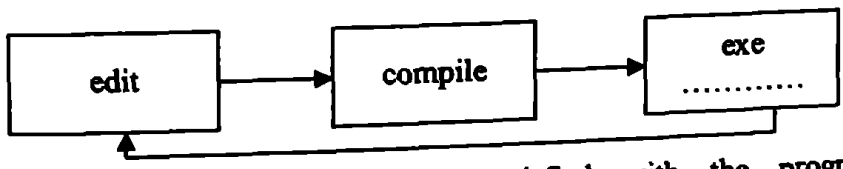
care of the details of locating it in an appropriate place and keeping track of its name, size, and date of creation. Second, the operating system provides tools to users, many of which are useful to the C++ programmer. Two of these tools are of paramount importance: the text editor and the C++ compiler.

We assume the reader can use a text editor to create and modify files containing C++ code. C++ code is also called *source code*, and a file containing source code is called a *source file*. After a file containing source code (a program) has been created, the C++ compiler is invoked. This process is system-dependent. For example, on many UNIX systems, we can invoke the C++ compiler with the command

CC pgm.cpp

where *pgm.cpp* is the name of a file that contains a program. If there are no errors in *pgm.cpp*, this command produces an *executable file* – one that can be run, or executed. Although we think of this as compiling the program, what actually happens is more complicated.

When we compile a simple program, three separate actions occur: first the preprocessor is invoked, then the compiler, and finally the linker. The preprocessor modifies a copy of the source code by including other files and by making other changes. The compiler translates this into *object code*, which the linker then uses to produce the final executable file. A file that contains object code is called an *object file*. Object files, unlike source files, usually are not read by humans. When we speak of compiling a program, we really mean invoking the preprocessor, the compiler, and the linker. For a simple program, this is all done with a single command. After the programmer writes a program, it has to be compiled and tested. If modifications are needed, the source code has to be edited again. Thus, part of the programming process consists of this cycle:



When the programmer is satisfied with the program performance, the cycle ends.

C. Review Questions

Exercise 1. Open Ended

1. What is programming?
2. What are the steps in the development of a program?
3. How much time (in percents) does each step generally take?
4. What step(s) do **you** usually omit and why?
5. What do you call any collection of statements or declarations written in some human-readable computer programming language?
6. Look at the following program and try to guess what it does without running it.

```
1: # include <iostream.h>
2: int main()
3: {
4: int x = 5;
5: int y = 7;
6: cout << "\n";
7: cout << x + y << " " << x * y;
8: cout << "\n";
9: return 0;
10: }
```

7. What is the compiler and what function does it perform?
8. What is meant by a good coding style?
9. What are the qualities of a good program?

Real programmers code in binary

Exercise 2. True/False

1. Coding is writing a program.
2. A program can be defined as a set of written instructions created by a programmer or an executable piece of software.
3. Terminal screens and printers are typical input devices.
4. The program has to be compiled and tested before it is written.
5. Your source code file is not a program, and it can't be executed, or run, as a program can.
6. To turn your source code into a program, you use a compiler.
7. After your source code is compiled, an object file is produced.
8. C++ code is also called object code.
9. A file that contains object code is called an executable file.
10. The files you create with your editor are called source files.
11. Whatever type of bug you find, you mustn't fix it.

Exercise 3. Word Building

Look at the groups of words and decide what part of speech each word is. Then complete the sentences with the correct word.

program programmer programming programmable

1. A ..... earns £20, 000 a year.
2. A computer ..... is a set of instructions that tells the computer what to do.
3. Converting an algorithm into a sequence of instructions in a programming language is called .....

compile compiler compilation

4. Programs written in a high-level language require ....., or translation into machine code.
5. A ..... generates several low-level instructions for each source language statement.
6. Programmers usually ..... their programs to create an object program and diagnose possible errors.

bug debug debugger debugging

7. It has been estimated that fully 90 percent of the cost of software is the combined cost of ..... and maintenance.

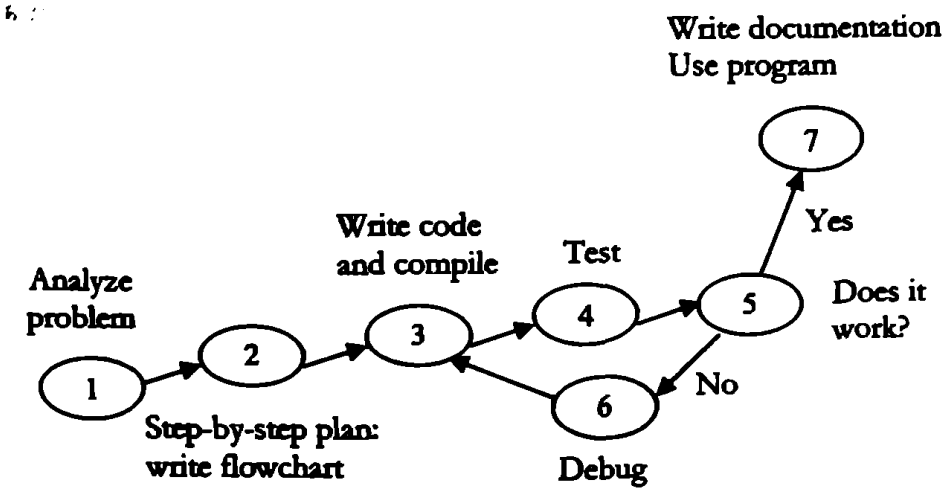
8. There is *always* one more ..... /proverb/.
9. The best compilers usually include an integrated ..... which detects syntax errors.

tester **test** **testing** **testable**

10. *Gamma* ..... is an informal phrase that refers ironically to the release of "buggy" products.
11. During the design phase, ... works with developers in determining what aspects of a design are .....
12. Modern operating systems usually ..... the memory when the computer is switched on.

D. Writing

1. Create a flowchart that describes what you do each morning to get ready for the day.
2. This diagram illustrates the most important programming steps. Write a short description (about 200 words) of these steps on the basis of the diagram. Use time sequencers such as *first*, *then*, *the next step is...*, *finally*, etc.



Life cycle of software

1. Programmer produces code he believes is bug-free.
2. Product is tested. 20 bugs are found.
3. Programmer fixes 10 of the bugs and explains to the testing department that the other 10 aren't really bugs.
4. Testing department finds that five of the fixes didn't work and discovers 15 new bugs.
5. See 3.
6. See 4.
7. See 5.
8. See 6.
9. See 7.
10. See 8.
11. Due to marketing pressure and an extremely pre-mature product announcement based on over-optimistic programming schedule, the product is released.
12. Users find 137 new bugs.
13. Original programmer, having cashed his royalty check, is nowhere to be found.
14. Newly-assembled programming team fixes almost all of the 137 bugs, but introduce 456 new ones.
15. Original programmer sends underpaid testing department a postcard from Fiji. Entire testing department quits.
16. Company is bought in a hostile takeover by competitor using profits from their latest release, which had 783 bugs.
17. New CEO is brought in by board of directors. He hires programmer to redo program from scratch.
18. Programmer produces code he believes is bug-free.
19. See step 2 ...

☺ ☺ ☺

Check Your Progress (2)

Units 4 – 6

Fill in the blanks in the sentences below using information from units 4 – 6.

1. Where a proposition is believed to be true but a direct proof is difficult to secure, an _____ _____ is sometimes obtainable by a process of reasoning called _____ _____.
2. If Eq. (*d*) is valid for $n = k$, it is also _____ for $n = k + 1$.
3. As a second illustration, we shall demonstrate that $x - y$ is a factor of the _____ $x^n - y^n$ for all positive integral values of n.
4. A computer system consists of two parts: the _____ and the hardware.
5. The _____ components are the electronic and mechanical parts of the system.
6. Larger computers may have two or more CPUs, in which case they are simply called "_____" because each is no longer a "central" unit.
7. The _____ _____ holds the instructions and data which are currently being processed by the CPU.
8. The main physical units of a computer system are generally known as the _____.
9. A step-by-step procedure that accomplishes a desired task is called an _____.
10. C++ code is also called _____ _____, and a file containing source code is called a _____ _____.
11. When we compile a simple program, three separate actions occur: first the preprocessor is invoked, then the compiler, and finally the _____.
12. After the programmer _____ a program, it has to be compiled and tested.

SPOKEN NUMBERS AND MEASUREMENTS (2)

1. *The sentences below are written as they would be spoken. Rewrite them as they would normally be written using numbers and abbreviated forms.*

E.g. I take a size fourteen and a half shirt.
I take a size 14½ shirt.

a) The square root of sixteen is four.
b) *a* is not equal to *b*.
c) *y* equals *f* of *r*.
d) *c* prime.
e) In a proper fraction *a* over *b*, *a* is less than *b*.
f) In Figure one-two, *P* is an arbitrary point in the plane.
g) Lviv was founded in twelve fifty-six by Prince Daniel of Galicia.
h) France beat England three nil (football).
i) Brazil and Italy drew three all (football).
j) The score stands at thirty love to Becker (tennis).

2. *Write out the following sentences exactly as they would be spoken, i.e. as in the exercise above.*

a) $\sqrt{25}=5$.
b) $x \neq y$.
c) $f(x) = 0$.
d) a'.
e) p/q, with $p < q$, can be converted to an equivalent decimal fraction.
f) In Fig. 1.4, PQ is a continuous curve.
g) The euro was adopted as a unit of exchange in January 1999.
h) We won 3:0 (football).
i) The final score was 2:2 (football).
j) The score's 15:0 to me at the moment (tennis).

Vocabulary List

algebraic sign
assume, *v*
axis, *n*
complex number
conclude, *v*
conjugate, *adj*
consecutive, *adj*
consider, *v*
constant, *n*, *adj*
degree, *n*
denote, *v*
depressed equation
descending, *adj*
determine, *v*
divide, *v*
division, *n*
ensue, *v*
equation, *n*
even, *adj*
factor theorem
factor, *n*, *v*
figure, *n*, *v*
follow, *v*
following, *n*, *adj*
general form
graph, *n*
illustrate, *v*
illustration, *n*
imaginary, *adj*
integer, *n*
integral, *adj*
intersect, *v*
irrational, *n*, *adj*
let, *v*
multiplication, *n*
negative, *n*, *adj*
odd, *adj*
p form
perfect square
plus sign
point of inflection
polynomial, *n*, *adj*
positive, *n*, *adj*
power, *n*
prove, *v*
rational, *adj*
real, *adj*
relationship, *n*
remainder, *n*
respectively, *adv*
root, *n*
sequence, *n*
set, *v*
sign variation
since, *conj*
single unknown
slope, *n*
solution, *n*
square root
substitution, *n*
successive, *adj*
synthetic division
tangent, *adj*
term, *n*
theorem, *n*
therefore, *adv*
thus, *adv*
unknown, *n*, *adj*
upper limit
valid, *adj*
value, *n*
yield, *v*

A. Brainstorming

– *How many times can you subtract the numeral 2 from the numeral 24?*

- *How can you make four 9s equal 100?*
- *One month has 28 days. Of the remaining 11 months, how many have 30 days?*
- *Imagine that you are a pilot of the plane with 150 people aboard which flies from New York to Paris. How old is the pilot if the distance between the cities is 1500 km?*

B. Reading 1

Polynomial Equations in Single Unknown

by Max Kurtz

Part 1

A polynomial equation of *n*th degree in a single unknown x has the general form

$$a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \cdots + a_n = 0$$

where n is a positive integer, the a's are all constants, and $a_0 \neq 0$. If both sides of the equation are divided by a_0, the equation assumes the following form, which is known as the *p form*:

$$x^n + p_1x^{n-1} + p_2x^{n-2} + \cdots + p_n = 0$$

The equation is described as real, rational, and integral if all the p's are real, rational, and integral.

The polynomial equation of *n*th degree has n roots, but some may be identical. Let $r_1, r_2, r_3, \ldots, r_n$ denote the roots. The remainder theorem (If a polynomial in x is divided by $x - r$, where r is constant, the remainder is $f(r)$) yields the following principle, which is known as the *factor theorem*:

Theorem 7.1. If r_i is a root of the equation $f(x) = 0$, then $x - r_i$ is a factor of $f(x)$.

The converse of this statement is also valid, and therefore the p form of the equation can be recast as

$$(x - r_1)(x - r_2)(x - r_3) \ldots (x - r_n) = 0$$

When the indicated multiplication is performed, the following relationships ensue:

$$\begin{aligned} -p_1 &= r_1 + r_2 + r_3 + \cdots + r_n \\ p_2 &= r_1r_2 + r_1r_3 + \cdots + r_{n-1}r_n \\ -p_3 &= r_1r_2r_3 + r_1r_2r_4 + \cdots + r_{n-2}r_{n-1}r_n \\ &\cdots \\ (-1)^n p_n &= r_1r_2r_3 \cdots r_n \end{aligned}$$

For example, the equation

$$x^4 - 11x^3 + 17x^2 + 107x - 210 = 0$$

has the roots 2, –3, 5, and 7, as can readily be proved by substitution. We have the following:

$$\begin{aligned} -p_1 &= 11 = 2 + (-3) + 5 + 7 \\ p_2 &= 17 = 2(-3) + 2 \times 5 + 2 \times 7 + (-3)5 + (-3)7 + 5 \times 7 \\ -p_3 &= -107 = 2(-3)5 + 2(-3)7 + 2 \times 5 \times 7 + (-3)5 \times 7 \\ p_4 &= -210 = 2(-3)5 \times 7 \end{aligned}$$

When a root r_i of an equation becomes known, it is possible to divide the polynomial by $x - r_i$ and thereby depress the equation to one of $(n - 1)$th degree.

The three theorems that follow yield information pertaining to the nature of the roots of a polynomial equation.

Theorem 7.2. If a real, rational, integral equation has a root r_i that is real and rational, then r_i is an integer and a factor of p_n.

Theorem 7.3. If the complex number $a + b_i$ is a root of a real equation, the conjugate complex number $a - b_i$ is also a root.

Theorem 7.4. If an irrational number of the form $a + \sqrt{b}$ is a root of a rational equation, the irrational number $a - \sqrt{b}$ is also a root.

These theorems are illustrated by the equation

$$x^5 - 6x^4 + 11x^3 + 110x^2 - 614x - 952 = 0$$

which has the roots $3 + 5_i$, $3 - 5_i$, $2 + \sqrt{11}$, $2 - \sqrt{11}$, and –4. The last root is a factor of –952.

When a real polynomial is arranged in descending powers of x, a *sign variation* occurs when two successive terms differ in algebraic

sign. (The powers need not necessarily be consecutive.) As an illustration, consider the polynomial

$$2x^7 - 9x^6 + 12x^5 + 7x^3 - 13x + 5$$

With a plus sign before the first term, the sequence of algebraic signs is + – + + – +, and four sign variations occur.

If the unknown x in a given equation is replaced with $-x$, the resulting equation is termed the *negative* of the given equation. With reference to a real polynomial equation, let P and N denote the number of positive and negative real roots, respectively, and let V and V' denote the number of sign variations in the given equation and in its negative, respectively. The principle that follows, known as *Descartes' rule of signs*, provides a relationship between P and V.

Theorem 7.5. The number P is either equal to V or it is less than V by an even integer.

The same relationship exists between N and V'. Thus, we obtain an upper limit to the values of P and N. A root that is neither a positive nor a negative real number is either imaginary, complex, or 0.

EXAMPLE Determine the nature of the roots of the equation $x^6 - x - 4 = 0$.

SOLUTION The negative equation is $x^6 + x - 4 = 0$. Thus, $V = 1$ and $V' = 1$. Since P and N are restricted to positive values, it follows that $P = N = 1$. The equation has six roots. Therefore, the roots consist of one positive real number, one negative real number, and four complex numbers.

C. Review Questions

Exercise 1. Text-based Translation

Рівняння багаточлена n-го степеня з одним невідомим x; усі a є константами; рівняння набуває такий вигляд...; зведене рівняння; рівняння з дійсними, раціональними, цілими коефіцієнтами; остача; дільник; комплексно-спряжене число; знакозміна; наприклад, розглянемо багаточлен; позначимо через P і N кількість, відповідно, додатних і від'ємних дійсних коренів; корені рівняння; роз-

в'язок; обернене рівняння; правило знаків; верхня границя; з того, що P і N обмежуються додатними величинами, випливає, що...; додатне дійсне число; від'ємне дійсне число; комплексні числа.

Exercise 2. Gap Filling

imaginary	*square root*	*complex number*	*rational*	*real*
set	~~*integer*~~	*irrational*	*negative*	*perfect*

An [1] ___*integer*___ is a whole number, and it can be positive, [2] _____ , or 0. A number that can be expressed in the form p/q, where p and q are integers and $q \neq 0$, is said to be [3] _____ ; a number that cannot be expressed in this form is called [4] _____. For example, it can be demonstrated that the [5] _____ of any positive integer that is not a [6] _____ square is irrational. Since we may [7] _____ $q = 1$, all integers are rational.

The square root of a negative number is described as [8] _____ ; for contradistinction, all other numbers are described as [9] _____ . A number that is formed by combining a real and imaginary number is called a [10] _____ . Thus $5 + \sqrt{-7}$ and $8 - \sqrt{-2}$ are complex numbers.

Exercise 3. Language Work: Perfect Tenses

GRAMMAR NOTE

Past	had	V-en	*I had written* *He had tested*
Present	have / has		*I have written* *He has tested*
Future	shall / will have		*I shall have written* *He will have tested*
Future in the Past	should / would have		*I should have written* *He would have tested*

Translate the following into English using Perfect tenses.

1) Вітаємо! Ви щойно написали, скомпілювали і запустили свою першу програму на C#. 2) Ти вже купив новий процесор? 3) До того часу, як програмне забезпечення надійде у продаж (to go for / on sale), компанія витратить 5 мільйонів доларів на його розроблення. 4) Поліція заарештувала Фреда через три дні після того, як він скачав таємні файли із бази даних ФБР. 5) Вони пообіцяли, що налагодять програму до початку презентації. 6) Очікувалося, що до кінця десятиліття C# стане домінуючою мовою для розроблення комерційного програмного забезпечення. 7) Компанія Senseboard Technologies розробила нову віртуальну клавіатуру, що використовує технологію давачів і штучний інтелект. Тепер користувачі персональних комп'ютерів можуть друкувати на будь-якій твердій поверхні чи навіть у повітрі. 8) Програми стали набагато складнішими, ніж ті, що були написані десять років тому. 9) Хоча сам Ферма (Fermat) заявляв, що він довів теорему, його записи були втрачені, і математики, жоден із яких не міг її розв'язати, часто сумнівалися в існуванні формального доведення.

☺ ☺ ☺

To find a woman you need time and money. Thus,

$$\text{Woman} = \text{Time} \times \text{Money}$$

Since "Time is money", we write

$$\text{Time} = \text{Money}$$

Therefore,

$$\text{Woman} = \text{Money} \times \text{Money}$$

$$\text{Woman} = \text{Money}^2$$

"Money is the root of all problems", i.e.

$$\text{Money} = \sqrt{\text{Problems}}$$

Since

$$\text{Woman} = (\sqrt{\text{Problems}})^2$$

it follows, that

$$\text{Woman} = \text{Problems}$$

☺ ☺ ☺

D. Reading 2

Polynomial Equations in Single Unknown

by Max Kurtz

Part 2

In many instances, Descartes' rule of signs leads to ambiguity concerning the nature of the roots. As an illustration, consider the equation

$$x^5 - 7x^4 + 9x^2 - 2 = 0$$

for which $V = 3$ and $V' = 2$. All we can conclude is that P is either 3 or 1 and N is either 2 or 0. Therefore, the number of complex roots is either 0, 2, or 4.

It is often desirable to establish a range of values within which the real roots of an equation are located. The principle that follows is helpful in this respect.

Theorem 7.6. Consider that the real polynomial $f(x)$ is divided by $x - b$ by synthetic division. If all numbers in the third row are positive, b is greater than all real roots of the equation $f(x) = 0$.

This principle imposes an upper limit on the real roots. A lower limit can be found by forming the negative of the given equation.

If $f(x) = 0$ is a real equation, a wealth of information pertaining to this equation can be obtained by constructing the graph of $f(x)$. Let

r_i = real root of the equation
m_i = number of roots having the value r_i
S_i = slope of graph at $x = r_i$

$$g(x) = \frac{f(x)}{x - r_i}$$

$a + b_i$ = complex root of the equation

The graph of $f(x)$ has the following properties:

1. Slope $S_i = g(r_i)$. Therefore, $S_i = 0$ if $m_i > 1$, and $S_i \neq 0$ if $m_i = 1$.

2. The graph intersects the x axis at $x = r_i$ if m_i is an odd number (including 1).
3. The graph is tangent to the x axis at $x = r_i$ if m_i is an even number.
4. The graph does not intersect the x axis at $x = a$ (unless a is a real root).

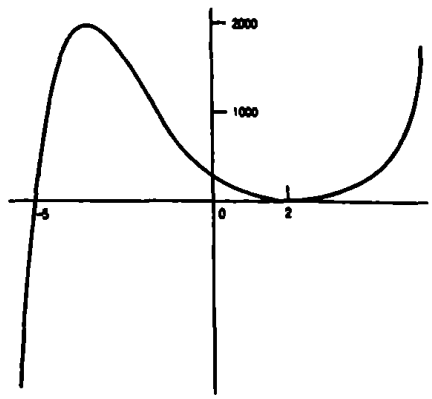


FIGURE 7.1 *Graph of*
$f(x) = x^5 - 5x^4 - 12x^3 + 126x^2 - 280x + 200$

As an illustration consider the equation

$$x^5 - 5x^4 - 12x^3 + 126x^2 - 280x + 200 = 0$$

which has the following roots: $r_1 = -5$; $r_2 = r_3 = 2$; $r_4 = 3 + i$; $r_5 = 3 - i$. Figure 7.1 is the graph of the polynomial. The graph intersects the x axis at $x = -5$, and it is tangent to the x axis at $x = 2$.

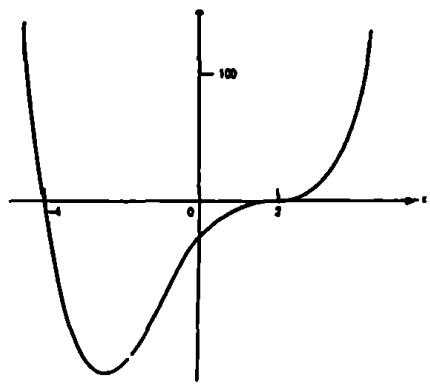


FIGURE 7.2 *Graph of*
$f(x) = x^4 - 2x^3 - 12x^2 + 40x - 32$

Now consider the equation

$$x^4 - 2x^3 - 12x^2 + 40x - 32 = 0$$

which has the following roots: $r_1 = -4$; $r_2 = r_3 = r_4 = 2$. Figure 7.2 is the graph of the polynomial. The graph intersects the x axis at $x = -4$ and at $x = 2$ and it has zero slope at $x = 2$. (The graph has a *point of inflection* at $x = 2$).

Assume that a polynomial equation is real, rational, and integral and that the term p_n in the p form of the equation is not unduly large. The equation can be tested for real integral roots by applying Theorems 7.3 and 7.2, in that order. For example, consider the equation $x^4 + 8x^3 + 5x^2 - 74x - 120 = 0$. The factors of -120 are $1, -1, 2, -2$, etc. Applying each factor in turn and using synthetic division, we find that $x + 2$ is a factor of the polynomial, and therefore -2 is a root of this equation. After division by $x + 2$, the depressed equation is $x^3 + 6x^2 - 7x - 60 = 0$. Repeating the foregoing procedure, we find that 3 is a root of the depressed equation, and therefore of the original equation. Thus, all real integral roots of a given equation can be identified with relative ease.

E. Review Questions

Exercise 1. Text-based Translation

Наприклад, розглянемо рівняння; область значень, у межах якої містяться дійсні корені рівняння; припустимо, що поліном над полем дійсних чисел $f(x)$ ділиться на $x - b$ з остачею; верхня границя; нижня границя; спад; графік $f(x)$ має такі властивості; графік перетинає вісь абсцис; непарне число; графік дотикається до осі x; парне число; рисунок 7.1 є графіком полінома; графік має точку перегину в $x = 2$; рівняння пониженого степеня.

Exercise 2. Matching

Use the words from the two lists: A and B to make ten expressions connected with algebra. Then use the expressions to

complete the sentences. The first one has been done for you as an example.

A	B
synthetic	integer
complex	equation
single	form
positive	~~induction~~
algebraic	limit
~~mathematical~~	unknown
upper	signs
arbitrary	division
polynomial	number
general	constant

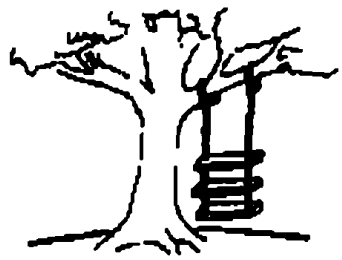
1. The equation can be proved by ***...mathematical induction...***
2. The plus and minus signs are referred to as .....
3. A ..... is formed when a polynomial is set equal to 0.
4. Let n denote a .....
5. A second-degree equation in a single unknown x has the ..... $ax^2 + bx + c = 0$.
6. It is frequently necessary to divide a polynomial in x by a bynomial of the form $x - b$, where b is a constant; the operation can be performed rapidly by the use of .....
7. When several complex numbers are multiplied, the product is also a .....
8. Since all numbers in the third row are positive, the ..... is established.
9. Let $z = f(x, y)$, and let r denote an .....
10. If the equation contains a ...., a solution is called a root of the equation.

F. Problem Solving

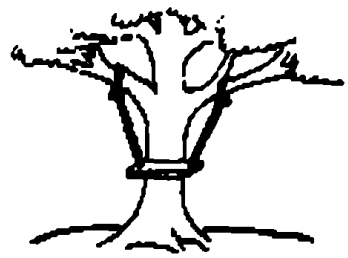
1. Determine the nature of the roots of the equation $x^5 - 8x^2 = 0$.
2. Demonstrate that the real roots of the equation $x^4 - 3x^3 - 5x^2 + 19x - - 60 = 0$ lie between -4 and 5.

☺ ☺ ☺

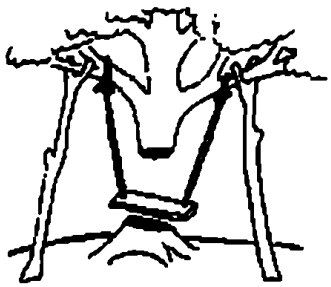
HOW PROGRAMS ARE GENERALLY WRITTEN



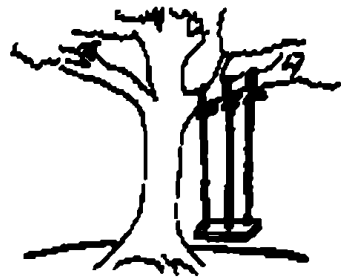
1. How the customer explained it



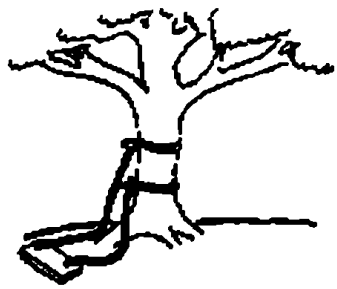
2. How the Project Manager understood it



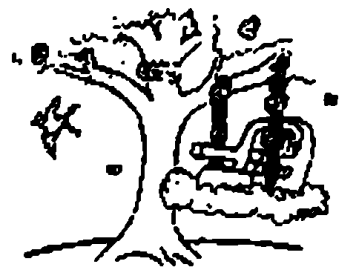
3. How the Analyst designed it



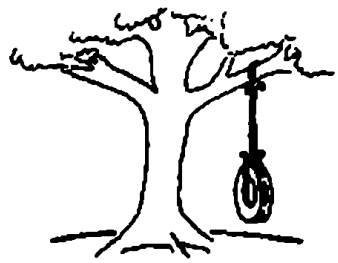
4. How the Programmer wrote it



5. How the project was documented



6. How the Business Consultant described it



7. What the customer really needed

☺ ☺ ☺

Vocabulary List

black box	inheritance, *n*
C#	object, *n*
class, *n*	object-oriented programming
data hiding	polymorphism, *n*
data member	support, *n*, *v*
derived type	user-defined type
encapsulation, *n*	user interface

A. Pre-reading

- *What languages are known as object-oriented?*
- *Who was C# designed by? When did C# 1.0 appear?*
- *What are the differences and similiarities between C# and C++, C# and Java?*

B. Reading

C# and Object-Oriented Programming

by Davis Chapman

C# fully supports object-oriented programming, including the four pillars of object-oriented development: encapsulation, data hiding, inheritance, and polymorphism.

Encapsulation and Data Hiding. When an engineer needs to add a resistor to the device she is creating, she doesn't typically build a new one from scratch. She walks over to a bin of resistors, examines the colored bands that indicate the properties, and picks the one she

needs. The resistor is a "black box" as far as the engineer is concerned – she doesn't much care how it does its work as long as it conforms to her specifications; she doesn't need to look inside the box to use it in her design.

The property of being a self-contained unit is called encapsulation. With encapsulation, we can accomplish data hiding. Data hiding is the highly valued characteristic that an object can be used without the user knowing or caring how it works internally. Just as you can use a refrigerator without knowing how the compressor works, you can use a well-designed object without knowing about its internal data members.

Similarly, when the engineer uses the resistor, she need not know anything about the internal state of the resistor. All the properties of the resistor are encapsulated in the resistor object; they are not spread out through the circuitry. It is not necessary to understand how the resistor works in order to use it effectively. Its data is hidden inside the resistor's casing.

C# supports the properties of encapsulation and data hiding through the creation of user-defined types, called classes. Once created, a well-defined class acts as a fully encapsulated entity – it is used as a whole unit. The actual inner workings of the class should be hidden. Users of a well-defined class do not need to know how the class works; they just need to know how to use it.

Inheritance and Reuse. When the engineers at Acme Motors want to build a new car, they have two choices: they can start from scratch, or they can modify an existing model. Perhaps their Star model is nearly perfect, but they'd like to add a turbocharger and a six-speed transmission. The chief engineer would prefer not to start from the ground up, but rather to say, "Let's build another Star, but let's add these additional capabilities. We'll call the new model a Quasar." A Quasar is a kind of Star, but one with new features.

C# supports the idea of reuse through inheritance. A new type, which is an extension of an existing type, can be declared. This new subclass is said to derive from the existing type and is sometimes called a derived type. The Quasar is derived from the Star and thus inherits all its qualities, but can add to them as needed.

Polymorphism. The new Quasar might respond differently than a Star does when you press down on the accelerator. The Quasar might engage fuel injection and a turbocharger, while the Star would simply let gasoline into its carburetor. A user, however, does not have to know about these differences. He can just "floor it," and the right thing will happen, depending on which car he's driving.

C# supports the idea that different objects do "the right thing" through what is called function polymorphism and class polymorphism. *Poly* means many, and *morph* means form. Polymorphism refers to the same name taking many forms.

C. Review Questions

Exercise 1. Matching

	encapsulation with data hiding	class
object	polymorphism	black box principle
inheritance	C#	object-oriented programming

a) a self-contained entity with an identity and certain characteristics of its own;

b) the ability to distinguish an object's internal state and behavior from its external state and behavior;

c) an object-oriented programming language from Microsoft and ECMA that is based on C++ with elements from Visual Basic and Java.

d) hiding unnecessary implementation details and presenting the simplest possible useful user interface to the client;

e) programming methodology based on treating data and procedures that act upon the data as a single object;

f) the ability to create new types by importing or reusing the description of existing types;

g) a template for objects;

h) the ability of objects to be responsible for interpreting function invocation.

Exercise 2. Open Ended

1. What are the four major OOP language characteristics?
2. In what way is object-oriented analysis and design fundamentally different from other approaches?
3. Are inherited members and functions passed along to subsequent generations? If Dog derives from Mammal, and Mammal derives from Animal, does Dog inherit Animal's functions and data?
4. Suppose you had to simulate the intersection of Park Avenue and 56th Street – two typical two-lane roads, with traffic lights and crosswalks. The purpose of the simulation is to determine if the timing of the traffic signal allows for a smooth flow of traffic. What kinds of objects should be modeled in the simulation? What should be the classes defined for the simulation?

Exercise 3. Quiz: Know Your Abbreviations

Anyone who works with computers needs a good knowledge of abbreviations – short forms of words or phrases. Do you know what these abbreviations stand for? Some are easy, some are more difficult. Check yourself by writing their full expressions as in the example below.

e.g.: **WWW** – World Wide Web – всесвітня павутина.

AI, BIOS, bps, CPU, DP, dpi, DVD, FAQ, FIFO, GUI, HTML, HTTP, LAN, LCD, MIPS, OCR, PIN, RAM, ROM, SCSI, TFT, UPS, USB, WAN, WIMP, WORM, WYSIWYG.

D. Writing

Write a paragraph describing C# (or any other programming language of your choice). You can start like this:

C# (pronounced C Sharp) is a multi-paradigm programming language that encompasses functional, imperative, generic, object-oriented (class-based), and component-oriented programming disciplines...

GAME PROGRAMMING

Vocabulary List

2D, *abbr*
3D, *abbr*
action, *n*
adventure, *n*
artificial intelligence
code, *n*, *v*
computer game
CPU, *abbr*
fighting, *n*
game demo
game designer
game developer/programmer
game loop
gameplay, *n*
gamer, *n*
genre, *n*
hard drive
headphone, *n*
input, *n*, *v*
interface, *n*
joystick, *n*
level, *n*
map, *n*
massively multiplayer online (MMO)
mouse, *n*
operating system
platformer, *n*
play, *v*
puzzle, *n*
racing, *n*
RAM, *abbr*
real time strategy (RTS)
role-playing game (RPG)
screen, *n*
shooter, *n*
side scroller, *n*
simulation, *n*
sports, *n*
system clock
team, *n*
tile, *n*
video card

A. Pre-reading

- *Do you play computer games? If so, what sort of games do you like to play?*
- *What game are you playing now, if any?*
- *Have you ever tried to make your own game? If so, tell your group mates about it.*

B. Reading

"How do I make games?" A Path to Game Development

by Geoff Howland

Starcraft, Everquest, Quake and Counter-Strike were all made by teams of professionals who had budgets usually million dollar plus. All of these games were made by people with a lot of experience at making games. They did not just decide to make games which turned out mega-hits, they started out small and worked their way up.

So where do I start? Tetris is the perfect game to begin your journey on the path to becoming an able bodied game developer. Why? Because Tetris has all the individual components that ALL games share in common. It has a game loop (the process of repeating over and over until the game is quit). The game loop reads in input, processes the input, updates the elements of the game (the falling tetraminos), and checks for victory/loss conditions. Also, you don't have to be an artist to make a good looking Tetris game. Anyone who can draw a block, which is everyone with a paint program, can make a commercial quality version of Tetris. Something I need to mention is that when you make your Tetris game, you can't call it "Tetris". However, this means nothing to you if you call your game "The Sky is Falling", or anything without a "tris" in it.

What's next? After you have totally, completely, absolutely finished your version Tetris, you are ready for your next challenge: Breakout. It is a similar game, but adds in much more advanced collision detection than was necessary in Tetris. Here you will need to add some simple deflection physics of the ball rebounding off different portions of the paddle and the blocks. Level layout also becomes an issue in Breakout, and in order to have more than one level you will need to come up with a way to save the maps. This deals with another component found in all larger games, which is saving and loading resources and switching levels.

After you finish your Breakout masterpiece you should move on to making Pac-Man. Pac-Man is an evolutionary step because it adds in the element of enemy artificial intelligence (AI). You may not have been aware of this, but in the original Pac-Man the four different ghosts had different goals to trying to defeat you as a team. The aggressor would try to follow the shortest path to you, making you directly avoid him. The interceptor would try to go to a junction that was closest to where you would have to move to avoid the aggressor. A second interceptor would try to stay more towards the middle and try to cut you off from using the tunnel through the sides. The last ghost would sort of wander aimlessly about which often kept him staying in a section you needed to finish the map.

Pac-Man also increases the complexity of maps, and adds a good deal more flexibility for using sounds, as sound was certainly a crucial elements to the success of Pac-Man. (After all, what would Pac-Man be without some sort of "wakka-wakka" sound?)

The last game I suggest you should create is a side scroller, such as Super Mario Brothers, where you can jump on multiple platforms, shoot, duck and interact with enemies. Now you must make a screen that is capable of scrolling in at least two directions, if not four, and deal with screen clipping, which can have a bit of a learning curve. You must also work on the physics of any jumping, bouncing of the character or shooting projectiles.

There will additionally need to be a lot more enemies than before, and you will need to keep track of their current game state (alive/dead, active/inactive), by whether they are on the screen or have already been dealt with. The level editor should be capable of placing tiles, scrolling through tiles, scrolling over the map, choosing tiles as brushes, cycling through the brushes, cutting and pasting, an undo, and placing enemies. I would also suggest making back ups of previously saved maps, as it is often easier to just back things up by versions, than redrawing them.

Finally, the side scroller has a real victory condition! When you get to the end of the side scroller, you have actually GONE somewhere, so you can add on a story to progress through the game as well (and don't forget some sort of fireworks on the screen for the end

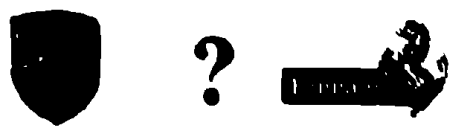
of a level, so that the player has a sense of accomplishment and a REAL show of fireworks for beating the game. Merely putting the words "You Have Won!" on the screen when a player has spent endless hours trying to beat your game is anti-climatic).

But, these games are stupid! Actually, these games clearly show the basis for ALL games' gameplay. Throw a fancy 3D interface over a shooter and it's still a shooter. You could create the same game in a 2D overhead view and the gameplay would be coded exactly the same.

Is it stupid to be able to make a game with EXACTLY the same controls, responses and enemies as Quake? If you remove the 3D interface, and look at what is really happening from a directly overhead view, does it still seem as out of reach?

One thing that you need to clarify to yourself before starting anything, is what you want out of it. Do you want to make games, or just duplicate the technology in Quake? If all you are interested in is the technology, then skip all the games stuff and get started on graphics technology.

I made my game, now where's my Ferrari? Sorry, one game, two games, five games probably won't cut it. Last year there were 3,500 games released on the PC, and only a few handfuls made back a large portion of cash. Most of those that did weren't made by small groups who were self-funded, they were funded by large publishers and probably had multi-million dollar budgets, and definitely near or well over million dollar advertising campaigns. This isn't a world you can't join though, it just takes a good deal of time and experience and track record of making quality games, that hopefully sell well, to give publishers confidence in your team, so that they will entrust you with this kind of financial responsibility.



Decisions, decisions. However, there is more to making a living of games than the multi-million dollar budgets. Just have an understanding of what you really want out of making games and then concentrate on making that come true.

C. Review Questions

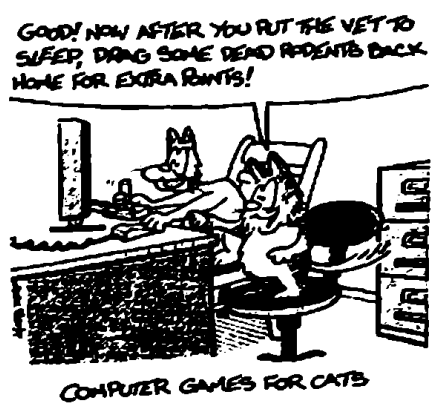
Exercise 1. Gap Filling

headphones	MMORPG	computer games	mouse
joystick	RTS	first-person shooter	video card
Internet	CPU	screen	genres

Personal computer games are commonly referred to as [1]"_____" or "PC games". They are played on the personal computer with standard computer interface devices such as the keyboard and [2]_____, or additional peripherals, such as [3]_____. Video feedback is received by the user through the computer [4]_____, sound through speakers or [5]_____.

The game may perform poorly or not run at all if a computer doesn't meet certain minimum requirements such as [6]_____ speed, Random access memory, system clock speed, [7]_____ memory, hard drive space, operating system, [8]_____ connection speed (for online games) and other criteria.

Games, like most other forms of media, may be categorized into [9]_____ based on gameplay, atmosphere, and various other factors. The most common genres in use today include platformer, adventure, role-playing game (RPG), [10]_____ (FPS), third-person shooter, sports, racing, fighting, action, puzzle, simulation, and real time strategy ([11]_____). The increase in the popularity of online gaming has also resulted in sub-genres being formed, e. g. massively multiplayer online role-playing game ([12]_____).



Exercise 2. Multiple Choice

1. A software engineer who primarily develops computer or video games or related software (such as game development tools) is referred to as a _____.

   a) game designer c) game producer
   b) game programmer d) game publisher

2. At the center of any game program is the _____ that repeatedly performs certain steps until the game is won, lost or interrupted.

   a) game machine c) game CD
   b) game development d) game loop

3. Variant of the game released to a limited audience outside of the company for a user test is termed its _____ version.

   a) alpha b) beta c) gamma d) delta

4. Testers report any _____ that they found.

   a) bugs b) viruses c) gamers d) graphics

5. The Electronic Entertainment Exposition, commonly known as _____, is the world's largest annual trade show of the computer and video games.

   a) EEE b) 3E c) E3 d) E_3

6. When the game is deemed complete and bug-free, it is said to have _____ and is shipped off to the publisher.

   a) "gone gold" c) "gone with the wind"
   b) "gone goose" d) "gone away"

7. _____ is what the player does during the game.

   a) game demo c) game control
   b) gameplay d) game design

8. Computer games often – almost always today – require _____ to fix compatibility problems after their initial release.

   a) game engines b) game assets c) platforms d) patches

Exercise 3. Language Work: ***Perfect Continuous Tenses***

Perfect Continuous	Past	had been	V-ing	*I had been writing* *He had been testing*
	Present	have / has been		*I have been writing* *He has been testing*
	Future	shall / will have been		*I shall have been writing* *He will have been testing*
	Future in the Past	should / would have been		*I should have been writing* *He would have been testing*

Translate the following into English using Perfect Continuous tenses.

1) "Скільки часу ти вже граєш у цю гру?" – "Майже три тижні." 2) До кінця вересня наша команда буде працювати над футбольним стимулятором вже два роки. 3) Я отримую "Computer Games Magazine" упродовж чотирьох місяців, і мені дуже подобається його читати. 4) До того, як Аня перекваліфікувалась на (to retrain for) тестера, вона працювала (to work as) секретаркою. 5) Декан викликав (to call) Андрія у свій кабінет через те, що він не відвідував (to attend) заняття. 6) Учора я допізна (by midnight) тестував Вашу програму. 7) Системний адміністратор сказав, що до четвертої години він модернізуватиме (to upgrade) сервер.

D. Role Play

Student A You are a keen gamer.

Student B You are Student A's parent who strongly dislikes computer games.

Begin your conversation as follows:

Parent: You just sit in front of your computer playing games. That's bad for your eyes, you know...

Check Your Progress (3)

Units 7 – 9

Fill in the blanks in the sentences below using information from units 7 – 9.

1. The equation is described as real, rational, and integral if all the p's are real, rational, and _____.
2. Let $r_1, r_2, r_3, \ldots, r_n$ _____ the roots.
3. If r_i is a root of the equation $f(x) = 0$, then $x - r_i$ is a _____ of $f(x)$.
4. When a real polynomial is arranged in descending powers of x, a _____ _____ occurs when two successive terms differ in algebraic sign.
5. If $f(x) = 0$ is a real equation, a wealth of information pertaining to this equation can be obtained by constructing the _____ of $f(x)$.
6. C# fully supports object-oriented programming, including the four pillars of object-oriented development: encapsulation, _____ _____, inheritance, and polymorphism.
7. C# supports the idea of reuse through _____.
8. This new subclass is said to _____ from the existing type and is sometimes called a derived type.
9. C# supports the idea that different objects do "the right thing" through what is called function _____ and class polymorphism.
10. Tetris is the perfect game to begin your journey on the path to becoming an able bodied _____ _____.
11. Actually, these games clearly show the basis for ALL games' _____.
12. Last year there were 3,500 games _____ on the PC, and only a few handfuls made back a large portion of cash.

SPOKEN NUMBERS AND MEASUREMENTS (2)

1. *The sentences below are written as they would be spoken. Rewrite them as they would normally be written using numbers and abbreviated forms.*

E.g. I take a size fourteen and a half shirt.
I take a size 14½ shirt.

a) In matrix **B**, *b* sub three five equals two.
b) Let **I** be a two by two identity matrix.
c) **A** plus **O** equals **O** plus **A** equals **A**.
d) **A** minus **A** equals **O**.
e) **I** times **A** equals **A** times **I** equals **A**.
f) The norm of **A** equals seventeen.
g) Pluto has a diameter of about one thousand four hundred and fifty five miles (two thousand three hundred and forty kilometers), roughly two-thirds that of the Moon.
h) The third generation of Russian space stations, *Mir* passed its fifteenth anniversary on February the twentieth, two thousand and one.

2. *Write out the following sentences exactly as they would be spoken, i.e. as in the exercise above.*

a) In matrix **A**, $a_{14} = 6$.
b) **A** and **B** are matrices of 3×2 size.
c) $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$.
d) $\mathbf{C} = \mathbf{A} - \mathbf{B}$.
e) $0\mathbf{A} = \mathbf{O}$.
f) $\|\mathbf{C}\| = 1$.
g) Moon is less than 1/3 the size of Earth (diameter about 2,160 mi, or 3,476 km, at its equator), about 1/80 as massive, and about 2/3 as dense.
h) The 21st century and 3rd millennium AD began on Jan. 1, 2001.

Vocabulary List

affine, *adj*
algebraic symbol
array, *n*
assign, *v*
boldface, *n*
bracket, *n*, *v*
column, *n*
column vector
consider, *v*
correspond, *v*
definition, *n*
delete, *v*
denote, *v*
determinant, *n*
dimension, *n*
dot product
element, *n*
equal, *v*, *adj*
foregoing, *adj*
form, *n*
identify, *v*
italicized letter
label, *n*, *v*
lie, *v*
lowercase letter
matrix, *n*
member, *n*
negative matrix
negative transposed matrix
nonzero, *adj*
notation, *n*
null matrix
number, *n*, *v*
order, *n*
parenthesis, *n*
payoff matrix
rectangular, *adj*
reflexive, *adj*
rotation matrix
row, *n*
row vector
size, *n*
submatrix, *n*
subscript, *n*
therefore, *adv*
transform, *v*
transposed matrix
uppercase letter

A. Brainstorming

- *In a certain African village there live 800 women. Three per cent of them are wearing one earring. Of the other 97 per cent, half are wearing two earrings, half are wearing none. How many earrings altogether are being worn by the women?*
- *If nine thousand, nine hundred and nine pounds is written as £9,909, how should twelve thousand, twelve hundred and twelve pounds be written?*
- *A farmer has $4^{7}/_{9}$ haystacks in one corner of the field and $5^{2}/_{9}$ haystacks in another corner of his field. If he puts them all together, how many haystacks will he have?*

B. Reading

Matrix Algebra

by Max Kurtz

A *matrix* is a rectangular array of numbers (or letters that represent numbers). The array is enclosed in brackets to indicate its extent, and each matrix is assigned an uppercase boldface letter for identification.

The numbers that compose the matrix are termed its *elements* or *members*. Elements that lie on a horizontal line constitute a *row*, and those that lie on a vertical line constitute a *column*. The rows and columns are assigned identifying numbers by the following convention: number the rows from top to bottom; number the columns from left to right.

The *size* of a matrix is referred to as its *order* or *dimension*, and it is expressed by specifying the number of rows and the number of columns composing the matrix, in that sequence. Thus, a matrix having m rows and n columns is said to be of order $m \times n$, or it is described as an $m \times n$ matrix.

To illustrate the foregoing definitions, consider the following matrix:

$$\mathbf{A} = \begin{bmatrix} 7 & 5 & 6 & 3 \\ -11 & 8 & -3 & 2 \\ 4 & 9 & 0 & -15 \end{bmatrix}$$

This matrix contains 3 rows and 4 columns; therefore, it is a 3×4 matrix. The third row of the matrix is 4, 9, 0, –15; the second column is 5, 8, 9.

Where the elements of a matrix are represented by algebraic symbols, the practice is as follows: each element is assigned the lowercase italicized letter corresponding to the matrix label, and two subscripts are appended to specify the location of the element. The first subscript is the row number, the second is the column number. For example, the symbol b_{25} (read "b sub two five") identifies the element in matrix **B** that lies in the second row and fifth column. Thus, with reference to the foregoing matrix **A**, we have $a_{13} = 6$ and $a_{32} = 9$.

A matrix that consists of a single row is called a *row vector*, and one that consists of a single column is called *column vector*. A matrix in which all elements are zero is termed a *null matrix*, and designated as 0.

Consider that we have two sets of items, A and B. The item in A and the item in B that have the same identifying number are said to correspond to each other.

Two matrices are *equal* to each other if they are identical in all respects, that is, if they are of the same order and if corresponding elements in the two matrices are equal to each other. If matrices **A** and **B** are equal we write **A** = **B**.

Consider that one or more rows or columns of a matrix **A** are deleted, and let **B** denote the matrix that remains. Then **B** is a submatrix of **A**. For example, let

$$\mathbf{A} = \begin{bmatrix} -3 & 2 & 9 & -12 \\ 11 & 7 & -4 & 1 \\ 5 & 8 & 6 & 3 \end{bmatrix}$$

If the third row and second and fourth columns of **A** are deleted, the submatrix is

$$\mathbf{B} = \begin{bmatrix} -3 & 9 \\ 11 & -4 \end{bmatrix}$$

A row or column of a matrix is described as *nonzero* if it contains at least one nonzero element.

Consider that a given matrix **A** is transformed by converting its rows to columns and its columns to rows, without disturbing the relative order of the rows and columns. Thus, the *i*th row becomes the *i*th column, and the *i*th column becomes the *i*th row. The matrix that results is known as the transpose of **A**, and it is denoted by $\mathbf{A}^T$ or $\mathbf{A}'$.

For example, if $\mathbf{A} = \begin{bmatrix} 3 & 9 \\ 2 & 6 \\ 4 & 1 \\ 8 & 5 \end{bmatrix}$ then $\mathbf{A}^T = \begin{bmatrix} 3 & 2 & 4 & 8 \\ 9 & 6 & 1 & 5 \end{bmatrix}$

Now consider that a given matrix **A** is transformed by changing the algebraic sign of each element. The matrix that results is the negative of **A**, and it is denoted by $-\mathbf{A}$.

In matrix algebra (as in real-number algebra), parentheses are applied to specify the sequence for performing operations. For example, the notation $(-\mathbf{A})^T$ constitutes the following instruction: form the negative of **A**, and then form the transpose of the negative.

If the transpose of $\mathbf{A}^T$ is formed, the result is the original matrix **A**. Expressed symbolically, $(\mathbf{A}^T)^T = \mathbf{A}$ and therefore, the transpose relationship is reflexive. Similarly, we have $-(-\mathbf{A}) = \mathbf{A}$ and therefore the negative relationship is also reflexive.

Consider that a given matrix **A** is transformed to its transpose and the latter is then transformed to its negative. The matrix that evolves is called the *negative transposed matrix.*

There are numerous applications of matrices, both in math and other sciences. For example, 4×4 transformation affine rotation matrices are commonly used in computer graphics. In game theory, the payoff matrix encodes the payoff for two players, depending on which out of a given (finite) set of alternatives the players choose. Early encryption techniques such as the Hill cipher also used matrices. However, due to the linear nature of matrices, these codes are comparatively easy to break.

C. Review Questions

Exercise 1. Jumbled Sentences

1. A matrix, array, elements, is, of, a rectangular.
2. and, rows, Each, has, matrix, columns.
3. a matrix, row, has, is, one, one, only, or, column, that, A vector.
4. columns, equals, In, matrix, the number, the number, of, of, rows, a square.
5. added, are, be, can, if, matrices, only, the same, size, they, Two.
6. A determinant, is, matrix, number, a square, of, a real.
7. the diagonal, entries, is, a matrix, of, of, the sum, The trace.
8. all, are, called, elements, is, matrix, A matrix, whose, a zero, zero.

Exercise 2. Puzzle

In the table, cross out the words, corresponding to the definitions given below, to read the ***mystery clue****. The numbers in brackets at the end of each definition mean the number of letters, forming the sought words.*

⊕	M	A	I	X	E	O	R	T	H	O
C	L	T	R	G	L	L	A	N	O	G
O	I	N	A	N	S	C	H	W	A	R
L	U	E	A	D	S	U	B	M	A	T
N	M	J	R	I	A	X	I	R	T	Z
D	R	O	E	M	L	G	L	E	N	G
A	N	S	N	N	A	E	B	R	O	T
C	Y	I	P	S	R	T	R	A	W	H
A	H	O	O	V	E	O	R	M	B	S
U	C	N	S	E	C	T	M	E	E	R

1. A rectangular array of numbers. (6)
2. A horizontal line which consists of the elements of a matrix. (3)
3. A vertical line which consists of the elements of a matrix. (6)
4. The size of a matrix. (9)
5. The operation denoted as $\mathbf{A}^T$ or $\mathbf{A}'$. (9)
6. Matrix **A** without one or more rows or columns. (9)
7. The numbers that compose a matrix. (7)
8. Matrix which consists of one column. (6)
9. Famous mathematicians who invented and proved the inequality $|\vec{u} \cdot \vec{v}| \leq \|\vec{u}\| \cdot \|\vec{v}\|$ for any $\vec{u}$, $\vec{v} \in \mathrm{R}^n$. (6, 8)
10. Symbol $\|\vec{u}\|$ in the definition $\|\vec{u}\| = \sqrt{v_1^2 + \ldots v_n^2}$. (6)
11. Symbol θ in the definition $\theta = \arccos \dfrac{(\vec{v} \cdot \vec{u})}{\|\vec{v}\| * \|\vec{u}\|}$. (5)
12. Vectors whose dot product is zero. (10)
13. Mathematician who gave name to the matrix $\mathbf{J} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}$. (6)

Exercise 3. *Language Work:* ***Plural of Nouns***

Everyone is familiar with the letter "s" to denote the plural in English. There are certain interesting exceptions to this rule, many of which are scientific or technical terms which have come into English from other languages like Latin and Greek, and which form their plurals in a completely different way.

Below you will find a selection of words. Some are singular, others plural:

matrices	radius	minimum	radix	vertices	axis
moduli	calculus	parentheses	series	analysis	loci
formula	criteria	hypothesis	index	maxima	data

Fill in the columns with the singular and plural forms of the words. The first one has been done for you:

Singular	**Plural**
matrix	matrices
...	...

D. Problem Solving

1. For the following matrices

$$\mathbf{A}=\begin{bmatrix} 3 & 0 \\ -1 & 2 \\ 1 & 1 \end{bmatrix} \quad \mathbf{B}=\begin{bmatrix} 4 & -1 \\ 0 & 2 \end{bmatrix} \quad \mathbf{C}=\begin{bmatrix} 5 & 2 \\ 3 & 5 \\ 6 & 7 \end{bmatrix}$$

find where possible:
a) $4\mathbf{A}+5\mathbf{C}$; b) $\mathbf{AB}$; c) $\mathbf{A}-2\mathbf{C}$.

2. Consider there are only two computer companies in a town. The companies are named "Dude" and "Imac". Each year, Dude keeps $1/5^{th}$ of its customers, while the rest switch to Imac. Each year, Imac keeps $1/3^{rd}$ of its customers, while the rest switch to Dude. If in 2009, Dude has $1/6^{th}$ of the market and Imac has $5/6^{th}$ of the market, what is the distribution of the customers between the two companies in 2010. Write the answer as multiplication of two matrices.

☺ ☺ ☺

DICTIONARY OF DEFINITIONS OF TERMS COMMONLY USED IN MATH. LECTURES

The following is a guide to terms which are commonly used but rarely defined. In the search for proper definitions for these terms we found no authoritative, nor even recognized, source. Thus, we followed the advice of mathematicians handed down from time immortal: "Wing It."

CLEARLY: I don't want to write down all the "in-between" steps.

TRIVIAL: If I have to show you how to do this, you're in the wrong class.

OBVIOUSLY: I hope you weren't sleeping when we discussed this earlier, because I refuse to repeat it.

SIMILARLY: At least one line of the proof of this case is the same as before.

RECALL: I shouldn't have to tell you this, but for those of you who erase your memory tapes after every test...

IT CAN EASILY BE SHOWN: Even you, in your finite wisdom, should be able to prove this without me holding your hand.

CHECK or CHECK FOR YOURSELF: This is the boring part of the proof, so you can do it on your own time.

BY A PREVIOUS THEOREM: I don't remember how it goes (come to think of it I'm not really sure we did this at all), but if I stated it right (or at all), then the rest of this follows.

BRIEFLY: I'm running out of time, so I'll just write and talk faster.

☺ ☺ ☺

COMPUTER CRIMES

Vocabulary List

attachment, *n*
browser, *n*
bug, *n*
computer virus
cracking, *n*
cure, *n*, *v*
cyberlaw, *n*
denial-of-service attack
domain name
dumpster diving
e-mail box
e-mail spoofing
fix, *v*
hacker, *n*
hacking, *n*
hakish, *adj*
ID, *abbr*
IP address
packet sniffing
password, *n*
protect, *v*
resident monitor
salami slicing
server, *n*
software piracy
spam, *n*, *v*
spread, *n*, *v*
VBScript
virus scanner
Web, *n*
website, *n*
workstation, *n*
worm, *n*

A. Pre-reading

- *What threatens the security of computer systems and data?*
- *What do the following names represent: MyDoom, Melissa, Michelangelo?*

B. Reading

ILOVEYOU

From Wikipedia, the free encyclopedia

The **ILOVEYOU** worm, also known as **VBS/Loveletter** and **Love Bug worm**, is a computer worm written in VBScript.

The virus arrived in e-mail boxes on May 4, 2000, with the simple subject of "ILOVEYOU" and an attachment "LOVE-LETTER-

FOR-YOU.TXT.vbs". It spread across the world in just one day (traveling westward from Hong-Kong to Europe to the United States as workers arrived at their offices), infecting 10 percent of all computers connected to the Internet and causing about $5.5 billion in damage. Most of the "damage" was the labor of getting rid of the virus. The worm was responsible for a denial-of-service attack on the official White House website. The Pentagon, CIA, and the British Parliament had to shut down their e-mail systems to get rid of the virus, as did most large corporations. On the whole, this particular malware caused widespread outrage, making it the most damaging virus ever. The virus overwrote important files, as well as music, multimedia and more, with a copy of itself. It also sent the virus to everyone on a user's contact list. Because it was written in Visual Basic Script, this particular virus only affected computers running the Microsoft Windows operating system.

Two aspects of the worm made it effective:

- It exploited the weakness of the e-mail system design that an attached program could be run easily by simply opening the attachment; the underlying mechanism – VBScript – had not been exploited to such a degree previously to direct attention to its potential, thus the necessary layers of protection were not in place.
- It used the right psychological button to entice users to open the e-mail and ensure its continued propagation. The prospect of love is a powerful motivation.

Because the virus used mailing lists as its source of targets, the messages often appeared to come from an acquaintance and so might be considered "safe", providing further incentive to open them. All it took was a few users at each site to access the VBS attachment to generate the thousands and thousands of e-mails that would cripple e-mail systems under their weight, not to mention overwrite thousands of files on workstations and accessible servers.

Variants of the worm:

1. Attachment: LOVE-LETTER-FOR-YOU.TXT.vbs Subject Line: ILOVEYOU Message Body: kindly check the attached LOVELETTER coming from me.
2. Attachment: Very Funny.vbs Subject Line: fwd: Joke Message Body: empty.
3. Attachment: mothersday.vbs Subject Line: Mother's Day Order Confirmation Message Body: We have proceeded to charge your credit card for the amount of $326.92 for the mothers day diamond special. We have attached a detailed invoice to this e-mail. Please print out the attachment and keep it in a safe place.Thanks Again and Have a Happy Mothers Day! mothersday@subdimension.com
4. Attachment: virus_warning.jpg.vbs Subject Line: Dangerous Virus Warning Message Body: There is a dangerous virus circulating. Please click attached picture to view it and learn to avoid it.
5. Attachment: Important.TXT.vbs Subject Line: Important! Read carefully!! Message Body: Check the attached IMPORTANT coming from me!
6. Attachment: Virus-Protection-Instructions.vbs Subject Line: How to protect yourself from the ILOVEYOU bug! Message Body: Here's the easy way to fix the love virus.
7. Attachment: KillEmAll.TXT.VBS Subject Line: I Cant Believe This!!! Message Body: I Can't Believe I have Just Recieved*[sic]* This Hate E-mail. Take A Look!
8. Attachment: ArabAir.TXT.vbs Subject Line: Thank You For Flying With Arab Airlines Message Body: Please, check if the bill is correct by opening the attached file.

Legislative aftermath. The alleged author of the virus Reomel Lamores from Manila, Philippines was briefly held in May 2000 in connection with the virus outbreak. He denied writing the virus, later he claimed the release of the code had been accidental. As there were no laws in the Philippines against virus-writing at the time, he was soon released.

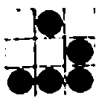
C. Review Questions

Exercise 1. Matching: Computer Crimes

a) spamming	b) dumpster diving	c) denial of service attack
d) hacking	e) salami slicing	f) e-mail spoofing
g) cracking	h) software piracy	i) packet sniffing

1. unauthorized access to a computer system for the fun and challenge of it;
2. the practice of sifting refuse from an office or technical installation to extract confidential data, especially security compromising information;
3. the practice of stealing money repeatedly in extremely small quantities;
4. flooding company websites with service requests so that their website is overloaded and either slowed or crashes completely;
5. capturing data from information packets as they travel over the network;
6. sending commercial e-mail to a large number of addresses without the permission of the recipients;
7. unauthorized use of a third-party domain name as the sender's name in an e-mail message in order to gain illegal entry into a secure system;
8. unauthorized access to a computer system for malicious purposes;
9. copying and/or selling software without the permission of the person or company who owns the copyright.

Exercise 2. Paraphrasing: ***Hacker Slang***

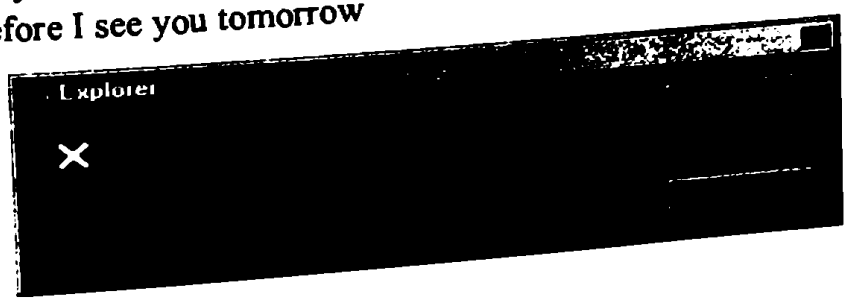


As usual with slang, the special vocabulary of hackers helps hold their culture together – it helps hackers recognize each other's places in the community and expresses shared values and experiences. Also as usual, not knowing the slang (or using it inappropriately) defines one as an outsider, a mundane, or (worst of all in hackish vocabulary) possibly even a suit.

Translate the following hackerisms into plain English by substituting the words in bold type in each sentence (1-10) with one of the phrases (a-j)

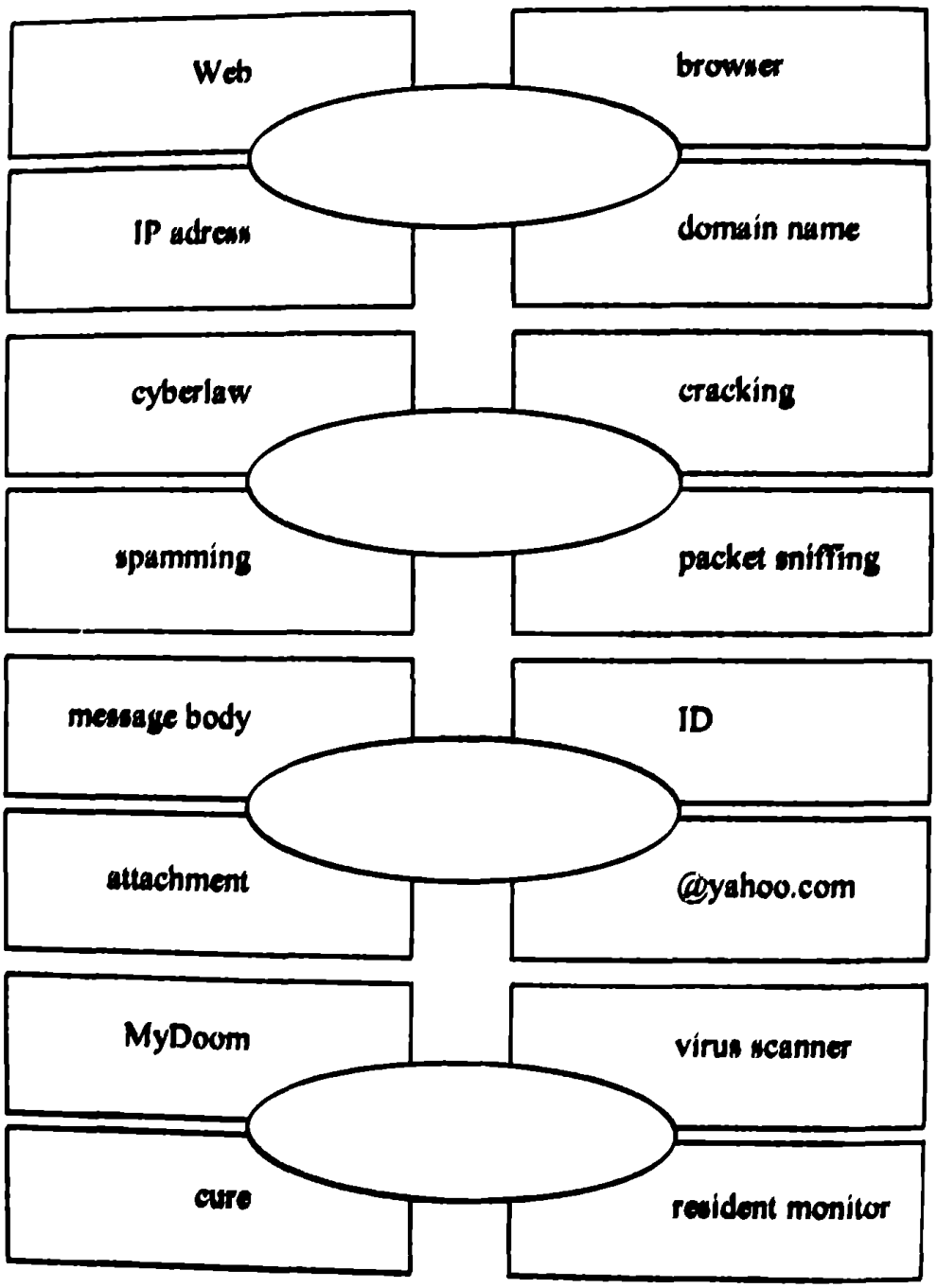
1. I accidentally deleted all my files – **Lose, lose**.
2. His girlfriend is **in beta**.
3. I just need **one bit** from you.
4. It's only a **one-banana** problem; what's taking them so long?
5. **How's hacking**?
6. Fred is a winner, but he has a few **bugs**.
7. **b4 i c u 2moro**.
8. That algorithm is **bogus**.
9. OK, fix that bug and we'll **freeze** for release.
10. **See you on the net!**

a) to ask you one question that can be answered "yes" or "no"
b) lock the evolving software document against changes
c) goodbye
d) being tested for compatibility
e) dear, dear
f) how are you getting on?
g) personality problems
h) incorrect
i) very simple
j) before I see you tomorrow



Exercise 3. Word Associations

What word can you find to connect each set of four? Be prepared to explain your choice: there may be more than one possibility.



D. Discussion Questions

THE ORIGINAL FIREWALL

1. What is your attitude to software piracy?
2. What kind of professional skills should a person possess to function as a hacker?
3. Are hackers criminals or heroes?
4. The University of Calgary set off a controversy last fall by offering students in its computer-science program a course that teaches the ins and outs of writing computer viruses. Critics condemned the course for encouraging criminal activity and threatening to unleash viruses on the Internet, but officials at the Canadian university argued that students who are taught to analyze viruses are better equipped to prevent them. Can writing viruses be taught responsibly? If not, what other steps can colleges take to prepare their students for careers in this key area of computer security?

E. Writing

Write your own science fiction account of the way that a virus might affect society. Speculate and dream all you want, but relate your ideas to the facts that we know about technology and the ways that viruses usually work. If this seems like a crazy idea, think about the movie "Independence Day", where a virus saves the world. Be sure to focus your discussion – you could easily write a novel if you tried to cover everything, but you only need to write a two-page paper. You might write from your personal perspective (what will happen to you?). Or you could write about what happens at our university, to the police department in town, and so forth.

BIOMETRICS

Vocabulary List

access, *n*, *v*
access control
account name
analyze, *v*
application, *n*
authenticate, *v*
authentication, *n*
behavioral, *adj*
biometric, *adj*
biometric data
biometric passport
biometric(s), *n*
card key
chief information security officer
data, *n pl*
detect, *v*
detection, *n*
device, *n*
digital camera
enrollment, *n*
face recognition
fingerprint, *n*, *v*
gait, *n*
hand geometry
identification, *n*
identifier, *n*
identify, *v*
identity, *n*
iris, *n*
keystroke pattern
minutiae, *n*
moiré fringe
optical coupler
palm geometry
password, *n*
pattern, *n*
PIN, *abbr*
protect, *v*
reader, *n*
receptacle, *n*
repository, *n*
retina, *n*
retinal scan(ning)
scam artist
scan, *n*, *v*
secure, *v*, *adj*
SecurID
security, *n*
security personnel
signature, *n*
smart card
technology, *n*
template, *n*
token, *n*
ultrasonics, *n*
verification, *n*
verify, *v*
voice, *n*

A. Pre-reading

- *What are the ways of keeping computer systems safe from criminals, natural hazards, and other threats?*
- *How do you protect your computer equipment?*
- *What is "biometrics"? If you don't know the meaning of the word, here is a clue:*

> *"When they were out of sight Ali Baba came down, and, going up to the rock, said, "Open, Sesame." The door at once opened, and Ali Baba, entering, found himself in a large cave, lighted from a hole in the top, and full of all kinds of treasure – rich silks and carpets, gold and silver ware, and great bags of money. He loaded his three asses with as many of the bags of gold as they could carry; and, after closing the door by saying, "Shut, Sesame," made his way home."*
>
> *(Ali Baba and the Forty Thieves)*

B. Reading

A Practical Guide to Biometric Security Technology

by Simon Liu and Mark Silverman

The security field uses three different types of authentication:

- something you know – a password, PIN, or piece of personal information (such as your mother's maiden name);
- something you have – a card key, smart card, or token (like a SecurID card); and/or
- something you are – a biometric.

Of these, a biometric is the most secure and convenient authentication tool. It can't be borrowed, stolen, or forgotten, and forging one is practically impossible. (Replacement part surgery, by the way, is outside the scope of this article.)

Biometrics measure individuals' unique physical or behavioral characteristics to recognize or authenticate their identity. Common physical biometrics include fingerprints, hand or palm geometry, and retina, iris, or facial characteristics. Behavioral characters include signature, voice (which also has a physical component), keystroke pattern, and gait. Of this class of biometrics, technologies for signature and voice are the most developed.

Fingerprints A fingerprint looks at the patterns found on a fingertip. There are a variety of approaches to fingerprint verification. Some emulate the traditional police method of matching minutiae; others use straight pattern-matching devices; and still others are a bit more unique, including things like moiré fringe patterns and ultrasonics. Some verification approaches can detect when a live finger is presented; some cannot.

A greater variety of fingerprint devices is available than for any other biometric. As the prices of these devices and processing costs fall, using fingerprints for user verification is gaining acceptance – despite the common-criminal stigma.

Fingerprint verification may be a good choice for in-house systems, where you can give users adequate explanation and training, and where the system operates in a controlled environment. It is not surprising that the workstation access application area seems to be based almost exclusively on fingerprints, due to the relatively low cost, small size, and ease of integration of fingerprint authentication devices.

Hand Geometry Hand geometry involves analyzing and measuring the shape of the hand. This biometric offers a good balance of performance characteristics and is relatively easy to use. It might be suitable where there are more users or where users access the system infrequently and are less disciplined in their approach to the system.

Accuracy can be very high if desired, and flexible performance tuning and configuration can accommodate a wide range of applications. Organizations are using hand geometry readers in various scenarios, including time and attendance recording, where they have proved extremely popular. Ease of integration into other systems and processes, coupled with ease of use, makes hand geometry an obvious first step for many biometric projects.

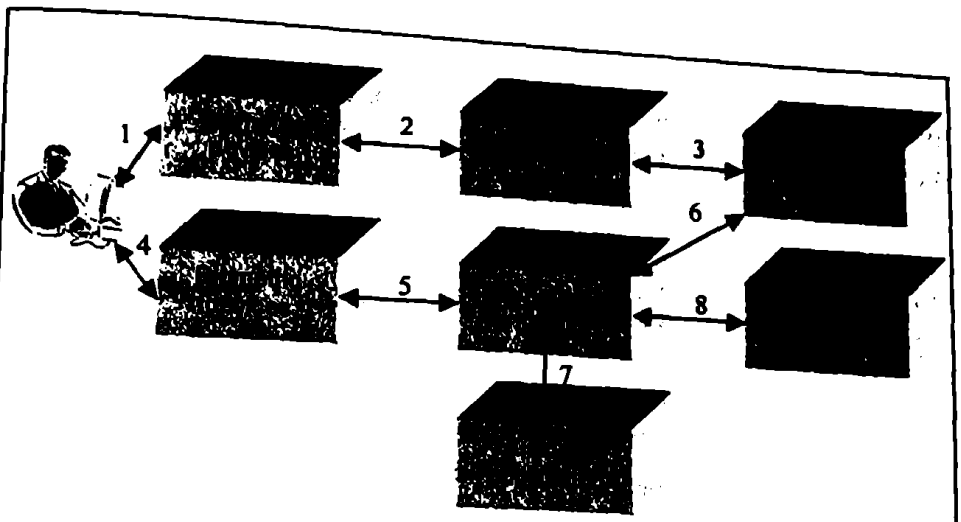


FIGURE 12.1 *How a biometric system works*

(1) Capture the chosen biometrics; (2) process the biometric and extract and enroll the biometric template; (3) store the template in a local repository, a central repository, or a portable token such as a smart card; (4) live-scan the chosen biometric; (5) process the biometric and extract the biometric template; (6) match the scanned biometric template against stored templates; (7) provide a matching score to business applications; (8) record a secure audit trail with respect to system use.

Retina A retina-based biometric involves analyzing the layer of blood vessels situated at the back of the eye. An established technology, this technique involves using a low-intensity light source through an optical coupler to scan the unique patterns of the retina. Retinal scanning can be quite accurate but does require the user to look into a receptacle and focus on a given point. This is not particularly convenient if you wear glasses or are concerned about having close contact with the reading device. For these reasons, retinal scanning is not warmly accepted by all users, even though the technology itself can work well.

Iris An iris-based biometric, on the other hand, involves analyzing features found in the colored ring of tissue that surrounds the pupil. Iris scanning, undoubtedly the less intrusive of the eye-related biometrics, uses a fairly conventional camera element and requires no close contact between the user and the reader. In addition,

it has the potential for higher than average template-matching performance. Iris biometrics work with glasses in place and is one of the few devices that can work well in identification mode. Ease of use and system integration have not traditionally been strong points with iris scanning devices, but you can expect improvements in these areas as new products emerge.

Face Face recognition analyzes facial characteristics. It requires a digital camera to develop a facial image of the user for authentication. This technique has attracted considerable interest, although many people don't completely understand its capabilities. Because facial scanning needs an extra peripheral not customarily included with basic PCs, it is more of a niche market for network authentication. However, the casino industry has capitalized on this technology to create a facial database of scam artists for quick detection by security personnel.

Signature Signature verification analyzes the way a user signs her name. Signing features such as speed, velocity, and pressure are as important as the finished signature's static shape. People are used to signatures as a means of transaction-related identity verification, and most would see nothing unusual in extending this to encompass biometrics. Signature verification devices are reasonably accurate in operation and obviously lend themselves to applications where a signature is an accepted identifier.

Voice Voice authentication is not based on voice recognition but on voice-to-print authentication, where complex technology transforms voice into text. Voice biometrics has the most potential for growth, because it requires no new hardware – most PCs already contain a microphone. However, poor quality and ambient noise can affect verification. In addition, the enrollment procedure has often been more complicated than with other biometrics, leading to the perception that voice verification is not user friendly. Therefore, voice authentication software needs improvement. One day, voice may become an additive technology to finger-scan technology. Because many people see finger scanning as a higher authentication form, voice biometrics will most likely be relegated to replacing or enhancing PINs, passwords, or account names.

Uses for Biometrics For decades, many highly secure environments have used biometric technology for entry access. Today, the primary application of biometrics is in physical security: to control access to secure locations (rooms or buildings). Unlike photo identification cards, which a security guard must verify, biometrics permit unmanned access control. Biometric devices, typically hand geometry readers, are in office buildings, hospitals, casinos, and health clubs. Biometrics are useful for high-volume access control. For example, biometrics controlled access of 65,000 people during the 1996 Olympic Games, and Disney World uses a fingerprint scanner to verify season-pass holders entering the theme park. Physical lock-downs can protect hardware, and passwords are currently the most popular way to protect data on a network. Biometrics, however, can increase a company's ability to protect its data by implementing a more secure key than a password. Using biometrics also allows a hierarchical structure of data protection, making the data even more secure: passwords supply a minimal level of access to network data; biometrics, the next level.

C. Review Questions

Exercise 1. Gap Filling

Comparison of Biometrics

	Ease of Use	Error Incidence	Accuracy
Fingerprints	High	?	High
Hand Geometry	?	Age, hand injury	?
Retina	Low	?	Very high
Iris	Medium	Poor lighting	?
Face	Medium	?	Medium
Signature	High	Changing signatures	?
Voice	High	?	High

Exercise 2. Matching: Biometrics

1. The initial process of collecting biometric data from a user and then storing it in a template for later comparison.
2. The science of identifying individuals by their physical or behavioral characteristics.
3. A mathematical representation of biometric data.
4. The authentication process by which the biometric system matches a captured biometric against the person's stored template (1:1).
5. An Information Age identity document that uses biometrics to authenticate the citizenship of travelers. The current staged biometrics for this type of identification system is fingerprint recognition, digital imaging, and retinal scans.
6. A biometric technique that uses the unique patterns on a person's retina to identify them.
7. A computer technology that identifies human faces in arbitrary images. It detects facial features and ignores anything else, such as buildings, trees and bodies.
8. The practice of restricting entrance to a facility or property to authorized persons.
9. The process by which the biometric system identifies a person by performing a one-to-many (1:*n*) search against the entire enrolled population.

a) access control

b) biometrics

c) biometric passport

d) enrollment

e) face recognition

f) identification

g) retinal scan

h) template

i) verification

Exercise 3. Language Work: Passive Voice

Change the following into the Passive Voice.

e. g.: We use electronic chips to make powerful computers.
Electronic chips are used to make powerful computers.

Simple	Past	was / were	V-en	*I was tested* *They were written*
	Present	am / is / are		*I am tested* *They are written*
	Future	shall / will be		*I shall be tested* *They will be written*
	Future in the Past	should / would be		*I should be tested* *They would be written*

1. Windows will now check the disks.
2. Someone hacked into the Pentagon computer.
3. Delon said, he would teach his sister to use the word processor.
4. Leonardo da Vinci made one of the first recorded designs of a humanoid robot in around 1495.
5. We denote a function by the letter *f*.

Continuous	Past	was / were being	V-en	*I was being tested* *They were being written*
	Present	am / is / are being		*I am being tested* *They are being written*

6. Windows is verifying files and folders.
7. Setup is inspecting your computer's hardware configuration.
8. At that time we were designing a children's game – a simulation of a farm.
9. E-commerce developers are exploring the use of biometrics and smart cards.
10. The U.S. military were using bomb disposal robots such as iRobot's Packbot to defuse roadside bombs.

Perfect	Past	had been	V-en	*I had been tested* *They had been written*
	Present	have / has been		*I have been tested* *It has been written*
	Future	shall / will have been		*I shall have been tested* *They will have been written*
	Future in the Past	should /would have been		*I should have been tested* *They would have been written*

11. Have you unplugged a network cable?
12. DS-Media Group will have released the film on DVD by the end of July.
13. He asked me why I hadn't installed antivirus on my machine yet.
14. I overheard, that they would have upgraded Jane to senior developer by the end of the year.
15. We have proposed and implemented a fast and reliable algorithm for face detection.

D. Role Play

Job Title: **INFORMATION SECURITY OFFICER**
Company: Robert Half Technologies
Location: US-NY-New York
Base Pay: N/A
Employee Type: Full-Time Employee

We have an excellent and challenging employment opportunity for a highly motivated and innovative individual to serve as the Chief Information Security Officer.

This position will report directly to the Executive Director and will provide the organization with tactical information and security advice aimed at examining the ramifications of current and new technologies, in order to establish overall accountability for information security.

Student A Act as an interviewer at Robert Half Technologies. The company is seeking for a person to fill the vacancy of the chief information security officer. Prepare the questions you should ask every candidate.

Student B
Student C
Student D You are applying for the position of the chief information security officer of a high tech company. Your predecessor was dismissed for secret information leakage.

Establish your own information protection program, including firewalls, intrusion detection systems, and anti-virus software. Prove you are the best contender for the job.

☺ ☺ ☺

New York (CNN): At John F. Kennedy International Airport today, a high school mathematics teacher was arrested trying to board a flight while in possession of a compass, a protractor and a graphical calculator. According to law enforcement officials, he is believed to have ties to the Al-Gebra network. He will be charged with carrying weapons of math instruction. It was later discovered that he taught his students to solve their problems with the help of radicals!

☺ ☺ ☺

Check Your Progress (4)
Units 10 – 12

Fill in the blanks in the sentences below using information from units 10 – 12.

1. A _____ is a rectangular array of numbers (or letters that represent numbers).
2. The numbers that compose the matrix are termed its _____ or members.
3. The size of a matrix is referred to as its _____ or dimension.
4. A matrix in which all elements are zero is termed a _____ _____, and it is designated as 0.
5. The matrix that results is the _____ of A, and it is denoted by –A.
6. The virus arrived in e-mail boxes on May 4, 2000, with the simple subject of "ILOVEYOU" and an _____ "LOVE-LETTER-FOR-YOU.TXT.vbs".
7. The worm was responsible for a denial-of-service attack on the official White House _____.
8. He denied writing the virus, later he claimed the release of the _____ had been accidental.
9. _____ measure individuals' unique physical or behavioral characteristics to recognize or authenticate their identity.
10. A fingerprint looks at the patterns found on a _____.
11. _____ _____is undoubtedly the less intrusive of the eye-related biometrics.
12. _____ biometrics has the most potential for growth, because it requires no new hardware – most PCs already contain a microphone.

1. *The sentences below are written as they would be spoken. Rewrite them as they would normally be written using numbers and abbreviated forms.*

E.g. I take a size fourteen and a half shirt.
I take a size 14½ shirt.

a) The absolute value of negative nine is nine.
b) x is congruent to y modulo thirteen.
c) Twelve point three four approximately equals twelve point three.
d) y equals the logarithm of x to the base five.
e) Capital A equals the integral from a to b of $y\ dx$.
f) The sum from j equals one to j equals n of p sub j, x sub j.
g) The product from i equal to one to three of a sub ii.
h) a sub k sup two.

2. *Write out the following sentences exactly as they would be spoken, i.e. as in the exercise above.*

a) $|-3| = 3$.
b) $5 \equiv 23 \pmod 9$.
c) $3.2465 \approx 3.25$.
d) $\log_2 8 = 3$.
e) $A = \int_c^d x\ dy$.
f) $\sum_{k=1}^{n} x_k y_k$.
g) $\prod_{k=1}^{n} a_{kk}$.
h) z_n^3.

MODULO ARITHMETIC

Vocabulary List

algebra, *n*
apply, *v*
assign, *v*
boundary, *n*
circle, *n*, *v*
clockwise, *adj*, *adv*
congruence, *n*
congruent, *adj*
constant, *n*, *adj*
conversely, *adv*
corresponding, *adj*
counterclockwise, *adv*
cycle, *n*, *v*
cyclically, *adv*
decimal number
decrease, *n*, *v*
deduce, *v*
denote, *v*
depict, *v*
diagram, *n*, *v*
digit, *n*
divide, *v*
dividend, *n*
divisible, *adj*
divisor, *n*
equality, *n*
equation, *n*
express, *v*
factor, *n*, *v*
follow, *v*
geometrically, *adv*
illustrate, *v*
illustration, *n*
increase, *v*
indefinitely, *adv*
integer, *n*
label, *n*, *v*
let, *v*
linear congruential sequence
linear equation
modulo arithmetic
modulus, *n*
multiple, *n*, *adj*
multiply, *v*
negative, *n*, *adj*
nonnegative, *adj*
notation, *n*
number, *n*
obtain, *v*
parenthesis, *n*
point, *n*
positive, *n*, *adj*
transposition, *n*
proceed, *v*
proof, *n*
pseudo-random, *adj*
quotient, *n*
range, *n*, *v*
recur, *v*
recurrent, *adj*
relationship, *n*
remainder, *n*
seed, *n*
sequence, *n*
set, *n*, *v*
space, *n*
state diagram
statement, *n*
succession, *n*
successively, *adv*
table, *n*
transpose, *n*, *v*
uppercase letter
valid, *adj*
value, *n*

A. Pre-reading

- *Two fathers and two sons shot three deer. Yet each took home one deer. How was that possible?*
- *You go to bed at 8 o'clock in the evening and set the alarm to get at 9 in the morning. How many hours of sleep would this allow you?*

B. Reading

Modulo Arithmetic

by Max Kurtz

When an integer a is divided by another integer m, the result consists of a quotient q and a remainder r. (We include cases where $q = 0$ or $r = 0$.) In many instances, our interest centers about the remainder exclusively, the quotient being of no consequence. The branch of algebra that concerns itself with the study of remainder is known as *modulo arithmetic*. In the subsequent material, it is understood that all numbers are integers.

The divisor m is called the *modulus*, and we shall restrict it to positive values. The dividend a can be expressed as $a = qm + r$, where $|r| < m$, and a can be positive, negative, or 0. In modulo arithmetic, the remainder r is generally restricted to nonnegative values. Where a is negative, r can be made positive by decreasing q by 1. For example, if 17 is the modulus, we have $-81 = (-4)\ 17 - 13$; thus, $q = -4$ and $r = -13$. However, we can express -81 in the alternative form $-81 = (-5)\ 17 + 4$, making $q = -5$ and $r = 4$.

Let a and b denote two numbers that have the same remainder when they are divided by m. Then a and b are said to be *congruent* to each other with respect to m. In the notation devised by Gauss, we write

$$a \equiv b \pmod{m}$$

and this statement is read "a is congruent to b modulo m."

As an illustration, let $m = 7$, $r = 3$, and $a = 7q + 3$. Allowing q to vary from -2 to 2, we obtain the following set of congruences:

$$-11 \equiv -4 \equiv 3 \equiv 10 \equiv 17 \qquad (\text{mod } 7)$$

Congruences can be depicted geometrically by placing numbers on a circle, which is known as the *congruence circle*. To illustrate the procedure, we shall take 7 as the modulus. In Fig. 13.1, we divide a circle into seven equal parts and label the boundaries with uppercase letters. Starting at A and proceeding in a clockwise direction, we place the number 0 at A, 1 at B, 2 at C, etc., for two cycles. Now returning to A and proceeding in a counterclockwise direction, we place the number -1 at G, -2 at F, etc. Numbers that lie at the same point are congruent to one another. Thus, taking point E, we find that $-3 \equiv 4 \equiv 11$ (mod 7).

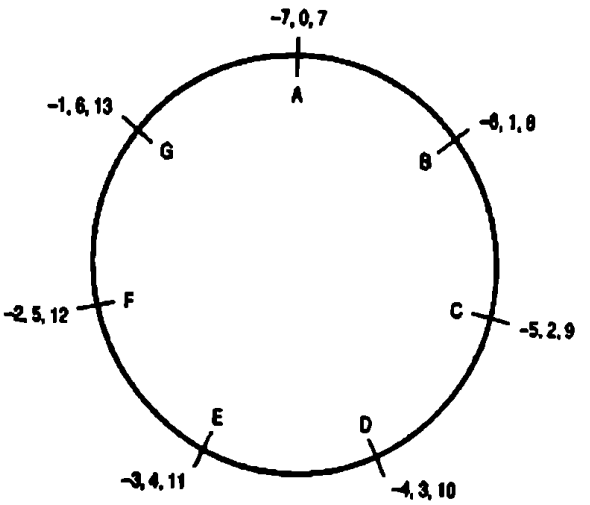


FIGURE 13.1 *Congruence circle*

The congruence circle confers a practical meaning upon the congruence relationship. Thus, consider that we spin a wheel in a game of chance. What is significant is merely the point at which the marker comes to rest; how many revolutions the wheel makes is irrelevant. Thus, with reference to Fig. 13.1, it would be immaterial whether the marker traversed 2, 9, or -5 spaces.

If $a \equiv b \pmod{m}$, then $a - b$ is a multiple of m, and we may write $a - b = pm$, or $a = b + pm$. Conversely, if $a - b$ is a multiple of m, then $a \equiv b \pmod{m}$. Now let $a \equiv a'$ and $b \equiv b' \pmod{m}$. Then

$$a = a' + cm \qquad b = b' + dm$$

and the following relationships exist:

$$\begin{aligned} a + b &= a' + b' + (c + d)m \\ a - b &= a' - b' + (c - d)m \\ ab &= a'b' + (a'd + b'c)m + cdm^2 \end{aligned}$$

It follows that

$$\begin{aligned} a + b &\equiv a' + b' \pmod{m} \\ a - b &\equiv a' - b' \pmod{m} \\ ab &\equiv a'b' \pmod{m} \end{aligned} \qquad (e)$$

For example, since $12 \equiv 21$ and $5 \equiv 23 \pmod 9$, the foregoing equations yield the following congruences: $17 \equiv 44$, $7 \equiv -2$, and $60 \equiv 483 \pmod 9$.

Let m' be a factor of m. If $a \equiv b \pmod{m}$, then $a \equiv b \pmod{m'}$. The proof is as follows: Let $m = nm'$. From the foregoing discussion, we have $a - b = pm = pnm'$. Thus, $a - b$ is a multiple of m', and the conclusion follows. For example, since $23 \equiv 51 \pmod{14}$, then $23 \equiv 51 \pmod 7$. The converse of this statement is not necessarily true. For example, $105 \equiv 15 \pmod 6$, but $105 \equiv 9 \pmod{12}$ and $15 \equiv 3 \pmod{12}$.

The principle of transposition, that pertains to equations is of course valid for congruences as well. For example, taking 11 as modulus, we have $34 - 19 \equiv 4$. Transposing the 19, we obtain $34 \equiv 23 \equiv 1$.

From Eq.(e), we deduce the following: If $a \equiv a' \pmod{m}$, then $a^n \equiv a'^n \pmod{m}$. This equality yields interesting relationships between a decimal number and its digits. We shall illustrate these relationships with 13 as modulus. We start with $10 \equiv -3$. Multiplying successively by -3, we obtain the following: $10^2 \equiv 9$, $10^3 \equiv -27 \equiv -1$, $10^4 \equiv 3$, $10^5 \equiv -9$, $10^6 \equiv 27 \equiv 1$. From this point on, the set of numbers -3, 9, -1, 3, -9, 1 recurs cyclically. By transposition, we obtain $10 + 3 \equiv 0$, $10^2 - 9 \equiv 0$, $10^3 + 1 \equiv 0$, etc.

Now consider that we have a seven-place decimal number $x = d_6d_5d_4d_3d_2d_1d_0$. Then

$$x = d_0 + 10d_1 + 10^2d_2 + 10^3d_3 + 10^4d_4 + 10^5d_5 + 10^6d_6$$

We now form the corresponding number

$$y = d_0 - 3d_1 + 9d_2 - d_3 + 3d_4 - 9d_5 + d_6$$

Then

$$x - y = (10 + 3)d_1 + (10^2 - 9)d_2 + (10^3 + 1)d_3 + (10^4 - 3)d_4 + \\ + (10^5 + 9)d_5 + (10^6 - 1)d_6$$

Applying the previous results, we find that each expression in parentheses has a remainder of 0; that is, it is divisible by 13. Therefore, $x - y$ is divisible by 13.
Then

$$x - y \equiv 0 \pmod{13} \qquad x \equiv y \pmod{13}$$

Thus, when x is divided by 13, it has the same remainder as y. In particular, if y is divisible by 13, x is also divisible by 13.

As an illustration, let $x = 5{,}182{,}741$. Then

$$y = 1 - 3 \times 4 + 9 \times 7 - 2 + 3 \times 8 - 9 \times 1 + 5 = 70$$

When x and y are divided by 13, the remainder is 5 in both instances.

By selecting 11 as the modulus and proceeding in the same manner, we discover that a decimal number is divisible by 11 if and only if the corresponding number $y = d_0 - d_1 + d_2 - d_3 + \cdots$ is divisible by 11. For example, the number 7, 481, 639 is divisible by 11 because the corresponding number $9 - 3 + 6 - 1 + 8 - 4 + 7 = 22$ is divisible by 11.

A *linear congruential sequence* is a set of numbers that is generated by starting with an initial number (or *seed*) X_0 and then repeatedly applying the recursive linear equation

$$X_{n+1} \equiv aX_n + c \pmod{m}$$

Where X_n is the nth number in the sequence, a and c are constants, all numbers are positive, and X_n, a, and c can range from 0 to $m - 1$. Since the number of possible values of X_n is m, every linear

congruential sequence eventually becomes a recurrent cycle of numbers.

To illustrate the procedure for generating a linear congruential sequence, we shall set $X_0 = 2$ and apply the equation

$$X_{n+1} \equiv 4X_n + 7 \pmod 9$$

Then $X_1 \equiv 6, X_2 \equiv 4, X_3 \equiv 5, X_4 \equiv 0, X_5 \equiv 7, X_6 \equiv 8, X_7 \equiv 3, X_8 \equiv 1, X_9 \equiv 2$, and the set of numbers from X_0 to X_8 then recurs cyclically.

For our illustrative case, the following table records every possible value of X_n and the corresponding value of X_{n+1}:

X_n	0 1 2 3 4 5 6 7 8
X_{n+1}	7 2 6 1 5 0 4 8 3

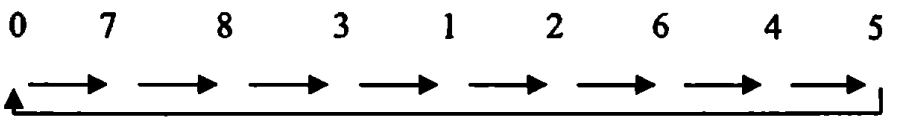


FIGURE 13.2 *State diagram*

The results can be tested for consistency by constructing the congruence circle. When X_n increases by 1, the point corresponding to X_{n+1} advances four places in the clockwise direction.

The succession of numbers in our illustrative linear congruential sequence can be depicted readily by the diagram in Fig. 13.2. This diagram contains every possible value of X_n, and an arrow leads from X_n to X_{n+1}. Thus, if we start with $X_0 = 8$, we obtain the recurrent sequence 8, 3, 1, 2, 6, 4, 0, 7. Figure 13.2 is called a *state diagram.*

If we now apply the equation

$$X_{n+1} \equiv 3X_n + 7 \pmod 9$$

we obtain the state diagram in Fig. 13.3. Whatever value X_0 may have, the sequence eventually culminates in 1, and this number then recurs indefinitely.

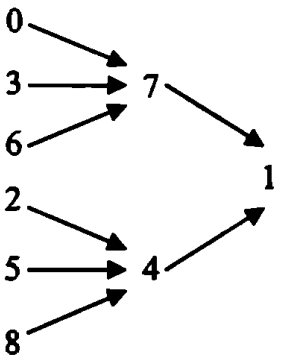


FIGURE 13.3 *State diagram*

Linear congruential sequences are often applied to generate a set of pseudo-random numbers. The sequence must be designed to give the recurrent cycle of numbers the minimum desired length, and this is accomplished by assigning values to m, a, c, and X_0 in accordance with certain rules.

C. Review Questions

Exercise 1. Text-based Translation

Частка q і остача r; розділ алгебри; модульна арифметика; дільник m називається модулем; два числа, які при діленні на m дають однакову остачу; у записі, запропонованому Гауссом; a конгруентне b за модулем m; на рисунку 13.1 ми ділимо коло на сім рівних частин і позначаємо межі великими літерами; починаючи з A і рухаючись за годинниковою стрілкою, покладемо число 0 в A; десяткове число і його цифри; семизначне десяткове число; вираз у дужках; початкове число; лінійне рівняння; c може набувати значення від 0 до $m - 1$; діаграма станів; псевдовипадкові числа.

Exercise 2. Matching

1. a number by which another number is divided in a congruence;
2. a statement that two mathematical expressions are equal;
3. the branch of mathematics that deals with the general properties of numbers, and generalizations arising therefrom;
4. one of the written signs that represent the numbers 0 to 9;
5. a number or polynomial that is divided by another number or polynomial;
6. a cyclic permutation of two members of a set;
7. the result of dividing one number or polynomial by another;
8. a number or polynomial that divides another number or polynomial;
9. a number that is a product of a given number and an integer;
10. a positive integer;
11. a number remaining after one number is divided into another an exact number of times;
12. a completely round shape;
13. a mathematical quantity shown by a letter of the alphabet or sign;
14. a statement that two numbers or geometric figures have the difference divisible by a given modulus;
15. a pictorial representation of a quantity or of a relationship.

a) quotient
b) remainder
c) multiple
d) number
e) divisor
f) congruence
g) modulus
h) diagram
i) value
j) dividend
k) circle
l) algebra
m) equation
n) transposition
o) digit

Exercise 3. Opposites

A	B
The opposite of *positive* is _____	counterclockwise
The opposite of *even* is _____	division
The opposite of *real* is _____	imaginary
The opposite of *rational* is _____	indirect
The opposite of *equality* is _____	increase
The opposite of *integral* is _____	odd
The opposite of *clockwise* is _____	invalid
The opposite of *direct* is _____	lowercase
The opposite of *valid* is _____	irrational
The opposite of *given* is _____	arbitrary
The opposite of *constant* is _____	subtract
The opposite of *multiplication* is _____	negative
The opposite of *uppercase* is _____	variable
The opposite of *add* is _____	inequality
The opposite of *decrease* is _____	nonintegral

D. Problem Solving

You may have heard of Fermat's last theorem. Its proof involves modular arithmetic (and **much** more!). Fermat also used modular arithmetic to discover many other interesting things about numbers. For instance: can you find two numbers x and y so that the sum of their squares, $x^2 + y^2$, is exactly 3 more than a multiple of 4, that is

$$x^2 + y^2 = 4n + 3 \quad \text{or} \quad x^2 + y^2 = 3 \pmod 4?$$

Try it!
(Please choose x and y between 0 and 9999 to prevent overflow).

CRYPTOGRAPHY

Vocabulary List

attacker, *n*
cipher, *n*, *v*
ciphertext, *n*
cleartext, *n*
conventional cryptography
conventional encryption
crack, *n*, *v*
cryptanalysis, *n*
cryptographic algorithm
cryptography, *n*
cryptology, *n*
cryptosystem, *n*
data, *n pl*
Data Encryption Standard (DES)
decipher, *n*, *v*
decrypt, *v*
decryption, *n*
digital signature
encode, *v*
encrypt, *v*
encryption, *n*
hybrid cryptosystem
information, *n*
key, *n*
key distribution
keyring, *n*
message, *n*
PGP, *abbr*
plaintext, *n*
private key
public key
public key cryptography
random number
receiver, *n*
recipient, *n*
secret key
secret-key encryption
secure, *v*, *adj*
security, *n*
sender, *n*
session key
symmetric-key encryption
transmit, *v*
transmission, *n*

A. Pre-reading

- *The ancient Spartans were the first to use secret messages over 2000 years ago. Who uses secret messages today and why?*
- *Do you know any methods of "secret writing"? If so, tell your group mates about it.*

B. Reading

Introduction to Cryptography

From The International PGP Home Page

The Basics of Cryptography When Julius Caesar sent messages to his generals, he didn't trust his messengers. So he replaced every A in his messages with a D, every B with an E, and so on through the alphabet. Only someone who knew the "shift by 3" rule could decipher his messages.

And so we begin.

Encryption and Decryption Data that can be read and understood without any special measures is called *plaintext* or *cleartext*. The method of disguising plaintext in such a way as to hide its substance is called *encryption*. Encrypting plaintext results in unreadable gibberish called *ciphertext*. You use encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting ciphertext to its original plaintext is called *decryption*. Figure 14.1 illustrates this process.

FIGURE 14.1. *Encryption and decryption*

What Is Cryptography? *Cryptography* is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks so that it cannot be read by anyone except the intended recipient.

While cryptography is the science of securing data, *cryptanalysis* is the science of analyzing and breaking secure communication.

Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called *attackers.*

Cryptology embraces both cryptography and cryptanalysis.

How Does Cryptography Work? A *cryptographic algorithm,* or cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a *key* – a word, number, or phrase – to encrypt the plaintext. The same plaintext encrypts to different ciphertext with different keys. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key.

A cryptographic algorithm, plus all possible keys and all the protocols that make it work comprise a *cryptosystem.* PGP is a cryptosystem.

Conventional Cryptography In conventional cryptography, also called *secret-key* or *symmetric-key* encryption, one key is used both for encryption and decryption. The Data Encryption Standard (DES) is an example of a conventional cryptosystem that is widely employed by the Federal Government. Figure 14.2 is an illustration of the conventional encryption process.

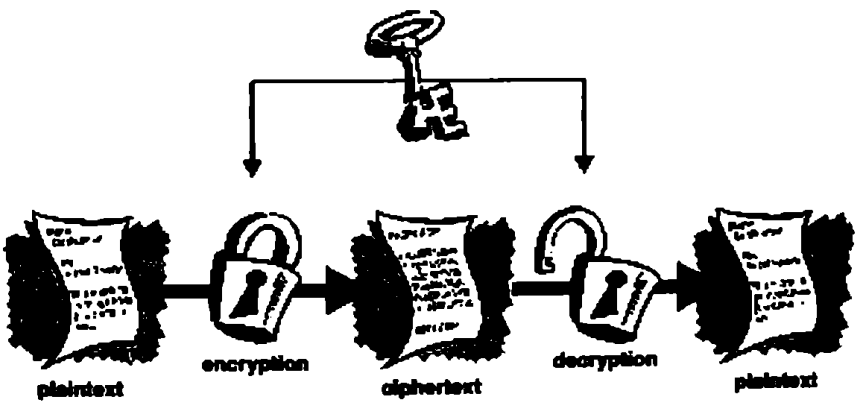


FIGURE 14.2. *Conventional encryption*

Caesar's Cipher An extremely simple example of conventional cryptography is a substitution cipher. A substitution cipher substitutes one piece of information for another. This is most frequently done by offsetting letters of the alphabet.

For example, if we encode the word "SECRET" using Caesar's key value of 3, we offset the alphabet so that the 3rd letter down (D) begins the alphabet.

So starting with

ABCDEFGHIJKLMNOPQRSTUVWXYZ

and sliding everything up by 3, you get

DEFGHIJKLMNOPQRSTUVWXYZABC

where D = A, E = B, F = C, and so on.

Using this scheme, the plaintext, "SECRET" encrypts as "VHFUHW". To allow someone else to read the ciphertext, you tell them that the key is 3.

Obviously, this is exceedingly weak cryptography by today's standards, but hey, it worked for Caesar, and it illustrates how conventional cryptography works.

Key Management and Conventional Encryption Conventional encryption has benefits. It is very fast. It is especially useful for encrypting data that is not *going* anywhere. However, conventional encryption alone as a means for transmitting secure data can be quite expensive simply due to the difficulty of secure key distribution.

For a sender and recipient to communicate securely using conventional encryption, they must agree upon a key and keep it secret between themselves. If they are in different physical locations, they must trust a courier, the Bat Phone, or some other secure communication medium to prevent the disclosure of the secret key during transmission. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all information encrypted or authenticated with that key. Thus, the persistent problem with conventional encryption is *key distribution:* how do you get the key to the recipient without someone intercepting it?

Public Key Cryptography The problems of key distribution are solved by *public key cryptography,* the concept of which was introduced by Whitfield Diffie and Martin Hellman in 1975. Public key cryptography is an asymmetric scheme that uses a *pair* of keys for encryption: a *public key,* which encrypts data, and a corresponding *private,* or *secret key* for decryption. You publish your public key to the world while keeping your private key secret. Anyone with a copy

of your public key can then encrypt information that only you can read. Even people you have never met.

It is computationally infeasible to deduce the private key from the public key. Anyone who has a public key can encrypt information but cannot decrypt it. Only the person who has the corresponding private key can decrypt the information.

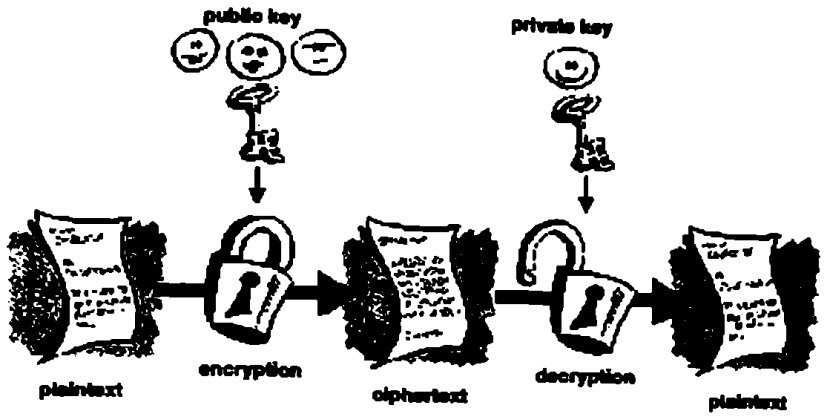


FIGURE *14.3. Public key encryption*

The primary benefit of public key cryptography is that it allows people who have no preexisting security arrangement to exchange messages securely. The need for sender and receiver to share secret keys via some secure channel is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. Some examples of public-key cryptosystems are Elgamal (named for its inventor, Taher Elgamal), RSA (named for its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman), Diffie-Hellman (named, you guessed it, for its inventors), and DSA, the Digital Signature Algorithm (invented by David Kravitz).

How PGP Works PGP combines some of the best features of both conventional and public key cryptography. PGP is a *hybrid cryptosystem.* When a user encrypts plaintext with PGP, PGP first compresses the plaintext. Data compression saves modem transmission time and disk space and, more importantly, strengthens cryptographic security. Most cryptanalysis techniques exploit patterns found in the plaintext to crack the cipher. Compression reduces these patterns in the plaintext, thereby greatly enhancing resistance to

cryptanalysis. (Files that are too short to compress or which don't compress well aren't compressed.)

PGP then creates a *session key,* which is a one-time-only secret key. This key is a random number generated from the random movements of your mouse and the keystrokes you type. This session key works with a very secure, fast conventional encryption algorithm to encrypt the plaintext; the result is ciphertext. Once the data is encrypted, the session key is then encrypted to the recipient's public key. This public key-encrypted session key is transmitted along with the ciphertext to the recipient.

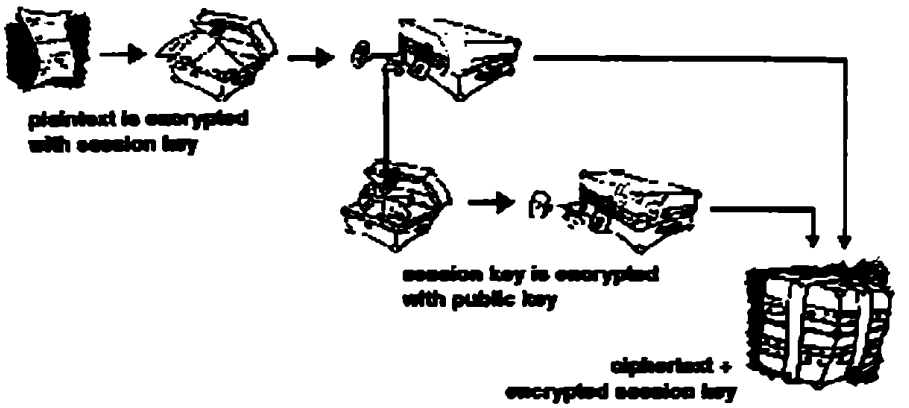


FIGURE 14.4. *How PGP encryption works*

Decryption works in the reverse. The recipient's copy of PGP uses his or her private key to recover the temporary session key, which PGP then uses to decrypt the conventionally-encrypted ciphertext.

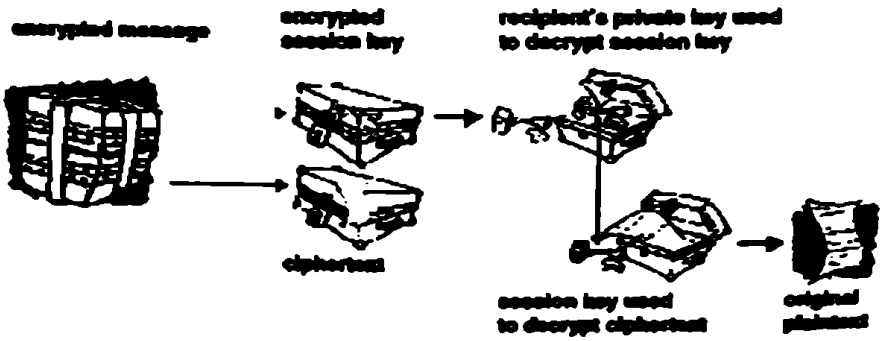


FIGURE 14.5. *How PGP decryption works*

Keys A key is a value that works with a cryptographic algorithm to produce a specific ciphertext. Keys are basically really, really, really big numbers. Key size is measured in bits; the number representing a 1024-bit key is darn huge. In public key cryptography, the bigger the key, the more secure the ciphertext.

Keys are stored in encrypted form. PGP stores the keys in two files on your hard disk; one for public keys and one for private keys. These files are called *keyrings*. As you use PGP, you will typically add the public keys of your recipients to your public keyring. Your private keys are stored on your private keyring. If you lose your private keyring, you will be unable to decrypt any information encrypted to keys on that ring.

Digital Signatures A major benefit of public key cryptography is that it provides a method for employing *digital signatures*. A digital signature serves the same purpose as a handwritten signature. However, a digital signature is superior to a handwritten signature in that it is nearly impossible to counterfeit, plus it attests to the contents of the information as well as to the identity of the signer.

The basic manner in which digital signatures are created is illustrated in *Figure 14.6*. Instead of encrypting information using someone else's public key, you encrypt it with your private key. If the information can be decrypted with your public key, then it must have originated with you.

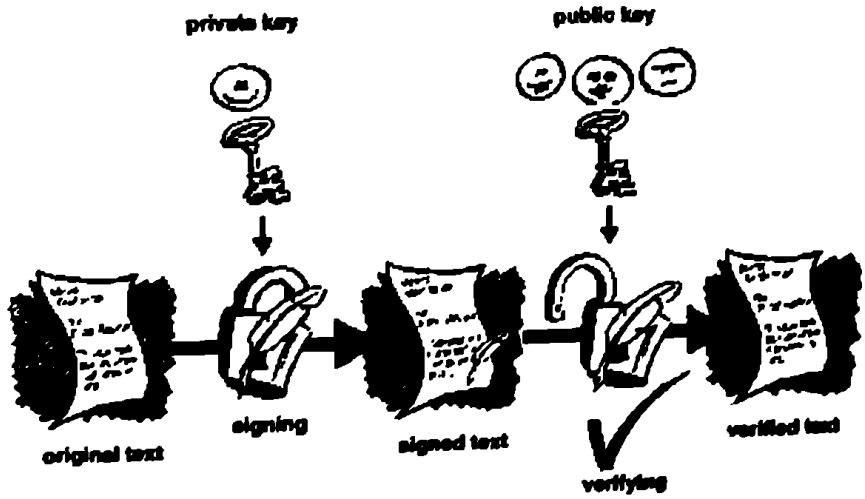


FIGURE 14.6. *Simple digital signatures*

C. Review Questions

Exercise 1. Gap Filling

encryption	cryptanalysis	plaintext
cryptosystem	cryptography	decryption
key	ciphertext	cipher

The term [1]"_____" ("secret writing," from the Greek *kryptós*, "hidden," and *gráphein*, "to write") is often used to refer to the field as a whole, as is "cryptology" ("the study of secret writing"). The study of how to circumvent the use of cryptography is called [2]"_____" or, loosely, "codebreaking."

Classically, "cryptography" referred almost exclusively to [3]"_____", the process of converting ordinary information ([4]"_____") into an unreadable [5]"_____". [6]"_____" is the reverse process, recovering the plaintext back from the ciphertext.

A [7]"_____" is a set of algorithms for encryption and decryption. The exact operation of a cipher is normally controlled by a [8]_____ – a secret parameter for the cipher algorithms. Historically, ciphers were often used directly for encryption or decryption, but in modern techniques, a cipher is only one part of a [9]"_____", a set of algorithms, protocols, and operating procedures for encryption and decryption that may use the cipher.

Exercise 2. Open Ended

Explain the difference between the two terms:

1. code and cipher
2. plaintext and ciphertext
3. cipher and decipher
4. cryptography and cryptology
5. public-key and private-key
6. handwritten signature and digital signature

Exercise 3. Language Work: Conditional Clauses

GRAMMAR NOTE	
I Conditional *(possible situation)*	
Future	If A happens ⇒ B will happen **V(s) shall / will + V** *If you **upgrade** your PC, you **will be** able to run this game.*

II Conditional *(unlikely situation)*	
Future/ Present	If A happened ⇒ B would happen **V-ed would + V** *If you **upgraded** your PC, you **would be** able to run this game.*
Past	If A had happened ⇒ B would have happened **had + V-en would have + V-en** *If you **had upgraded** your PC, you **would have been** able to run this game.*

a) Answer the following questions using II Conditional:

- What could happen if computers suddenly stopped working? For example, would public health and safety be disrupted and lives be endangered if computers went down?
- What would happen if a hacker altered the chemical formulas for prescription drugs, or the flight patterns and other data in air traffic control computers?
- If you thought you were now living in computer simulation created by some advanced civilization, how should it change the way you live your life?
- If you could have a free chip put in your brain so that you would automatically gain some supernatural power, which supernatural power would it be and why?
- What super computer would you buy if you were to spend 5000 dollars (4000 euros)?

- Over the next decade, say some experts, electronic miniaturization will produce small, portable devices that you can wear like clothing. These devices would do all kinds of wonderful things. For example, they will incorporate display screens, keyboards, CD memories, faxes, telephones, scanners, cameras, and satellite transmitters and will recognize handwriting and voice. What kind of wearable, lightweight "information and entertainment machine" would you design for yourself? What would it do?

☺ ☺ ☺

If my mom designed a computer...

If my mom designed a computer, it would say please and thank you. It would know when I was really focused on something and wouldn't bug me with e-mail, IM and other notifications. It would know just when to call me back from that Internet playground, so I could be sure and finish my homework and chores, ah, work for the day. It would wake up before I do and be ready for the day and have my day organized when I logged on. When the applications began to squabble about who was most important, it would patiently remind everyone that there's only one of me and they'd better just work out their differences on their own. Though my mom's well into her senior years, she's still bright and spry so if anybody in design land wants to hire her, let me know 'cause I want to buy the computer environment she'd design.

☺ ☺ ☺

b) Match the two halves from the lists A and B to make eight meaningful sentences.

A

1. Consider the polynomial $(x-a)(x-b)= x^2 -(a+b)x+ab$. If $(a+b)$ and ab were both algebraic, ...
2. It would be cool ...
3. If the project manager fails to build the team, ...
4. If you turned your computer off, ...
5. If a math giant from the past – someone like Gauss – came back to Earth, ...
6. Theoretical computer science would exist ...
7. If you forget your login and password, ...
8. If modern computers were powerful enough to run the computational processes that take place in human brain, ...
9. If Einstein had been born into a primitive tribe which was unable to count beyond three, ...
10. If your computer breaks down after the warranty expires, ...

B

a) ... you will not be able to enter the system.

b) ... we wouldn't know how to program them to do it.

c) ... life-long application to mathematics probably would not have carried him beyond the development of a decimal system based on fingers and toes.

d) ... even if there were no computers.

e) ... you would decrease the risk of someone accessing your files or e-mail.

f) ... I will try to fix it.

g) ... then this would be a polynomial with algebraic coefficients.

h) ... then the project will suffer.

i) ... he would have a lot of catching up to do but he would find that today math is done much the same way that it was done during his life.

j) ...if you dropped me an email.

D. Role Play

How easy is it to crack a simple cipher? This task lets you find out. Below you can see a secret message encrypted using a cipher. The cipher's key is straightforward – each letter of the alphabet is represented by another letter. Are you ready to decipher? Then let's get to work!

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Bldlhgd, ygwg vwg clqw lwrgwm:

U'og wgbguogr flwr jyvj jyg zlcm vwg uh rgmxgwvjg hggr li mqxxdugm vhr jyvj nlwvdg um dlf. Jyum um rqg, hl rlqzj, jl clqw vxxvddduht uhblnxgjghbg. U vn mqwg clq bvh iuhr vdd jyg mqxxdugm clq hggr uh jyg mqxxdc yqj, vm uj yvm zggh wgbghjdc mjlbsgr. Vm ivw vm nlwvdg tlgm, qmg clqw ygvr! Rlh'j xqmy jyg zlcm jll yvwr (zqj rlh'j dgj jygn ilwtgj fyl um uh blnnvhr!) Vhr jygh jygwg'm vdfvcm muhtuht... vhr xdvcuht tvngm.

Nvsg mqwg jyg ylqmg um bdgvhgr vhr jurc fygh U wgjqwh ylng jlnlwwlf. Vhr rl hlj dgj ng iuhr jyg zlcm uh fgj ruvxgwm.

Clqw dlouht fuig,

Nvjudrv

FRACTALS

Vocabulary List

antenna, *n*
centimeter, *n*
complex constant
complex polynomial
computer graphics
coordinate, *n*, *adj*
dimension, *n*
equilateral triangle
finite, *adj*
form, *n*
formula, *n*
fractal, *n*
fractal geometry
Hausdorff dimension
IFS, *abbr*
imaginary number
infinite, *n*, *adj*
integer, *n*
Koch curve
length, *n*
line, *n*, *v*, *adj*
Mandelbrot set
measure, *n*, *v*
non-integer, *n*
plug, *v*
ratio, *n*
real number
recursive, *adj*
segment, *n*, *v*
self-similarity, *n*
Serpinski Carpet
Serpinski Triangle
size, *n*
square root
zoom in
zoom out

A. Pre-reading

- ***In the past, most people believed that the geometry of nature was strictly based on simple objects such as straight lines, circles, and polygons. For example, the sun and the planets all appear to be spherical, and the planets revolve around the sun in an elliptical path. But many objects in nature are so complicated and irregular that we would not be able to model them with simple geometric shapes. For instance, what geometric shape could be used to model a tree, a cloud or a mountain range?***

B. Reading

„To see a World in a Grain of Sand...
And a Heaven in a Wild Flower
Hold Infinity in the Palm of your hand
And Eternity in an hour."
William Blake

What Are Fractals?

by Jim Tucek

Fractals have come up as an important question two times before the invention of computers. The first time was when British map makers discovered the problem with measuring the length of Britain's coast. On a zoomed out map, the coastline was measured to be 5,000 something or other. By measuring the coast on more zoomed in maps, it got to be longer, like 8,000. And by looking at really detailed maps, the coastline was over double the original. You see, the coastline of Britain that's on a map of the world doesn't have all the bays and harbors. A map of just Britain has more of these, but not all the little coves and sounds. The closer they looked, the more detailed and longer the coastline got. Little did they know that this is a property of fractals. (A finite area, e.g., Britain, being bounded by an infinite line).

The second instance of pre-computer fractals was noted by the French mathematician Gaston Julia. He wondered what a complex polynomial function would look like, such as the ones named after him (in the form of $z^2 + c$, where c is a complex constant with real and imaginary numbers). The idea behind the formula is that you take the x and y coordinates of a point, and plug them into z in the form of $x + y*i$, where i is the square root of negative one, square this number, and then add c, a constant. Then plug the resulting pair of real and imaginary numbers back into z, run the equation again, and keep doing that until the result is greater than some number. The number of times you have to run the equations to get out of its "orbit" can be assigned a color and then the pixel (x, y) gets turned that color, unless

those coordinates can't get out of their orbit, in which case they're made black.

Later, Benoit Mandelbrot, an employee of IBM, thought about writing a program with a formula such as $z_{n+1} = z_n^2 + c$, and then running it on one of IBM's many computers. And they eventually got some pretty pictures. Mandelbrot was the first person to get computers do the many repetitive calculations to make a fractal look good. And now you know the mathematical aspects of fractals.

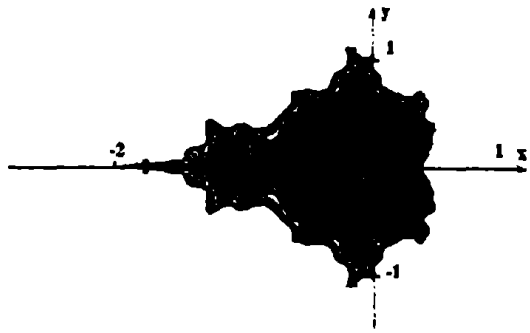


FIGURE 15.1 *The Mandelbrot set*

The basic concept of fractals is that they contain a large degree of self-similarity. This means that they usually contain little copies of themselves buried deep within the original. And they also have infinite detail. Like the costal problem, the more you zoom in on a fractal, the more detail (coastline) you get. And this keeps going on forever and ever, so you could make a pretty movie of a fractal zooming in.

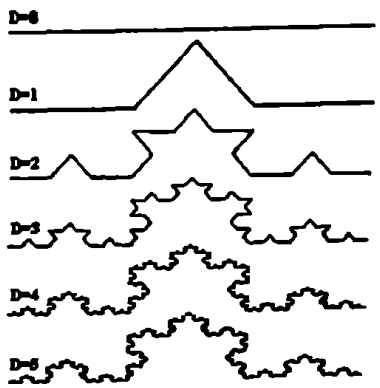


FIGURE 15.2 *The Koch curve constructed with IFS*

One of the unique things about fractals is that they have non-integer dimensions. That is, while you are in the 3rd dimension, looking at this on a flat screen which can be considered more or less the 2nd dimension, fractals are in between the dimensions. Fractals can have a dimension of 1.8, or 4.12. Although fractals may not be in integer dimensions, they always have a smaller dimension than what they're on. If you make a fractal by drawing lines that obey a certain rule, like Koch Curve, that fractal can't have a dimension higher than the paper it's drawn on, which would be 2.

And how exactly does one calculate how many dimensions a fractal has? Well, it can be assumed that for any fractal object (of size P, made up of smaller units of size p), the number of units (N) that fits into the larger object is equal to the size ratio (P/p) raised to the power of d, which is called the Hausdorff dimension.

$$N = \left(\frac{P}{p}\right)^d \qquad \text{or} \qquad d = \frac{\log N}{\log(P/p)}$$

Let's try this for Koch curve. Using only line segments that are 3 centimeters long (P), you make a simple Koch Curve, which is just a Star of David. 12 segments, 3 centimeters per segment. If you take that to the next level and use line segments which are 1 centimeter long (p), you use 48 line segments. By cutting the length of the line segments by one third ($P = 3$, $p = 1$, $P/p = 3$), the number of line segments used (N) goes up four times (48 segments for p divided by 12 segments for P equals 4). That means $N = 4$, $P/p = 3$, so $d = \log 4 / \log 3$. Using a little help from a calculator, we find that Koch Curve has a dimension of 1.2618595071429. Amazing but true.

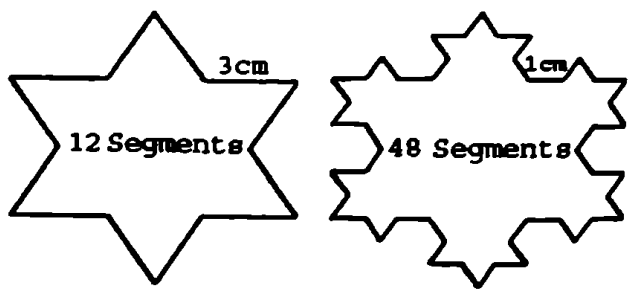


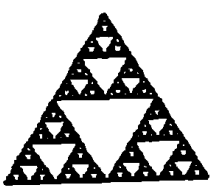
FIGURE 15.3 *A simple Koch curve*

What good are mathematical pictures that aren't even whole dimensions? Well, they're pretty. As mentioned before, nature is full of fractal-like stuff. Twigs on trees look like the branches which they grow on, which look like the tree itself. It's the same thing with fern leaves, and so many other living things. Remember that artist who made paintings by splashing and dribbling paint onto a canvas? Even though it looks like a mess, his paintings, especially the later ones, look good. The reason his paintings "look good" is that their fractal dimension is close to that of nature's, especially in the later paintings.

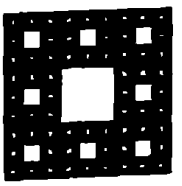
Anyway, self-similarity is part of this world, so fractals can make pretty good copies of it. Artists have created very realistic looking landscapes composed of a few fractal equations. Using FractInt, I've made not-so-bad looking mountains and even a moon, which looks more like one of the moons of Jupiter, but a moon none the less.

FIGURE 15.4 *Landscapes constructed with FractInt*

Fractals also have technological applications. Motorola has started using fractal antennas in many of its cellular phones, and reports that they're 25% more efficient than the traditional piece of wire. They're also cheaper to manufacture, can operate on multiple bands, and can be put into the body of the phone. The journal *Fractals* showed why fractals work so well as antenna. For an antenna to work equally well at all frequencies, it must be symmetrical around a point and it must be self-similar, both of which fractals can provide.



Serpinsky's Triangle



Serpinsky's Carpet

FIGURE 15.5 *Good Fractal Antennas*

C. Review Questions

Exercise 1. Matching

Use the words from the two lists: A and B to make nine expressions connected with fractals. Then use the expressions to complete the sentences.

A	B
fractal	function
Koch	graphics
geometric	dimension
computer	number
Sierpinski	Curve
complex	shape
polynomial	algorithm
Hausdorff	geometry
recursive	Triangle

1. In algebra, *z* is a symbol standing in for a ..... unknown.
2. Setting a polynomial equal to zero results in a polynomial equation; equating it to a variable results in a .....
3. A fractal is usually a rough ....
4. Fractals are also said to have a fractal dimension or .....
5. Here is a fractal with dimension 1.26, it's the .....
6. In mathematics, ... is the study of complex shapes with the property of self-similarity, known as fractals.
7. At present the role of fractals in ..... is rather high.
8. Trees and ferns are fractal in nature and can be naturally modeled on a computer using a .....
9. ..... is a fractal image formed from applying a fractal algorithm to Pascal's Triangle.

Exercise 2. Word-building

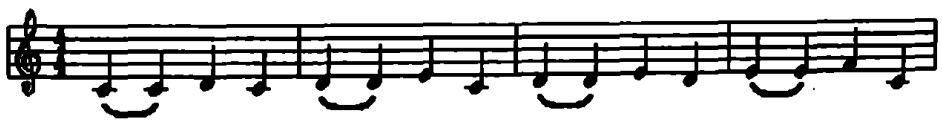
Use the word in capitals at the end of each line to form a word that fits in the space in the same line (be prepared to make grammatical changes). There is an example at the beginning (0).

What is Fractal Music?

Fractal music is a kind of music derived from fractal [0] *algorithms*. Instead of iterating an equation and applying the results to color a [1]_____ (as in fractal [2]_____), they are applied to audio parameters. The [3]_____ of fractals in music may seem a bit odd, but music is actually highly [4]_____.

Since [5]_____ it's known that tonal harmony is closely related to the [6]_____ relation of the frequencies. Modern multimedia PCs [7]_____ and [8]_____ music mathematically, as do CDs and [9]_____ stereo TVs. It was only natural then that the twentieth century [10]_____ of fractal be applied to [11]_____.

ALGORITHMIC
PIXELIZATION
GRAPHICAL
APPLY
MATHEMATICS
PYTHAGOREAN
NUMERAL
ENCODER
DECODED
DIGITIZE
CONCEPTUAL
MUSICIAN



Exercise 3. Open Ended

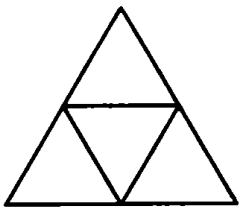
1. Who is the founder of fractal geometry and the discoverer of the Mandelbrot set?
2. What is a simple answer to the question *What are Fractals*?
3. What are some examples of fractals?
4. Are there natural fractal structures? If so, give some examples.
5. What does the concept of self-similarity mean?
6. How can a fractal be generated?
7. What is fractal dimension? How is it calculated?
8. How does fractal geometry differ from Euclidean geometry?
9. What practical applications does fractal theory have?

D. Making a Fractal

*Let's make a famous fractal called the **Sierpinski Triangle.***

Step One

Draw an equilateral triangle with sides of 2 triangle lengths each. Connect the midpoints of each side.



How many equilateral triangles do you now have?
Shade out the triangle in the center. Think of this as cutting a hole in the triangle.

Step Two

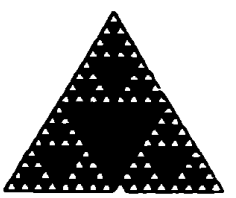
Draw another equilateral triangle with sides of 4 triangle lengths each. Connect the midpoints of the sides and shade the triangle in the center as before. Notice the three small triangles that also need to be shaded out in each of the three triangles on each corner – three more holes.

Step Three

Draw an equilateral triangle with sides of 8 triangle lengths each. Follow the same procedure as before, making sure to follow the shading pattern. You will have 1 large, 3 medium, and 9 small triangles shaded.

Step Four

Follow the above pattern and complete the Sierpinski Triangle. Use your artistic creativity and shade the triangles in interesting color patterns. Does your figure look like this one? Then you are correct!



Check Your Progress (5)
Units 13 – 15

Fill in the blanks in the sentences below using information from units 13 – 15.

1. The branch of algebra that concerns itself with the study of remainder is known as _____ _____.
2. Congruences can be depicted geometrically by placing numbers on a circle, which is known as the _____ _____.
3. Linear congruential sequences are often applied to generate a set of _____ numbers.
4. Data that can be read and understood without any special measures is called _____ or *cleartext.*
5. The method of disguising plaintext in such a way as to hide its substance is called _____.
6. Encrypting plaintext results in unreadable gibberish called _____.
7. The process of reverting ciphertext to its original plaintext is called _____.
8. While _____ is the science of securing data, _____ is the science of analyzing and breaking secure communication.
9. A *cryptographic algorithm*, or _____, is a mathematical function used in the encryption and decryption process.
10. The basic concept of fractals is that they contain a large degree of _____. This means that they usually contain little copies of themselves buried deep within the original.
11. One of the unique things about fractals is that they have _____ dimensions.
12. Motorola has started using fractal _____ in many of its cellular phones

Glossary of Notation

- У Великобританії та США розряди чисел відокремлюють один від одного комою, а ціле число від дробового у десяткових дробах – крапкою.
- Нуль як одну із цифр числа вимовляють *o* [əu], можна також вимовляти як *nought* [nɔ:t] або *zero* [ˌziərəu]; у ролі самостійного числа нуль вимовляють *zero* [ˌziərəu].
- Усі літери латинського алфавіту читають відповідно до їхніх англійських назв.

+	plus
−	minus
±	plus or minus
× ·	multiplied by
÷ : /	divided by
( )	round brackets; parentheses
{ }	curly brackets; braces
[ ]	square brackets; brackets
	and so on to
≡	(is) identical with; (is) always equal to; (is) congruent to
~	equivalent, similar; of the order of; proportional to
≅	(is) approximately equal to; approximately equals
∞	infinity

$x \to \infty$	x tends to infinity; x approaches infinity
$\propto$	varies directly as; (is) (directly) proportional to
$-a$	negative a
$!\ (n!)$	n factorial
$\hat{\varphi}$	phi hat
a'	a prime
a''	a double prime
a''_2	a second, double prime; a double prime, second
a'''	a triple prime
f'_c	f prime sub (suffix) c; f suffix (sub) c, prime
$\vec{a}$	a vector; the mean value of a
$\dot{a}$	the first derivative
$\ddot{a}$	the second derivative
$\dddot{a}$	the third derivative
a_1	a first; a sub one; a suffix one
a_n	a n-th; a sub n; a suffix n
a_k^2	a sub k sup 2; a sup 2 sub k

$10''$	ten seconds; ten inches
$90°$	ninety degrees
$87°6'10''$	eighty seven degrees six minutes ten seconds
$\sin 30.2°$	the sine of thirty point two degrees
$\tan\theta = \frac{\sin\theta}{\cos\theta}$	the tangent of theta equals the sine of theta over the cosine of theta
$\cos A = \frac{\tan b}{\cot c}$	the cosine of capital A is equal to the tangent of b divided by the cotangent of c
$\tan\theta = \frac{\sec\theta}{\csc\theta}$	tangent theta equals secant theta over cosecant theta
$a = b$	a is b; a equals b; a is equal to b
$a \neq b$	a is not b; a does not equal b; a is not equal to b
$a \approx b$	a approximately equals b
$a \pm b$	a plus or minus b
$a > b$	a is greater than b
$a >> b$	a is much greater than b
$a < b$	a is less than b
$a << b$	a is much less than b
$a \geq b$	a is greater than or equals b
$a \leq b$	a is less than or equals b
$a_2 > a_d$	a second is greater than a d-th

$\lvert a \rvert$	the modulus of a; the absolute value of a
$a + b = c$	a plus b is c; a plus b equals c; a plus b is equal to c; a plus b makes c
$4 + 7 = 11$	four plus seven is eleven; four plus seven equals eleven; four plus seven is equal to eleven
$12 > 5 + 5$	twelve is greater than five plus five
$5 + 5 < 12$	five plus five is less than twelve
$y = \sum_{k=0}^{4} a_k x^k$	y equals the sum from k equal to zero to k equal to four of a sub k, x to the power of k
$c - b = a$	c minus b is a; c minus b equals a; c minus b is equal to a; c minus b leaves a
$(2x - y)$	bracket two x minus y close the bracket
$18 - 6 = 12$	eighteen minus six is equal to twelve; eighteen minus six equals twelve; eighteen minus six is twelve; eighteen minus six leaves twelve
$1 \times 1 = 1$	once one is one
$2 \times 2 = 4$	twice two is four
$5 \times 5 = 25$	five times five is twenty five; five multiplied by five equals twenty five; five by five is equal to twenty five; five times five makes twenty five
$\prod_{i=1}^{n} 1 = 1$	the product from i equal to one to n of one equals one

$A \times B$	the Cartesian product of A and B
$S = v \cdot t$	distance = velocity × time; S equals v by t; S is equal to v multiplied by t; S equals v times t, where S means distance, v means velocity, t means time
$A = F \cdot S$	work = force ×distance; work is equal to the product of the force times distance; A equals F multiplied by S where A means work, F means force and S means distance
16 : 4 = 4	sixteen divided by four is four; sixteen by four equals four; sixteen by four is equal to four; the ratio of sixteen to four is four
20 : 5 = 16 : 4 $\frac{20}{5} = \frac{16}{4}$	the ratio of twenty to five equals (is equal to) the ratio of sixteen to four
1 : 2	the ratio of one to two
2 : 3 = 4 : 6	two to three is as four to six
1/2	a (one) half
1/3	a (one) third
1/4	a (one) quarter; a (one) fourth
1/8	one eighth
2/3	two thirds
3/4	three quarters; three fourths
5/6	five sixths
25/57	twenty-five fifty-sevenths

1/273	one two hundred and seventy third
2 ½	two and a half
3 ¾	three and three quarters
1.1	one point one
2.12	two point one two
15.505	fifteen point five o [ou] five
0.5 .5	o [ou] point five; zero point five; nought point five; point five; one half
0.002 .002	o [ou] point o [ou] o [ou] two; zero point zero zero two; point two oes[ouz] two; point two noughts two
0.0000001 .0000001	o [ou] point six noughts one
sin 30.2°= .5030	the sine of thirty point two degrees equals zero point five, zero, three, zero
12%	twelve percent
87	eighty-seven
101	one hundred (and) one
211	two hundred (and) eleven
1,024	one thousand (and) twenty-four
3,728	three thousand seven hundred (and) twenty-eight
100,000	one hundred thousand
1,048,576	one million forty-eight thousand five hundred (and) seventy-six
1,000,000,000	one billion

a^n	a to the n-th power;
$a\hat{}n$	a to the power of n;
	the n-th power of a;
	a raised to the n-th power
x^2	x square;
$x\hat{}2$	x squared;
	the square of x;
	the second power of x;
	x to the second power;
	x raised to the second power
$4^2 = 16$	four squared is sixteen;
	the square of four is sixteen;
	the second power of four is sixteen
$(a+b)^2$	a plus b all squared
y^3	y cube;
$y\hat{}3$	y cubed;
	the cube of y;
	the third power of y;
	y raised to the third power;
	y to the third power
$3^3 = 27$	three cube is twenty seven;
	the cube of three is twenty seven
a^5	a to the fifth power;
	a raised to the fifth power
y^{-10}	y to the minus tenth power
$\sqrt{16} = 4$	the square root of sixteen is four
$\sqrt{a}$	the square root of a
$\sqrt[3]{a}$	the cube root of a
$\sqrt[3]{27} = 3$	the cube root of twenty seven is three
$\sqrt[4]{16} = 2$	the fourth root of sixteen is two

$\sqrt[5]{a^2}$	the fifth root of a square
$\alpha = \sqrt{R^2 + x^2}$	alpha equals the square root of capital R square plus x square
$\sqrt{\frac{7_1 + A}{2xa''}}$	the square root of seven first plus capital A divided by two xa double prime
$\frac{x \pm \sqrt{x^2 - y^2}}{y}$	x plus or minus the square root of x square minus y square all over y
$a^{\frac{m}{n}} = \sqrt[n]{a^m}$	a to the m by n-th power equals the n-th root of (out of) a to the m-th power
$\frac{a+b}{a-b} = \frac{c+d}{c-d}$	a plus b over a minus b is equal to c plus d over c minus d
$a = \frac{e}{l}$	a is equal to the ratio of e to l
$\frac{ab^2}{b} = ab$	ab square (divided) by b equals ab
$\frac{a}{\infty} = 0$	a divided by infinity is infinitely small; a by infinity is equal to zero
$L = \sqrt{R^2 \pm x^2}$	capital L equals the square root out of capital R square plus minus x square
$E = \frac{\frac{P}{a}}{\frac{e}{l}} = \frac{Pl}{al}$	capital E is equal to the ratio of capital P divided by a to e divided by l is equal to the ratio of the product of capital P and l to the product of al
$\gamma = \frac{c'c}{ac'}$	gamma is equal to the ratio of the segment c prime c to the segment ac prime
$\frac{dz}{dx}$	dz over dx; the first derivative of z with respect to x
$\frac{d^2y}{dx^2}$	the second derivative of y with respect to x square; d two y over $d\,x$ square

$\frac{d^n y}{dx^n}$, $D^n x^y$	the n-th derivative of y with respect to x
$\int_n^m$	the integral from n to m; the integral between the limits n and m
$\frac{d}{dx}\int_{x_0}^{x} X dx$	d over dx of the integral from x nought to x of capital X dx
$\int \frac{dy}{\sqrt{c^2 - y^2}}$	the integral of dy divided by the square root out of c square minus y square
$\log_2 x = 2$	the logarithm of x to the base two equals two
$H[D]$	set of functions holomorphic in D (function spaces)
$\|f\|$	norm of f (function spaces)
$y = f(x)$	y is the value of the function corresponding to x; y is a function of x; y equals f of x
f^{-1}	the inverse function of the function f
$\|v\|$	the norm of v (vectors)
v^2	the norm square of v (vectors)
$u =: v$	u is defined as v (vectors)
u dot v	the vector dot product of u and v (vectors)
u cross v	means the vector cross product of u and v (vectors)
v/r	means the scalar vector quotent of v and r (vectors)
rv	means the scalar vector product of r and v (vectors)
$d(S1, S2)$	distance between the sets $S1$and $S2$ (curves, domains, regions)
$x(z1, z2)$	chordal distance of $z1$ and $z2$ (curves, domains, regions)

$x(z1, z2)$	Euclidean distance of z1 and z2 (curves, domains, regions)
C is a "scroc"	C is a simple closed rectifiable oriented curve
$b = I(a + bi)$	b is the imaginary part of a plus bi (complex variables)
$a = R(a + bi)$	a is the real part of a plus bi (complex variables)
$F = C_{\mu}\ HIL \sin\theta$	capital F equals capital C sub (suffix) mu HIL sine theta
$P_{cr} = \frac{\pi^2 El}{4l^2}$	capital P sub (suffix) cr (critical) equals pi square capital E by l all over four l square
$f: A \to B$	f is a function under which each element of set A has an image in set B
$Int\ (s)$	the interior of S (set theory)
$C(S)$	the complement of S
S'	the derived set of a given set S
$\overline{S}$	closure of the set S
$C \cup D$	the union of sets C and D; C unions D; C cup D
$C \cap D$	the intersection of sets C and D; C intersects D; C cap D
$a \in A$	small a is an element of the set capital A; a belongs to A
$A \subset B$	A is a proper subset of B
$A \not\subset B$	A is not a subset of B
$B \supset A$	B is a proper superset of A

$A \subseteq B$	A is a subset of B; A is included in B
$B \supseteq A$	B is a superset of A; B includes A
$M = \{2,4,6\}$	M is the set with the elements 2, 4, 6
$M = \varnothing$	M is an empty set; M is a null set
$P(A)$	probability of the event A
$P(A \mid B)$	probability of the event A conditional on the event B
$\lim_{x \to x_1} f(x) = L$	f of x approaches the limit L as x tends to the value x first in any way
$V - u\sqrt{\sin^2 i - \cos^2 i} = u$	V equals u the square root of sine square I minus cosine square I equals u
$K = \max \sum_{i=1}^{n} \lvert a_{ij}(t) \rvert (t \in [a,b]);$ $j = 1,2..n)$	K is equal to the maximum over j of the sum from I equals one to I equals n of the modulus of a sub ij of t, where t lies in the closed interval ab and where j runs from one to n
$u = \int f_1(x)\,dx + \int f_2(y)\,dy$	u is equal to the integral of f sub one of x multiplied by dx plus the integral of f sub two of y multiplied by dy
$(D - r_1)[(D - r_2)y] =$ $= (D - r_2)[(D - r_1)y]$	open round brackets capital D minus r first close the round brackets open square and round brackets capital D minus r second close round brackets by y close square brackets equals open round brackets capital D minus r second close round brackets open square and round brackets capital D minus r first close round brackets by y close square brackets

$$\left[(x+a)^p - \sqrt[r]{x}\right]^{-q} - s = 0$$

x plus a in round brackets to the power of p minus the r-th root of x (all in square brackets) to the minus q-th power minus s equals nothing (zero)

$$M = R_1 x - P_1(x - a_1) - P_2(x - a_2)$$

capital M is equal to capital R sub one multiplied by x minus capital P sub one round brackets opened x minus a sub one brackets closed minus capital P sub two round brackets opened x minus a sub two brackets closed

$$a_\upsilon = \frac{m\omega\omega^2\alpha^2}{\left[rp^2m^2 + R_2\left(R_1 + \dfrac{\omega^2\alpha^2}{rp}\right)\right]}$$

a sub v is equal to m omega, omega square alpha square divided by square brackets rp square m square plus capital R second round brackets opened capital R first plus omega square alpha square divided by rp round and square brackets closed

$$D'_{n-1}(x) = \prod_{s=0}^{n}\left(1 - x_s^2\right)^{\varepsilon-1}$$

D sub n minus one prime of x is equal to the product from s equal to zero to n of, parenthesis, one minus x sub s squared, close parenthesis, to the power of epsilon minus one

$$K(t,x) = \frac{1}{2\pi i}\int_{\left|\omega-\frac{1}{2}\right|=\rho}\frac{K(t,z)}{\omega - \omega(x)}\,d\omega$$

K of t and x is equal to one over two pi i, times the integral of K of t and z, over omega minus omega of x, with respect to omega along curve of the modulus of omega minus one half, is equal to rho

English-Ukrainian Dictionary of Mathematics & Computer Science

Умовні скорочення, прийняті у словнику

abbr - abbreviation – абревіатура
adj - adjective - прикметник
adv – adverb – прислівник
conj – conjunction – сполучник
n – noun – іменник
pl – plural – множина
p. p. – past participle – дієприкметник минулого часу
prep – preposition – прийменник
v – verb – дієслово

A

abstract ['æbstrækt] **1.** *n* абстракція, абстрактне поняття; **2.** *adj* абстрактний; ~ **set** абстрактна множина.

access ['ækses] **1.** *n* доступ; ~ **control** контроль за доступом, керування доступом; **2.** *v* мати доступ, отримати доступ [до чогось].

account [ə'kaunt] **1.** *n* 1) абонемент; 2) рахунок; розрахунок; 3) звіт; звітна доповідь; бюджет [ресурсів]; ~ **name** звітне ім'я; реєстраційне ім'я.

action ['ækʃ(ə)n] *n* екшин; бойовик.

Ada King ['eidə kiŋ] *n* Ада Кінг.

add [æd] *v* додавати, підсумовувати.

addition [ə'diʃ(ə)n] *n* додавання.

address [ə'dres] **1.** *n* адреса; **IP** ~ IP-адреса; **2.** *v* адресувати.

administrator [əd'ministreitə] *n* адміністратор; **system** ~ системний адміністратор [мережі].

adventure [əd'ventʃə] *n* пригодницька гра.

affine ['æfain] *adj* афінний.

algebra ['ældʒibrə] *n* алгебра; **linear** ~ лінійна алгебра; **matrix** ~ матрична алгебра.

algebraic [ˌældʒi'breiik] *adj* алгебричний; ~ **sign** алгебричний знак; ~ **symbol** алгебричний символ.

algorithm [ˈælgərɪðəm] *n* алгоритм; **cryptographic ~** криптографічний алгоритм.

al-Khwarizmi [ˌelkwəˈrɪzmɪ] *n* Аль-Хорезмі.

analogy [əˈnælədʒɪ] *n* аналогія.

analysis [əˈnæləsɪs] *n* аналіз; **discrete ~** дискретний аналіз; **functional ~** функціональний аналіз; **mathematical ~** математичний аналіз.

analyze [ˈænəlaɪz] *v* (=analyse) аналізувати.

android [ˈændrɔɪd] *n* андроїд, людиноподібний робот.

antenna [ænˈtenə] *n* антена.

application [ˌæplɪˈkeɪʃən] *n* 1) прикладна програма; 2) застосування, використання.

applied [əˈplaɪd] *adj* практичний, прикладний; **~ mathematics** прикладна математика.

apply [əˈplaɪ] *v* 1) прикладати; 2) застосовувати; вживати.

arbitrary [ˈɑ:bɪtrərɪ] *adj* 1) довільний; випадковий; **~ constant** довільна стала.

arithmetic **1.** *n* [əˈrɪθmətɪk] арифметика; лічба; **modulo (or modular) ~** модульна арифметика; арифметичні операції над абсолютними значеннями чисел; **2.** *adj* [ˌærɪθˈmetɪk] (= arithmetical) арифметичний.

array [əˈreɪ] *n* 1) структура; масив; **rectangular ~** прямокутна [геометрична] структура; прямокутна [графічна] структура; прямокутний [геометричний] масив; прямокутний [графічний] масив; [геометрична] структура у прямокутних координатах, [геометричний] масив у прямокутних координатах; 2) матриця; ґратка; сітка; 3) таблиця.

artificial [ˌɑ:tɪˈfɪʃ(ə)l] *adj* штучний; **~ intelligence** штучний інтелект.

assign [əˈsaɪn] *v* призначати, присвоювати.

assume [əˈsju:m] *v* вважати, припускати.

attachment [əˈtætʃmənt] *n* прикріплення, приєднання, вкладення, додаток [до листа].

attacker [əˈtækə] *n* 1) нападник, порушник, супротивник; 2) криптоаналітик.

authenticate [ɔ:'θentikeit] *v* засвідчувати, встановлювати справжність (автентичність).

authentication [ɔ:,θenti'keiʃ(ə)n] *n* автентифікація, перевірка (підтвердження) справжності.

axis ['æksis] *n* (*pl* axes) вісь.

B

Babbage ['bæbidʒ] *n* Бебідж.

Banach ['ba:nək] *n* Банах; ~ **space** банаховий простір.

biometric [,baiə'metrik] *adj* біометричний; ~ **access (control system)** біометрична система контролю доступу, біометричний доступ; ~ **data** біометричні дані; ~ **passport** біометричний паспорт.

biometric(s) [,baiə'metrik(s)] *n* (= biometry) 1) біометрія; 2) біометричні характеристики [у системах розпізнавання].

black box [,blæk 'bɔks] *n* "чорна скринька".

boldface [,bould'feis] *n* жирне накреслення букв.

boundary ['baund(ə)ri] *n* границя, межа; край.

bracket ['brækit] **1.** *n* дужка; **2.** *v* брати у дужки.

browser ['brauzə] *n* браузер, програма перегляду Web, навігатор.

bug [bʌg] *n* помилка [у програмі чи пристрої], дефект.

C

C++ [,si: plʌs 'plʌs] *n* мова C++ (Сі плюс плюс).

C# [,si: 'ʃa:p] *n* мова C#; **program in** ~ програма мовою C#.

calculate ['kælkjuleit] *v* 1) обчислювати, підраховувати.

CALL [kɔ:l] *abbr* (= computer-assisted (or -aided) language learning) система автоматизованого вивчення мови; вивчення мови за допомогою комп'ютера.

card [ka:d] *n* карта; карточка; плата; **smart** ~ смарт-карта, інтелектуальна карта, мікропроцесорна карточка; **video** ~ відеокарта.

card key [,ka:d 'ki:] *n* карточка-ключ.

Cartesian [kɑ:ˈti:zɪən] *adj* декартовий; ~ **geometry** аналітична геометрія (Декарта).

Cauchi [ˈkəʊʃi:] *n* Коші.

CD-ROM [ˌsi: di: ˈrɒm] *abbr* (= compact disk read-only memory) постійний запам'ятовувальний пристрій на компакт-диску; компакт-диск.

centimeter [ˈsentɪˌmi:tə] *n* (= centimetre) сантиметр.

central processing unit [ˌsentrəl ˈprəʊsesɪŋ ˌju:nɪt] *n* (= CPU) центральний процесор.

chip [ʧɪp] *n* мікросхема, чіп, кристал; **microprocessor** ~ кристал мікропроцесора, мікропроцесорний кристал; **silicon** ~ кремнієвий кристал.

cipher [ˈsaɪfə] **1.** *n* 1) цифра [арабська]; 2) нуль; 3) шифр, код; **in** ~ зашифрований; **2.** *v* 1) зашифровувати; 2) вираховувати.

ciphertext [ˈsaɪfətekst] *n* (= cipher text) зашифрований (шифрований, криптографічний) текст [повідомлення]; шифрограма, криптограма, шифротекст.

circle [ˈsə:kl] **1.** *n* 1) круг, коло; 5) цикл; **2.** *v* рухатися по колу; обертатися.

class [klɑ:s] *n* клас, тип об'єкта.

cleartext [ˈklɪətekst] *n* відкритий (вихідний) текст.

clockwise [ˈklɒkwaɪz] **1.** *adj* що рухається за стрілкою годинника; **2.** *adv* за стрілкою годинника; **counter-** ~ проти стрілки годинника.

code [kəʊd] **1.** *n* 1) код [програми]; **object** ~ об'єктний код, об'єктна програма; **source** ~ вихідний код, текст програми; 2) шифр, код, система кодування; **2.** *v* програмувати, писати програму, кодувати.

column [ˈkɒləm] *n* графа, стовпець, колонка; вертикальна лінія; ~ **vector** вектор(-стовпець).

compilation [ˌkɒmpɪˈleɪʃn] *n* компіляція, компілювання.

compile [kəmˈpaɪl] *v* компілювати.

compiler [kəmˈpaɪlə] *n* компілятор.

complex [ˈkɒmpleks] **1.** *n* 1) система, комплекс; **2.** *adj* 1) складний, комплексний, змішаний, комбінований; ~ **constant** комп-

лексна константа; ~ **number** комплексне число; ~ **polynomial** комплексний поліном; 2) заплутаний.

computational [ˌkɒmpjuˈteɪʃ(ə)n(ə)l] *adj* обчислювальний, числовий; ~ **mathematics** обчислювальна математика.

computer [kəmˈpjutə] *n* 1) комп'ютер, обчислювальна машина, ЕОМ; обчислювач; 2) обчислювальний, комп'ютерний; ~ **game** комп'ютерна гра; ~ **graphics** комп'ютерна графіка; ~ **virus** комп'ютерний вірус.

computer-aided [kəmˌpjutəˈeɪdɪd] *p. p.* автоматизований; такий, що виконується за допомогою обчислювальної машини; ~ **language learning** система автоматизованого вивчення мови; вивчення мови за допомогою комп'ютера.

computerized [kəmˌpjutəˈraɪzd] *adj* (= computerised) комп'ютеризований, оснащений комп'ютерами.

computing [kəmˈpjutɪŋ] **1.** *n* 1) обчислювальна техніка; 2) оброблення даних, робота із застосуванням комп'ютера, комп'ютеризація; 3) обчислення, підрахунок; **2.** *adj* обчислювальний; лічильний; рахунковий; ~ **center** обчислювальний центр; ~ **method** метод обчислення.

conclude [ˌkənˈkluːd] *v* робити висновок (умовивід).

configuration [kənˌfɪgəˈreɪʃn] *n* конфігурація, склад устаткування (обладнання).

congruence [ˈkɒŋgruəns] *n* конгруентність.

congruent(ial) [ˈkɒŋgruənt] *або* [ˈkɒŋgruənʃ(ə)l] *adj* конгруентний; що збігається; **linear ~ sequence** лінійна конгруентна послідовність.

conjugate [ˈkɒndʒugət] *adj* спряжений; ~ **complex number** комплексно-спряжене число.

consecutive [kənˈsekjutɪv] *adj* 1) послідовний; 2) логічний, упорядкований.

consider [kənˈsɪdə] *v* 1) розглядати; обмірковувати, обдумувати; 2) вважати.

constant [ˈkɒnstənt] **1.** *n* стала величина, константа, постійний коефіцієнт; **complex ~** комплексна константа; **2.** *adj* постійний; сталий.

conventional [kən'venʃ(ə)n(ə)l] *adj* звичайний, загальноприйнятий, традиційний; стандартний; ~ **cryptography** традиційна криптографія; ~ **encryption** традиційне шифрування (зашифровування).

converse ['kɒnvə:s] **1.** *n* 1) зворотне твердження (положення); 2) обернена теорема; **2.** *adj* обернений; зворотний; протилежний.

conversely ['kɒnvə:slɪ] *adv* навпаки.

coordinate **1.** *n* [kəu'ɔ:d(ɪ)nət] 1) координата; вісь координат; 2) *pl* координати; система координат; **2.** *v* [kəu'ɔ:dɪneɪt] координувати, погоджувати, узгоджувати; **3** *adj* (= co-ordinate) [kəu'ɔ:d(ɪ)nət] координатний.

correspond [ˌkɒrɪ'spɒnd] *v* 1) відповідати [чомусь]; узгоджуватись; співвідноситись; 2) збігатися [з чимось], бути аналогічним [до чогось].

corresponding [ˌkɒrɪ'spɒndɪŋ] *adj* відповідний.

counterclockwise [ˌkauntə'klɒkwaɪz] *adv* (= counter-clockwise) проти [руху] годинникової стрілки.

CPU [ˌsi: pi: 'ju:] *abbr* (= central processing unit) центральний процесор.

crack [kræk] **1.** *n* зламана програма; **2.** *v* зламувати [програму тощо].

cracking ['krækɪŋ] *n* кракінг, крекінг; злом, прорив [через захист комп'ютерної системи].

cryptanalysis [ˌkrɪptə'næləsɪs] *n* криптоаналіз.

cryptographic [ˌkrɪptə'græfɪk] *adj* криптографічний; шифрувальний; ~ **algorithm** криптографічний алгоритм.

cryptography [krɪp'tɒgrəfɪ] *n* криптографія; **conventional** ~ традиційна криптографія; **public key** ~ криптографія з відкритим ключем.

cryptology [krɪp'tɒlədʒɪ] *n* криптологія.

cryptosystem [ˌkrɪptə'sɪstɪm] *n* (= CS) криптографічна система, криптосистема; **hybrid** ~ гібридна криптосистема.

cure [kjuə] **1.** *n* "лікування" [зараженої вірусом програми]; **2.** *v* "лікувати", відновлювати [заражену вірусом програму або файл].

curve [kə:v] *n* 1) крива [лінія], дуга; **bell ~** крива нормального розподілу; **Gaussian error ~** крива помилок Гаусса; **Koch ~** крива Коха; 2) графік функції [у вигляді деякої кривої лінії]; 3) крива (діаграма).

cyberlaw ['saɪbəlɔ:] *n* кіберзакон.

cycle [saɪkl] **1.** *n* 1) цикл; період; 2) такт; **2.** *v* циклічно повторювати(сь); працювати циклами.

cyclically ['saɪklɪk(ə)lɪ] *adv* циклічно.

D

2D [ˌtu:'di:] *abbr* (= two dimensional) двовимірний; плоский.

3D [ˌθri:'di:] *abbr* (= three-dimensional) тривимірний, просторовий; стереоскопічний.

damage ['dæmɪʤ] **1.** *n* ушкодження, пошкодження; поломка; **2.** *v* пошкоджувати, руйнувати.

data ['deɪtə] *n pl* 1) *pl* від datum; 2) дані; відомості; інформація; **~ base** база даних; **~ hiding** обмеження доступу до даних; **~ member** 1) елемент [набору] даних; 2) об'єкт даних, визначений як елемент класу; **~ storage** 1) запам'ятовувальний пристрій для даних; 2) пам'ять [для зберігання] даних, сховище даних; 3) запам'ятовування (зберігання) даних; **to feed ~ into a computer** вводити дані у комп'ютер.

Data Encryption Standard [ˌdeɪtə ɪn'krɪpʃn ˌstændəd] *n* (= DES) стандарт шифрування даних, стандарт [шифрування] DES.

debug [ˌdi:'bʌg] *v* налагоджувати, виправляти помилки, усувати неполадки [у програмі або апаратурі].

debugger [ˌdi:'bʌgə] *n* налагоджувач, програма налагоджування.

debugging [ˌdi:'bʌgɪŋ] *n* 1) усунення дефектів [в устаткуванні], виправлення помилок; 3) налагоджування [програми].

decimal ['desɪm(ə)l] **1.** *n* десятковий дріб; десяткове число; **2.** *adj* десятковий; **~ fraction** десятковий дріб; **~ number** десяткове число.

decipher [dɪ'saɪfə] **1.** *n* дешифрування, розшифрування; **2.** *v* розшифровувати, дешифрувати.

decode [ˌdi:ˈkəud] *v* декодувати, дешифрувати, розшифровувати.

decrease **1.** *n* [ˈdi:kri:s] убування; зменшування; спадання, спад; пониження; **2.** *v* [di:ˈkri:s] убувати, убути; поменшати, зменшуватися; спадати.

decrypt [ˌdi:ˈkrɪpt] *v* розшифровувати.

decryption [ˌdi:ˈkrɪpʃn] *n* розшифрування, розшифровування.

deduce [dɪˈdju:s] *v* виводити [формулу]; робити [логічний] висновок.

definition [ˌdefiˈnɪʃ(ə)n] *n* 1) визначення, означення; опис; формулювання; 2) постановка [задачі]; задання [напр., початкових значень]; 3) чіткість; розбірливість.

degree [dɪˈgri:] *n* 1) степінь; 2) порядок; 3) градус.

delete [dɪˈli:t] *v* стирати, видаляти, викреслювати, знищувати, анульовувати, ліквідовувати, усувати.

denial-of-service attack [dɪˌnaɪəl əv ˈsə:vɪs əˌtæk] *n* атака з метою порушення нормального обслуговування користувачів.

denote [dɪˈnəut] *v* позначати; означати; значити.

depict [dɪˈpɪkt] *v* 1) малювати, зображати; 2) описувати, змальовувати.

depressed [dɪˈprest] *adj* понижений; ~ **equation** рівняння пониженого степеня.

derive [dɪˈraɪv] *v* 1) походити; ~**ed type** похідний тип; 2) виводити [формулу]; 3) диференціювати, брати похідну.

Descartes [deɪˈkɑ:t] *n* Декарт.

descending [dɪˈsendɪŋ] *adj* спадний, низхідний.

detect [dɪˈtekt] *v* виявляти; викривати.

detection [dɪˈtekʃ(ə)n] *n* виявлення; викриття.

determinant [dɪˈtə:mɪnənt] *n* детермінант, визначник.

determine [dɪˈtə:mɪn] *v* 1) визначати; встановлювати; 2) вираховувати, вирішувати.

device [dɪˈvaɪs] *n* 1) пристрій, прилад, пристосування, механізм, апарат; **input/output** ~ пристрій введення/виведення; 2) елемент, компонент; 3) схема; метод; спосіб.

diagram [ˈdaɪəgræm] **1.** *n* діаграма; схема, креслення; графік; **state** ~ діаграма станів; **2.** *v* 1) будувати діаграму; 2) зображати схематично.

differential [ˌdɪf(ə) ˈrenʃ(ə)l] **1.** *n* диференціал; **2.** *adj* диференціальний; ~ **equation** диференціальне рівняння; **partial** ~ **equation** диференціальне рівняння у часткових похідних.

digit [ˈdɪʤɪt] *n* 1) цифра, однозначне число [від 0 до 9]; 2) символ, знак; 3) ширина пальця [як міра = ¾ дюйма].

digital [ˈdɪʤɪtl] *adj* цифровий, дискретний, числовий; ~ **camera** цифрова (фото)камера; ~ **electronic machine** цифрова електронно-обчислювальна машина; ~ **signature** цифровий підпис.

dimension [daɪˈmenʃ(ə)n] **1.** *n* 1) вимір; 2) розмірність [масиву, матриці тощо]; **Hausdorff** ~ розмірність Хаусдорфа; 3) розмір; величина; об'єм; **2.** *v* проставляти розміри; надавати потрібного розміру.

discrete [dɪsˈkri:t] *adj* дискретний, переривчастий, окремий; ~ **analysis** дискретний аналіз.

disk [dɪsk] *n* диск; дисковий запам'ятовувальний пристрій; ~ **drive** дисковод, диск, дисковий механізм, накопичувач на дисках; **hard** ~ жорсткий диск, дисковод, вінчестер; **to copy data to** ~ копіювати дані на диск; **to save (write) data to** ~ записувати дані на диск; **to store data on** ~ зберігати дані на диску.

display [dɪˈspleɪ] **1.** *n* дисплей; **2.** *v* виводити на екран.

divide [dɪˈvaɪd] *v* 1) ділити(ся) (на – by); 2) градуювати, наносити поділки [на шкалу].

dividend [ˈdɪvɪdend] *n* ділене.

divisible [dɪˈvɪzəbl] *adj* що ділиться без остачі; подільний.

division [dɪˈvɪʒ(ə)n] *n* 1) ділення; знак ділення; 2) частина, секція; 3) розподіл; 4) підрозділ [фірми].

divisor [dɪˈvaɪzə] *n* дільник.

drive [draɪv] *n* дисковод, накопичувач [на дисках]; **disk** ~ дисковод, диск, дисковий механізм, накопичувач на дисках; **hard** ~ накопичувач на жорстких дисках; **tape** ~ стрічкопротяжний механізм.

dumpster diving [ˌdʌm(p)stə ˈdaɪvɪŋ] *n* "розгрібання сміття".

DVD [ˌdi: vi: ˈdi:] *abbr* (= Digital Versatile Disk) цифровий багатофункціональний диск.

E

E3 [ˌi: ˈθri:] *abbr* (= Electronic Entertainment Expo (or Exposition)) виставка E3; виставка електронних розваг.

edit [ˈedɪt] *v* 1) редагувати; 2) компонувати, зв'язувати.

editor [ˈedɪtə] редактор; текстовий редактор; програма редагування.

electronic [ˌelekˈtrɒnɪk] *adj* електронний; ~ **dictionary** електронний словник; ~ **equipment** електронне устаткування; ~ **mail** (= email, e-mail) електронна пошта.

element [ˈelɪmənt] *n* 1) елемент; складова частина; 2) мікросхема; 3) *pl* основи, початки [науки тощо].

elimination [ɪˌlɪmɪˈneɪʃ(ə)n, əˌlɪmɪˈneɪʃ(ə)n] *n* усунення, виключення; **Gauss – Jordan** ~ (метод) виключення Жордана – Гаусса.

e-mail [ˈi:meɪl] **1.** *n* (= electronic mail, = email) 1) електронна пошта; 2) текст електронного листа; ~ **address** адреса електронної пошти; ~ **box** електронна поштова скринька; **2.** *v* надсилати електронний лист.

email spoofing [ˌi:meɪl ˈspu:fɪŋ] *n* отримання доступу до електронної поштової скриньки обманом.

encapsulation [ɪnˌkæps(j)uˈleɪʃ(ə)n] *n* інкапсуляція.

encode [ɪnˈkəud, enˈkəud] *v* кодувати; шифрувати.

encrypt [ɪnˈkrɪpt] *v* шифрувати.

encryption [ɪnˈkrɪpʃn] *n* шифрування, зашифровування; **conventional** ~ традиційне шифрування; **private key** ~ шифрування індивідуальним ключем; **secret-key** ~ шифрування секретним ключем; **symmetric-key** ~ шифрування симетричним ключем.

engineering [ˌendʒɪˈnɪərɪŋ] *n* 1) проектування; конструювання; розроблення; **software** ~ розроблення програмного забезпечення; 2) інженерія; **mechanical** ~ механічна інженерія, машинобудування; 3) техніка; **robotic** ~ робототехніка.

enrollment [ɪnˈrəulmənt, enˈrəulmənt] *n* 1) реєстрація; 2) збирання біометричних даних.

ensue [ɪnˈsju:] *v* випливати [з – from, on].

equal ['i:kwəl] **1.** *v* 1) дорівнювати; 2) рівняти, зрівнювати, прирівнювати; **2.** *adj* 1) рівний, однаковий, рівносильний; 2) порівняний; конгруентний.

equality [ɪ(:)'kwɒlətɪ] *n* рівність.

equation [ɪ'kweɪʒ(ə)n] *n* 1) рівняння; 2) рівність; **depressed** ~ рівняння пониженого степеня; **integral** ~ інтегральне рівняння; **linear** ~ лінійне рівняння; **matrix** ~ матричне рівняння; **partial differential** ~ диференціальне рівняння у часткових похідних; **polynomial** ~ поліноміальне рівняння, рівняння многочлена.

equilateral [ˌi:kwɪ'læt(ə)r(ə)l] *adj* рівносторонній; ~ **triangle** рівносторонній трикутник.

error ['erə] *n* помилка [у програмі], похибка.

Euclid ['ju:klɪd] *n* Евклід.

even ['i:v(ə)n] *adj* 1) рівний, однаковий; 2) парний [про числа].

executable [ˌeksɪ'kju:təbl] *adj* що виконується; ~ **file** файл, який виконується.

execute ['eksɪkju:t] *v* виконувати [програму, команду].

express [ˌɪk'spres, ek'spres] *v* виражати.

expression [ˌɪk'spreʃ(ə)n] *n* [математичний] вираз.

F

face recognition [feɪs ˌrekəg'nɪʃ(ə)n] *n* розпізнавання облич.

factor ['fæktə] **1.** *n* 1) множник; 2) дільник; ~ **of algebraic polynomial** дільник багаточлена; 3) фактор, рухома сила; 4) показник; 5) коефіцієнт; **2.** *v* розкладати на множники.

factor theorem [ˌfæktə 'θɪərəm] теорема про подільність багаточлена на вираз $(x - a)$.

fighting ['faɪtɪŋ] *n* бійки; бійцівська гра.

figure ['fɪgə] **1.** *n* 1) цифра; 2) фігура; 3) рисунок, ілюстрація, креслення, діаграма; **2.** *v* 1) надавати форму; зображати [графічно]; 2) позначати цифрами; 3) розраховувати, вираховувати.

file [faɪl] *n* файл; **executable** ~ файл, який виконується; **object** ~ об'єктний файл, вихідний файл; **source** ~ вихідний файл;

2. *v* формувати (організовувати) файл; заносити у файл; зберігати у файлі.

fingerprint ['fɪŋgəprɪnt] **1.** *n* відбиток пальця; ~ **reader** зчитувач відбитків пальців, дактилоскопічний сканер; **2.** *v* знімати відбитки пальців.

finite ['faɪnaɪt] *adj* скінченний; ~ **series** скінченний ряд; ~ **set** скінченна множина.

fix [fɪks] *v* налагоджувати, виправляти [помилки у програмі].

flowchart ['fləʊtʃɑ:t] *n* блок-схема, структурна схема.

follow ['fɒləʊ] *v* випливати.

following ['fɒləʊɪŋ] **1.** *n* **the** ~ таке; **2.** *adj* наступний.

foregoing [fɔ:'gəʊɪŋ] *adj* попередній; вищезазначений.

form [fɔ:m] *n* 1) форма; 2) вираз, вигляд; **general** ~ загальний вигляд.

formula ['fɔ:mjələ, 'fɔ:mjulə] *n* (*pl* formulas, formulae) 1) формула; формулювання; аналітичний вираз; 2) формульний.

FPS [ˌef pi: 'es] *abbr* (= first-person shooter) шутер від першої особи.

fractal ['frækt(ə)l] *n* 1) фрактал; 2) фрактальний; рекурсивний; ~ **geometry** фрактальна геометрія.

functional ['fʌŋkʃ(ə)n(ə)l] *adj* функціональний; ~ **analysis** функціональний аналіз.

G

gait [geɪt] *n* хода.

game [geɪm] **1.** *n* [комп'ютерна] гра, ігрова програма; ~ **demo** демоверсія гри; *розм.* демка; ~ **designer** дизайнер гри; ~ **developer (programmer)** розробник гри; ~ **of chance** рулетка; ~ **theory** теорія ігор; **massively multiplayer online** ~ масова онлайнова гра. **2.** *v* грати [у комп'ютерну гру]; **3.** *adj* ігровий ~ **loop** ігровий цикл.

gameplay ['geɪmpleɪ] *n* геймплей, ігровий процес; сюжет, сценарій комп'ютерної гри.

gamer ['geɪmə] *n* геймер; гравець у комп'ютерні ігри.

Gauss [gaus] *n* Гаусс.

general ['ʤen(ə)rəl] *adj* 1) загальний; звичайний, основний; ~ **form** загальний вигляд; 2) загальний, повний.

genre [ʒɑ:ŋrə] *n* жанр [комп'ютерної гри].

geometrically [ˌʤi:əu'metrɪk(ə)lɪ] *adv* геометрично.

geometry [ʤɪ'ɒmɪtrɪ, ʤɪ'ɒmətrɪ] *n* 1) геометрія; геометрична форма; **Cartesian** ~ аналітична геометрія (Декарта); **fractal** ~ фрактальна геометрія; **non-Euclidean** ~ неевклідова геометрія; 2) форма, конфігурація; **hand** ~ конфігурація руки; **palm** ~ конфігурація долоні.

global ['gləub(ə)l] *adj* глобальний; ~ **network** глобальна мережа.

graph [grɑ:f] *n* 1) діаграма, графік; крива; 2) граф.

graphic ['græfɪk] 1. *n* 1) графіка; **computer** ~ комп'ютерна графіка; 2) графік, креслення, рисунок; графічні засоби; 3) графічні пристрої [введення-виведення]; пристрої введення-виведення графічних даних; 4) графічні дані; графічне зображення; 2. *adj* графічний; ~ **processor** графічний процесор.

group ['gru:p] 1. *n* група; ~ **theory** теорія груп; 2. *v* групувати.

H

hacker ['hækə] *n* хакер; комп'ютерний зломник.

hacking ['hækɪŋ] *n* хакерство; неавторизований доступ [до комп'ютерних даних], проникнення [у систему], злом програм.

hakish ['hækɪʃ] *adj* хакерський.

hand [hænd] *n* рука (кисть); ~ **geometry** конфігурація руки.

hard [hɑ:d] *adj* 1) апаратний; 2) жорсткий, твердий; ~ **disk** жорсткий диск, дисковод, вінчестер; ~ **drive** накопичувач на жорстких дисках; 3) постійний.

hardware ['hɑ:dweə] *n* 1) апаратні засоби; апаратура, обладнання; 2) апаратний.

headphone ['hedfəun] *n* навушник.

high tech [haɪ'tek] *n* (= high technology) високі технології.

Hilbert ['hɪlbət] *n* 1) Гільберт; 2) гільбертовий ~ **space** гільбертовий простір.

hypotenuse [haɪ'pɒt(ə)nju:z] *n* гіпотенуза.

I

I.Q. [ˌaɪ 'kju:] *abbr* (= intelligence quotient) коефіцієнт розумового розвитку.

ID [ˌaɪ 'di:] *abbr* (= identification, = identifier) 1) ідентифікація; 2) ідентифікатор, ознака; мітка.

identification [aɪˌdentɪfɪ'keɪʃn] *n* 1) ототожнення; 2) розпізнавання, пізнавання; установлення особи; ідентифікація; ~ **card** ідентифікаційна картка; посвідка особи.

identifier [aɪ'dentɪfaɪə] *n* ідентифікатор, ознака; мітка.

identify [aɪ'dentɪfaɪ] *v* 1) ототожнювати, встановлювати тотожність; 2) розпізнавати, пізнавати; ідентифікувати.

identity [aɪ'dentətɪ] *n* 1) тотожність, ідентичність; 2) справжність, правдивість; ~ **card** посвідка особи..

IFS [ˌaɪ ef 'es] *abbr* (= iterated function system) система ітерованих функцій.

illustrate ['ɪləstreɪt] *v* 1) пояснювати [прикладом]; 2) ілюструвати [малюнками тощо].

illustration [ˌɪləs'treɪʃ(ə)n] *n* 1) приклад, пояснення; 2) ілюстрація, малюнок.

image ['ɪmɪʤ] **1.** *n* образ, зображення; ~ **sensor** сенсор зображення; **2.** *v* зображати, відображати.

imaginary [ɪ'mæʤɪn(ə)rɪ] *adj* уявний; ~ **number** уявне число.

increase **1.** *n* ['ɪnkri:s] зростання, ріст; приріст; збільшення; **2.** *v* [ɪn'kri:s] зростати; збільшувати(ся); посилювати(ся)

indefinitely [ɪn'defɪnətlɪ, ɪn'def(ə)nətlɪ] *adv* необмежено, нескінченно, безмежно.

induction [ɪn'dʌkʃ(ə)n] *n* індукція; **mathematical** ~ математична індукція.

infinite ['ɪnfɪnət, 'ɪnfɪnɪt] **1.** *n* 1) безліч; 2) (the infinite) нескінченність, безмежність, безмежний простір; **2.** *adj* 1) нескінченний, безмежний; ~ **series** нескінченний ряд; 2) нескінченний, незліченний.

informatics [ˌɪnfəˈmætɪks] *n* інформатика; **social ~** соціальна інформатика.

information [ˌɪnfəˈmeɪʃn] *n* інформація, дані; **chief ~ security officer** керівник служби інформаційної безпеки; **~ system** інформаційна система; **~ technology** інформаційна технологія.

inheritance [ɪnˈherɪt(ə)ns] *n* наслідування.

input [ˈɪnpʊt] **1.** *n* 1) введення [інформації тощо], пристрій введення; 2) вхідний; 3) *pl* вхідні дані; вихідні дані; **2.** *v* вводити.

input/output [ˌɪnpʊt ˈaʊtpʊt] *adj* (= I/O) введення/виведення.

install [ɪnˈstɔːl] *v* 1) інсталювати, встановлювати, збирати, монтувати [апаратуру тощо]; 2) вводити у дію.

instruction [ɪnˈstrʌkʃ(ə)n] *n* 1) команда; 2) інструкція; програма дій; 3) навчання, інструктування, інструктаж.

integer [ˈɪntɪʤə] *n* 1) ціле число; **positive ~** додатне ціле число; 2) цілий, цілочисловий.

integral [ˈɪntɪgr(ə)l] **1.** *n* інтеграл; **2.** *adj* 1) інтегральний; **~ equation** інтегральне рівняння; 2) цілий, цілісний.

intelligent [ɪnˈtelɪʤ(ə)nt] *adj* інтелектуальний, "розумний"; оснащений мікропроцесором, мікропроцесорний; програмований, програмовуваний; **~ system** інтелектуальна система.

interface [ˈɪntəfeɪs] *n* інтерфейс; **user ~** інтерфейс користувача.

Internet [ˈɪntənet] *n* Інтернет, глобальна комп'ютерна мережа.

intersect [ˌɪntəˈsekt] *v* 1) перетинати(ся); 2) ділити на частини.

IP [ˌaɪ ˈpiː] *abbr* (= Internet Protocol) протокол IP; **~ address** IP-адреса.

iris [ˈaɪərɪs] *n* райдужна оболонка [ока]; **~ scanning** сканування [візерунка] райдужної оболонки [ока].

irrational [ɪˈræʃ(ə)n(ə)l] **1.** *n* ірраціональне число; **2.** *adj* ірраціональний.

J

joystick [ˈʤɒɪstɪk] *n* (маніпулятор) джойстик.

Jordan [ˈʤɔːd(ə)n] *n* Жордан.

K

key [ki:] **1.** *n* 1) ключ; кнопка, клавіша; перемикач; 2) ключ, шифр, код; ~ **distribution** розподіл ключів [у криптографії]; **private** ~ секретний (приватний) ключ; **public** ~ відкритий ключ; **secret** ~ секретний ключ; **session** ~ сеансовий ключ; **2.** *v* 1) перемикати, працювати ключем; 2) набирати на клавіатурі.

keyboard ['ki:bɔ:d] *n* клавіатура.

keyring ['ki:rɪŋ] *n* файл ключів, каталог ключів, "кільце для ключів".

keystroke ['ki:strəuk] *n* натискання клавіші *або* кнопки; ~ **pattern** характер (стиль) натискання клавіші *або* кнопки.

L

label ['leɪb(ə)l] **1.** *n* 1) мітка; ярлик; етикетка; 2) позначення; **2.** *v* 1) наліплювати ярлик; маркувати; 2) розмічати; позначати, помічати.

LAN [læn] *abbr* (= local area network) локальна (обчислювальна) мережа.

language ['læŋgwɪʤ] *n* 1) мова; **programming** ~ мова програмування; 2) мовний.

LED [ˌel i: 'di:] *abbr* (= light-emitting diode) світлодіод, світловипромінювальний діод; ~ **panel** світлодіодна панель.

length ['leŋ(k)θ] *n* 1) довжина; 2) відстань; 3) тривалість; 4) кількість елементів.

let [let] *v* (*як допоміжне дієслово виражає запрошення, наказ, дозвіл, припущення*); ~ ***ab*** **be equal to** ***cd*** припустимо, що *ab* дорівнює *cd*.

letter ['letə] **1.** *n* 1) буква, літера; **italicized** ~ літера, виділена курсивом; **lowercase** ~ мала літера, мала буква; **uppercase** ~ велика літера, велика буква, заголовна літера, заголовна буква; 2) символ, знак; 3) лист [електронною поштою]; **2.** *v* позначати буквами.

level ['lev(ə)l] *n* рівень.

lie [laɪ] *v* лежати; бути розташованим; знаходитись.

limit ['lɪmɪt] **1.** *n* границя, межа; **upper ~** верхня границя; **lower ~** нижня границя; **2.** *v* 1) обмежувати; ставити обмеження; 2) бути межею.

linear ['lɪnɪə] *adj* лінійний; прямолінійний; **~ algebra** лінійна алгебра; **~ congruential sequence** лінійна конгруентна послідовність; **~ equation** лінійне рівняння; **~ operation** лінійна операція; **~ space** лінійний простір.

linker ['lɪnkə] *n* лінкер, з'єднувач, редактор зв'язків.

M

machine [mə'ʃi:n] 1. *n* комп'ютер, машина, мережева станція; **Turing ~** машина Тьюрінга; 2) механізм; пристрій; **2.** *adj* машинний.

map [mæp] *n* карта.

mathematical [ˌmæθ(ə)'mætɪk(ə)l] *adj* математичний; **~ analysis** математичний аналіз; **~ concept** математичне поняття; **~ induction** математична індукція; **model(l)ing** математичне моделювання; **~ society** математичне товариство.

mathematician [ˌmæθ(ə)mə'tɪʃ(ə)n] *n* математик.

mathematics [ˌmæθ(ə)'mætɪks] *n* математика; **applied ~** прикладна математика.

matrix ['meɪtrɪks] *n* (*pl* matrices) 1) матриця; **negative ~** обернена матриця; **null ~** нульова матриця; **payoff ~** платіжна матриця, матриця виграшів; **rotation ~** матриця обертання, матриця повороту; **transposed ~** транспонована матриця; 2) матричний; **~ algebra** матрична алгебра; **~ equation** матричне рівняння; 3) таблиця.

measure ['meʒə] **1.** *n* 1) міра, одиниця вимірювання; **Lebesgue ~** лебегова міра; 2) показник, критерій; 3) масштаб; 4) дільник; **2.** *v* 1) міряти, вимірювати, відміряти; 2) мати розмір.

measurement ['meʒ(ə)mənt] *n* 1) вимірювання; вимір, розмір; *pl* розміри.

mechanical [mɪ'kænɪk(ə)l] *n* 1) механічний; 2) автоматичний; 3) машинний; машинобудівний; ~ **engineering** механічна інженерія, машинобудування.

mechanics [mɪ'kænɪks] *n* механіка.

member ['membə] *n* 1) член; ~ **of equation** член рівняння; 2) елемент [пристрою].

memory ['meməɪ] *n* [компютерна] пам'ять, запам'ятовуваль-ний пристрій [машини]; **main** ~ оперативна пам'ять; **random access** ~ оперативна пам'ять, пам'ять з випадковим доступом; **read only** ~ постійна пам'ять, постійний запам'ятовувальний пристрій.

message ['mesɪʤ] *n* 1) повідомлення; 2) лист.

method ['meθəd] *n* 1) метод; спосіб; ~**s of optimization** методи оптимізації; **numerical** ~ **of linear algebra** числові методи лінійної алгебри; 2) правило.

microphone ['maɪkrəfəun] *n* мікрофон; **stereo** ~ стереомікрофон.

microprocessor [ˌmaɪkrə'prəusesə] *n* мікропроцесор; ~ **chip** кристал мікропроцесора, мікропроцесорний кристал.

MIDI ['mɪdɪ] *abbr* (= musical instrument digital interface) цифровий інтерфейс музичних інструментів.

minutiae [maɪ'nju:ʃɪi:, maɪ'nju:ʃɪaɪ] *n* 1) шаблон відбитка пальця; детальний протокол введення відбитка пальця; 2) *n pl* дрібниці; деталі.

MMO [ˌem em 'əu] *abbr* (= Massively Multiplayer Online) масова онлайнова [гра].

modulo ['mɒdjələu] *adv* модуль; модульний; за модулем; ~ **arithmetic** модульна арифметика; арифметичні операції над абсолютними значеннями чисел; ~ **N** за модулем N.

modulus ['mɒdjələs] *n* 1) модель, основа системи числення; 2) показник степеня; 3) коефіцієнт; 4) абсолютне значення, модуль.

moiré fringe [ˌmwɑ: 'frɪnʤ] *n* муарова (інтерференційна) смуга.

monitor ['mɒnɪtə] *n* монітор.

motherboard ['mʌðəbɔ:d] *n* материнська плата, системна плата, основна плата.

mouse [maus] *n* (комп'ютерна) мишка; **to click on the ~** клацнути мишею.

multimedia [ˌmʌltɪ'mi:dɪə] **1.** *n* 1) мультимедіа; 2) мультимедійні засоби; **2.** *adj* 1) мультимедійний.

multiple ['mʌltɪpl] **1.** *n* кратне число; **2.** *adj* 1) кратний; 2) багаторазовий; численний.

multiplication [ˌmʌltɪplɪ'keɪʃ(ə)n] *n* множення; перемноження.

multiply ['mʌltɪplaɪ] *v* 1) множити; перемножувати; 2) збільшувати.

N

negative ['negətɪv] **1.** *n* 1) від'ємна величина; **~ number** від'ємне число; 2) знак мінус; **~ transpose** транспозиція зі зміною знака матриці; **2.** *adj* 1) від'ємний; 2) мінусовий; **~ sign** знак мінус; 3) негативний; 4) протилежний, обернений **~ matrix** обернена матриця.

network ['netwə:k] *n* (обчислювальна) мережа; **global ~** глобальна мережа; **local area ~** локальна мережа; **wide area ~** глобальна мережа.

non-Euclidean [ˌnɒnju'klɪdɪ(:)ən] *adj* неевклідовий; **~ geometry** неевклідова геометрія.

non-integer [ˌnɒn'ɪntɪʤə] *n* неціле.

nonnegative ['nɒnˌnegətɪv] *adj* невід'ємний.

nonzero [ˌnɒn'zɪərəu] *adj* 1) ненульовий, не дорівнює нулю; відмінний від нуля; 2) ненульовий елемент [матриці].

notation [nəu'teɪʃ(ə)n] *n* нотація; система позначення; запис.

null [nʌl] **1.** *n* нуль; **2.** *adj* нульовий; **~ matrix** нульова матриця.

number ['nʌmbə] **1.** *n* 1) число; цифра; **complex ~** комплексне число; **decimal ~** десяткове число; **negative ~** від'ємне число; **positive ~** додатне число; **pseudo-random ~** псевдовипадкове число; **random ~** випадкове число; **rational ~** раціональне число; **real ~** дійсне число; 2) число, кількість, сума; 3) номер; **2.** *v* 1) нумерувати; 2) рахувати, нараховувати.

O

object ['ɒbʤɪkt] *n* 1) об'єкт; 2) об'єктний; ~ **code** об'єктний код, об'єктна програма; ~ **file** об'єктний файл, вихідний файл.

object-oriented [ˌɒbʤɪkt 'ɔːrɪentɪd] *adj* (= OO, = O2) об'єктно-орієнтований; ~ **language** об'єктно-орієнтована мова [програмування]; ~ **programming** об'єктно-орієнтоване програмування.

obtain [əb'teɪn] *v* 1) одержувати; діставати; здобувати; 2) існувати; бути визнаним; застосовуватися.

odd ['ɒd] *adj* непарний.

on/off [ˌɒn'ɒf] *adj* 1) "ввімкнено – вимкнено"; 2) двопозиційний; ~ **switch** двопозиційний перемикач, перемикач на два напрямки.

online [ˌɒn'laɪn] *adj* (= on-line) 1) постійно увімкнений [пристрій], неавтономний режим роботи; 2) під'єднаний до комп'ютера або доступний через комп'ютер 3) електронний, екранний, оперативний, діалоговий, інтерактивний, онлайновий; ~ **language courses** онлайнові курси вивчення мови;

operating ['ɒpəreɪtɪŋ] *adj* операційний; ~ **system** операційна система; 2) робочий, діючий.

operation [ˌɒpə'reɪʃn] *n* операція; дія; ~**s research** дослідження операцій; 2) робота; функціонування; 3) режим [роботи]; 4) спрацювання [приладу].

optical coupler [ˌɒptɪkl 'kʌplə] *n* волоконно-оптичний елемент зв'язку, оптрон.

optimization [ˌɒptɪmaɪ'zeɪʃ(ə)n] *n* оптимізація; **methods of** ~ методи оптимізації.

order ['ɔːdə] *n* 1) степінь; порядок; кратність.

output ['aʊtpʊt] *n* 1) вивід, виведення [даних]; пристрій виведення; 2) вихідний [сигнал, контакт тощо]; 3) *pl* вихідні дані.

P

packet sniffing [ˌpækɪt snɪfɪŋ] *n* несанкціоноване перехоплення даних, які передаються по мережі; моніторинг пакетів.

palm ['pɑːm] *n* долоня; ~ **geometry** конфігурація долоні.

parenthesis [pə'renθɪsɪs] *n* (*pl* parentheses) кругла дужка.

password ['pɑ:swə:d] *n* пароль.

pattern ['pætən] **1.** *n* 1) шаблон; взірець; трафарет; модель; 2) візерунок, орнамент; 3) образ, зображення; ~ **recognition** розпізнавання образів; 4) копія; **2.** *v* 1) моделювати; 2) формувати зображення; 3) копіювати.

peripherals [pə'rɪfərəls] *n pl* периферійне обладнання, периферія; зовнішнє обладнання, зовнішні пристрої.

PGP [ˌpi: ʤi: 'pi:] *abbr* (= Pretty Good Privacy) цілком хороша секретність, програма шифрування PGP.

PIN [pɪn] *abbr* (= Personal Identification Number) особистий ідентифікаційний номер, PIN-код.

plaintext ['plemtekst] *n* відкритий (незашифрований) текст.

platformer ['plætfɔ:mə] *n* платформер.

play [pleɪ] *v* грати [у комп'ютерну гру].

player [pleɪə] *n* 1) програвач; 2) гравець.

plug [plʌg] **1.** *n* штепсель; роз'єм; **2.** *v* 1) підключати; 2) вставляти [у роз'єм, вилку у розетку тощо].

plus [plʌs] *n* **1.** 1) знак плюс ("+"), плюс; ~ **sign** знак плюс ("+"); 2) додатна величина; **2.** *adj* зі знаком плюс; **3.** *prep* плюс; **four ~ five is nine** чотири плюс п'ять буде дев'ять.

point [pɔɪnt] **1.** *n* 1) місце, точка; ~ **of inflection** точка перегину [кривої]; 2) кома [у десяткових дробах]; **three ~ five [3.5]** три і п'ять десятих; 3) крапка; 4) поділка шкали; **2.** *v* 1) перемістити вказівник миші на екрані, не клацаючи мишею; 2) показувати, указувати, посилатись.

polymorphism [ˌpɔlɪ'mɔ:fɪzm] *n* поліморфізм.

polynomial [ˌpɔlɪ'nəumj(ə)l] **1.** *n* поліном, багаточлен; **2.** *adj* поліноміальний, багаточленний; ~ **equation** поліноміальне рівняння, рівняння багаточлена.

positive ['pɔzətɪv] **1.** *n* додатна величина; **2.** *adj* додатний; ~ **integer** додатне ціле число.

power ['pauə] *n* 1) сила; потужність, енергія; продуктивність; 2) степінь, показник степеня; ***a* to the fifth** ~ *a* у п'ятому степені.

preprocessor [ˌpriːˈprəʊsesə] *n* препроцесор.

private [ˈpraɪvɪt] *adj* 1) приватний, особистий, персональний; 2) секретний, таємний; ~ **key** секретний (приватний) ключ.

problem [ˈprɒbləm] *n* 1) проблема; завдання; питання; 2) задача.

proceed [prəˈsiːd] *v* продовжувати.

process [ˈprəʊses] **1.** *n* процес; **2.** *v* обробляти [дані].

processor [ˈprəʊsesə] *n* процесор; **graphic(s)** ~ графічний процесор.

product [ˈprɒdʌkt] *n* 1) добуток; **dot** ~ скалярний добуток; 2) продукція; продукт; виріб; 3) результат, наслідок.

program [ˈprəʊgræm] **1.** *n* програма; **to load a** ~ **into a computer** завантажувати програму у комп'ютер; **to write a** ~ **in C#** писати програму [мовою] C#; **2.** *v* програмувати.

programmable [prəˈgræməbl] *adj* 1) програмований, програмовуваний; програмовний; 2) з програмним керуванням.

programmer [ˈprəʊgræmə] *n* програміст.

programming [ˈprəʊgræmɪŋ] *n* програмування; ~ **language** мова програмування; **object-oriented** ~ об'єктно-орієнтоване програмування.

project manager [ˌprɒdʒəkt ˈmænɪdʒə] *n* керівник проекту.

proof [ˈpruːf] *n* 1) доведення, доказ; 2) перевірка, випробування.

proposition [ˌprɒpəˈzɪʃ(ə)n] *n* 1) твердження, судження, висловлювання; 2) теорема; 3) припущення; 4) задача, проблема.

protect [prəˈtekt] *v* захищати (від – from, проти – against); запобігати, відвертати; охороняти.

prove [ˈpruːv] *v* 1) доводити; 2) перевіряти.

pseudo-random [ˌpsjuːdəuˈrændəm] *adj* (= pseudorandom) псевдовипадковий; ~ **numbers** псевдовипадкові числа.

public [ˈpʌblɪk] *adj* публічний, загальнодоступний, відкритий; ~ **key** відкритий ключ; ~ **key cryptography** криптографія з відкритим ключем.

puzzle [ˈpʌzl] *n* логічна гра; головоломка.

Pythagoras [paɪˈθægərəs] *n* Піфагор.

Pythagorean [paɪˌθægəˈriːən] **1.** *n* піфагорієць, послідовник Піфагора; **2.** *adj* піфагорів; ~ **theorem** теорема Піфагора.

Q

quotient ['kwəuʃ(ə)nt] *n* 1) частка; 2) коефіцієнт; **intelligence ~** коефіцієнт розумового розвитку (інтелекту).

R

racing ['reɪsɪŋ] *n* гонки, перегони.

RAM [ræm] *abbr* (= Random Access Memory) оперативна пам'ять; пам'ять із випадковим доступом.

random ['rændəm] *adj* випадковий, вибраний наздогад; безладний, нерегулярний; **~ number** випадкове число.

range [reɪndʒ] **1.** *n* 1) діапазон; 2) відрізок, зона, область, інтервал; 3) серія, ряд; 4) область значень функції; **2.** *v* розташовувати, розставляти [по порядку].

ratio ['reɪʃɪəu] *n* відношення, пропорція; коефіцієнт; співвідношення.

rational ['ræʃənl] *adj* раціональний; **~ number** раціональне число.

reader ['ri:də] *n* зчитувач; зчитувальний пристрій.

real [rɪəl] *adj* 1) дійсний; **~ number** дійсне число; 2) реальний.

reasoning ['ri:zɪŋ] *n* 1) міркування; 2) мислення; 3) пояснення; аргументація.

receiver [rɪ'si:və] *n* приймач.

receptacle [rɪ'septəkl] *n* 1) приймач; 2) отвір.

recipient [rɪ'sɪpɪənt] *n* приймач, реципієнт, одержувач [інформації]

rectangular [rek'tæŋgjulə] *adj* прямокутний; **~ array** прямокутна [геометрична] структура; прямокутна [графічна] структура; прямокутний [геометричний] масив; прямокутний [графічний] масив; [геометрична] структура у прямокутних координатах, [геометричний] масив у прямокутних координатах.

recur [rɪ'kə:] *v* повторюватися, (по)вертатися.

recurrent [rɪ'kʌr(ə)nt] *adj* періодичний; рекурентний.

recursive [rɪ'kə:sɪv] *adj* рекурсивний.

reflexive [rɪ'fleksɪv] *adj* рефлексивний; зворотний.

relationship [rɪ′leɪʃ(ə)nʃɪp] *n* (взаємо)зв'язок, взаємовідношення; співвідношення; залежність.

remainder [rɪ′meɪndə] *n* остаток, остача [від ділення].

repository [rɪ′pɒzɪtə rɪ] *n* 1) репозиторій; репозитарій; 2) сховище; 3) архів.

reprogrammable [ˌriː′prəʊgræməbl] *adj* який можна перепрограмувати.

resident [′rezɪd(ə)nt] **1.** *n* резидент, резидентна програма [дані, файл тощо]; **2.** *adj* резидентний; ~ **monitor** резидентний монітор.

respectively [rɪ′pektɪvlɪ] *adv* 1) стосовно кожного зокрема; 2) відповідно, у вказаній послідовності.

retina [′retnə] *n* сітківка, сітчаста оболонка (ока).

retinal [′retɪnl] *adj* який стосується сітківки; ~ **scanning** сканування [візерунка] сітківки очей.

robot [′rəubɒt] *n* робот.

roboticist [rəu′bɒtɪsɪst] *n* робототехнік.

robotics [rəu′bɒtɪks] *n* 1) робототехніка; 2) робототехнічний.

ROM [rɒm] постійна пам'ять, постійний запам'ятовувальний пристрій.

root [ruːt] *n* 1) корінь; **cube (third)** ~ кубічний корінь; ***n*-th** ~ корінь *n*-го степеня; *n*-й корінь рівняння; **real** ~ дійсний корінь; **square (second)** ~ квадратний корінь; 2) кореневий каталог.

row [rəu] *n* ряд; рядок; ~ **vector** вектор(-рядок).

RPG [ˌɑː piː ′dʒiː] *abbr* (= role-playing game) рольова гра.

RTS [ˌɑː tiː ′es] *abbr* (= real time strategy) стратегія у реальному часі.

run [rʌn] **1.** *n* 1) виконання, запуск [програми]; 2) робота [машини]; **2.** *v* виконувати [програму].

S

salami slicing [səˌlɑːmɪ ′slaɪsɪŋ] викрадення грошей регулярно невеликими сумами.

scan [skæn] **1.** *n* 1) сканування; 2) перегляд; пошук; **2.** *v* 1) сканувати; 2) переглядати, проглядати; шукати.

scanner ['skænə] *n* сканер; скануючий пристрій; **virus ~** програма пошуку вірусів.

scanning ['skænɪŋ] *n* 1) сканування; **retinal ~** сканування [візерунка] сітківки очей; 2) перегляд; пошук.

screen [skri:n] *n* екран.

scroller ['skrəʊlə] *n* скролер; **side ~** боковий скролер; скролер із видом збоку.

secure [sɪ'kjʊə] *adj* 1) безпечний, у безпеці (від – from); захищений; надійний; 2) секретний; таємний; **2.** *v* 1) гарантувати безпеку; захищати; охороняти.

SecurID [sɪ,kjʊraɪ'di:] *abbr* (= Security Identifier) ідентифікатор безпеки (захисту) [користувача].

security [sɪ'kjʊərɪtɪ] *n* 1) безпека, безпечність, надійність; 2) охорона, захист; 3) органи безпеки, служба безпеки; **chief information ~ officer** керівник служби інформаційної безпеки; **~ personnel** персонал (працівники) служби безпеки; 4) секретність, конфіденційність, захищеність.

seed [si:d] *n* початкове число.

segment ['segmənt] **1.** *n* сегмент, відрізок; **2.** *v* ділити(ся) на сегменти.

self-similarity ['self ,sɪmɪ'lærɪtɪ] *n* самоподібність.

self-similar ['self ,sɪmɪlə] *adj* подібний самому собі.

semigroup [,semɪ'gru:p] *n* півгрупа.

sender ['sendə] *n* відправник [повідомлення].

sensor ['sensə] *n* сенсор, давач, чутливий елемент; **acceleration ~** сенсор прискорення; **distance ~** далекомір, відстанемір; сенсор відстані; **image ~** сенсор зображення; **vibration ~** сенсор вібрації.

sequence ['si:kwəns] *n* послідовність; порядок, ряд.

series ['sɪəri:z] *n* ряд, послідовність.

server ['sə:və] *n* сервер.

set [set] **1.** *n* 1) множина; ряд; послідовність; **Mandelbrot ~** множина Манделброта; **~ theory** теорія множин; 2) набір;

комплект; 3) встановлення; **2.** *v* 1) присвоювати [початкове] значення [змінної]; 2) встановлювати у певне положення; 3) монтувати; налагоджувати.

shooter ['ʃu:tə] *n* шутер, "стрілялка"; **first-person** ~ шутер від першої особи; **third-person** ~ шутер від третьої особи.

sign [saɪn] **1.** *n* 1) знак, позначка, символ; **plus** ~ знак плюс ("+"); ~ **variation** зміна знака, знакозміна; **2.** *v* 1) ставити знак, відмічати; 2) підписувати(ся).

signature ['sɪgnɪtʃə] *n* 1) підпис; **digital** ~ цифровий підпис; 2) сигнатура.

silicon ['sɪlɪkən] *n* 1) кремній; 2) кремнієвий ~ **chip** кремнієвий кристал.

simulation [ˌsɪmju'leɪʃ(ə)n] *n* симулятор, симуляція.

since ['sɪns] *conj* через те що; з того що; оскільки.

size [saɪz] *n* розмір; об'єм, обсяг; величина.

slope [sləup] *n* спад [кривої].

slot [slɒt] *n* 1) (щілинний) отвір; 2) рознімач, гніздо, *розм.* слот.

smart card ['smɑ:t kɑ:d] *n* смарт-карта, інтелектуальна карта, мікропроцесорна карточка.

software ['sɒftwɛə] *n* 1) програмне забезпечення, програми, комплекс програм; 2) засоби програмування.

software piracy [ˌsɒftweə 'paɪrəsɪ] *n* комп'ютерне піратство.

solution [sə'lu:ʃ(ə)n] *n* розв'язок, розв'язання.

space ['speɪs] *n* 1) інтервал; пробіл; 2) проміжок, віддаль, відстань; 3) простір; **Banach** ~ банаховий простір; **Hilbert** ~ гільбертовий простір; **linear** ~ лінійний простір.

spam [spæm] **1.** *n* спам, "ковбасний фарш"; **2.** *v* посилати (відсилати) спам.

speaker ['spi:kə] *n* динамік ПК, акустична колонка.

sport(s) [spɔ:t(s)] *n* спортивна гра.

spread [spred] **1.** *n* 1) поширення; зростання; 2) розкид; **2.** *v* поширювати(ся); розкидати(ся).

square ['skwɛə] **1.** *n* 1) квадрат; 2) квадрат, другий степінь; **least ~s method** метод найменших квадратів; **perfect** ~ квадрат простого числа; повний квадрат; **2.** *adj* 1) квадратний; 2) пря-

мокутний; 3) у квадраті; квадратний; ~ **root** квадратний корінь; 4) під прямим кутом, перпендикулярний (чомусь – with, to).

state [steɪt] **1.** *n* 1) стан; ~ **diagram** діаграма станів; 2) режим; **2.** *v* 1) констатувати; формулювати; 2) встановлювати, точно визначати; 3) передавати знаками (формулами); 4) ставити [задачу].

statement ['steɪtmənt] *n* 1) твердження, висловлювання; формулювання.

storage ['stɔ:rɪdʒ] *n* 1) [зовнішня] пам'ять, зовнішній пристрій для збереження даних; 2) пам'ять [основна]; 3) зберігання [інформації]; **data** ~ 1) запам'ятовувальний пристрій для даних; 2) пам'ять [для зберігання] даних, сховище даних; 3) запам'ятовування [зберігання] даних; ~ **device** пристрій зберігання даних.

store ['stɔ:] *v* запам'ятовувати, зберігати.

submatrix ['sʌbmeɪtrɪks] *n* підматриця.

subscript ['sʌbskrɪpt] *n* 1) нижній [підрядковий] індекс; 2) список індексів.

substitution [ˌsʌbstɪ'tju:ʃ(ə)n] *n* підстановка.

subtract [səb'trækt] *v* віднімати.

succession [s(ə)k'seʃ(ə)n] *n* послідовність; безперервний ряд.

successive [s(ə)k'sesɪv] *adj* 1) наступний; 2) що йде один за одним, послідовний.

successively [s(ə)k'sesɪvlɪ] *adv* підряд; послідовно.

sum [sʌm] **1.** *n* 1) сума; кількість; підсумок; 2) арифметична задача; арифметичний приклад; 3) *pl* арифметика; **2.** *v* додавати; підсумовувати, підбивати підсумки.

support [sə'pɔ:t] *n* підтримка; забезпечення; **software** ~ програмна підтримка, підтримка програмного виробу (продукту); **2.** *v* підтримувати, забезпечувати.

synthetic division [sɪnˌθetɪk dɪ'vɪʒ(ə)n] *n* ділення [поліномів] з остачею.

system ['sɪstəm] *n* 1) система; 2) система; установка; пристрій; комплекс; ~ **intelligent** інтелектуальна система; **operating** ~ операційна система; 3) системний; ~ **administrator** системний

адміністратор [мережі]; ~ **Analysis and Management** системний аналіз і управління; ~ **clock** системний годинник, годинник системного часу; системний тактовий генератор; ~ **unit** системний блок; 4) програма.

T

table ['teɪbl] *n* таблиця.

tangent ['tændʒ(ə)nt] **1.** *n* 1) тангенс; 2) дотична; **2.** *adj* дотичний.

team [ti:m] *n* команда, група, колектив.

technology [tek'nɒlədʒɪ] *n* 1) технологія; технологічна культура; технологічний рівень; 2) техніка; технічні та прикладні науки; 3) спеціальна термінологія.

template ['templɪt] *n* 1) еталон (у системі розпізнавання), біометричний зразок; 2) шаблон, трафарет.

term [tə:m] *n* 1) член, елемент; 2) термін.

test [test] **1.** *n* 1) тест, тестова програма; **Turing** ~ тест Тьюрінга; 2) тестування, перевірка, випробування; **2.** *v* тестувати; випробовувати; перевіряти.

testable ['testəbl] *adj* придатний для контролю.

tester ['testə] *n* тестер.

testing ['testɪŋ] *n* тестування, перевірка, випробування.

theorem ['θɪərəm] *n* теорема; **factor**~ теорема про подільність багаточлена на вираз $(x - a)$; **the fundamental ~ of algebra** основна теорема алгебри (про існування кореня алгебричного рівняння); **to prove a** ~ доводити теорему.

theory ['θɪərɪ] *n* 1) теорія; теоретичні основи (принципи); **group** ~ теорія груп; **optimal processes** ~ теорія оптимальних процесів; **set** ~ теорія множин; **the ~ of measure and integration** теорія міри та інтеграла; **the ~ of the linear operations** теорія лінійних операцій [полів], теорія лінійних операторів; 2) метод.

therefore ['ðeəfɔ:] *adv* отже, тому.

three-dimensional [ˌθri:daɪ'menʃənl] *adj* (= 3D) тривимірний, просторовий; стереоскопічний.

thus [ðʌs] *adv* так, отже; тому; наприклад.

tile [taɪl] **1.** *n* 1) елемент мозаїчного зображення; 2) мозаїка [у графічному інтерфейсі користувача]; **2.** *v* розміщувати у мозаїчний спосіб.

token ['təuk(ə)n] *n* ознака; мітка, ярлик, маркер; засіб ідентифікації; **security ~** маркер доступу.

topology [tə'pɒlədʒɪ] *n* топологія; **general ~** загальна топологія.

transform [træn'sfɔ:m] *v* (видо)змінювати(ся), перетворювати(ся); трансформувати(ся).

transmit [træn'smɪt], [trænz'mɪt] *v* 1) передавати; 2) відправляти; 3) пересилати, надсилати.

transmission 1) передача; 2) пересилання [даних]; обмін; 3) пропускання, проходження [напр., сигналу].

transpose [træns'pəuz] **1.** *n* транспозиція матриці; **negative ~** транспозиція зі зміною знака матриці; **2.** *v* 1) переміщувати, переставляти; перегруповувати; 2) транспонувати; переносити до іншої частини рівняння з оберненим знаком; **to ~ a matrix** транспонувати матрицю; **~ed matrix** транспонована матриця.

transposition [ˌtrænspə'zɪʃ(ə)n] *n* транспонування; транспозиція; перенесення членів [з однієї частини рівняння в іншу]; перестановка.

triangle ['traɪæŋgl] *n* трикутник; **equilateral ~** рівносторонній трикутник; **right angled ~** прямокутний трикутник; **Sierpinski ~** трикутник Серпінського.

true [tru:] *adj* правильний; істинний.

Turing *n* ['tjʊərɪŋ] *n* Тьюрінг; **~ machine** машина Тьюрінга; **~ test** тест Тьюрінга.

U

unit ['ju:nɪt] *n* 1) пристрій, прилад, блок, вузол, структурний елемент; 2) одиниця; 3) одиниця [обладнання]; 4) одиниця виміру; 5) компонент (модуль) програми.

unknown [ˌʌn'nəun] *n* невідоме, невідома величина; **equation in single ~** рівняння з одним невідомим; **2.** *adj* невідомий.

uppercase [ˌʌpə'keɪs] *adj* (= upper case) [про букви] великий, заголовний; ~ **letter** велика літера, велика буква, заголовна літера, заголовна буква.

user ['ju:zə] *n* користувач; ~ **interface** інтерфейс користувача; ~**-defined type** визначений користувачем тип даних.

V

valid ['vælɪd] *adj* 1) допустимий, дійсний, достовірний, правильний, істинний; справедливий; чинний; 2) ефективний, дієвий.

value ['vælju:] *n* величина; значення.

variable ['ve(ə)rɪəb(ə)l] 1. *n* змінна величина; 2. *adj* змінний.

variation [ˌvɛərɪ'eɪʃ(ə)n] *n* 1) зміна; відхилення, коливання; **sign** ~ знакозміна, зміна знака; 2) різновид; варіант; 3) варіація.

VBScript [ˌvi: bi: 'skrɪpt] *abbr* (= Visual Basic Script) мова VBScript.

vector ['vektə] *n* 1) вектор; **column** ~ вектор(-стовпець); **row** ~ вектор(-рядок); 2) векторний; **complete normed** ~ **space** повний нормований векторний простір.

Venn [ven] *n* Венн.

verification [ˌverɪfɪ'keɪʃ(ə)n] *n* верифікація, контроль, перевірка, звіряння.

verify ['verɪfaɪ] *v* верифікувати, контролювати, перевіряти, звіряти.

via ['vaɪə] *prep* через.

virus ['vaɪərəs] *n* [комп'ютерний] вірус; ~ **scanner** програма пошуку вірусів.

voice [vɔɪs] *n* 1) голос; ~ **recognition** розпізнавання (ідентифікація) голосу; 2) голосовий, мовленнєвий.

W

Web [web] *n* "Всесвітня павутина".

website ['websaɪt] *n* Веб-сторінка, Веб-сайт, Інтернет-сайт.

wireless ['waɪəlɪs] *adj* бездротовий.

workstation [ˈwəːksteɪʃn] *n* робоча станція.
worm [ˈwəːm] *n* "черв'як".

Y

yield [jiːld] *v* давати результат, призводити до чогось.

Z

zoom [ˈzuːm] **1.** *n* [інструмент] "лупа"; **2.** *v* змінювати масштаб зображення; ~ **in** збільшити масштаб зображення; ~ **out** зменшити масштаб зображення.

List of Sources

1. Aibo.: [Електронний документ]. – (http://en.wikipedia.org/wiki/ Aibo). Перевірено 26.02.2009.
2. Asimov, A. The Fun They Had: [Електронний документ]. – (http://www.tasc.ac.uk/depart/media/staff/ls/Modules/MED2350/ Asimov.html). Перевірено 26.02.2009.
3. Brief History of L'viv University: [Електронний документ]. – (http://www. franko.lviv.ua). Перевірено 26.02.2009.
4. Chapman, D. Teach Yourself Visual C++ in 21 Days: [Електронний документ] / Sams Publishing, 1999. (http:// www-linux. gsi.de/ ~dantoncz/cpp/Cpp_21_days). Перевірено 26.02.2009.
5. Clarke, R. Asimov's Laws of Robotics: Implications for Information Technology: [Електронний документ]. – (http://www.asimov-laws.com/articles/archives/2004/07/asimovs_ laws_of.html). Перевірено 26.02.2009.
6. Computer and Video Games: [Електронний документ]. – (http:// en.wikipedia. org/wiki/ Computer and video games). Перевірено 26.02.2009.
7. Cryptography: [Електронний документ]. – (http://en.wikipedia. org/wiki/ Cryptography). Перевірено 26.02.2009.
8. E3: [Електронний документ]. – (http:www.org/wiki/E3). Перевірено 26.02.2009.
9. Esteras, S. R. Infotech. English for computer users. CUP, 1996.
10. Game Design: [Електронний документ]. – (http://en.wikipedia. org/wiki/Game design). Перевірено 26.02.2009.
11. Game Programmer: [Електронний документ]. – (http://en. wikipedia.org/wiki/Gameprogrammer). Перевірено 26.02.2009.
12. Game Programming: [Електронний документ]. – (http://en. wikipedia.org/wiki/ Gameprogramming). Перевірено 26.02.2009.
13. Home Network Security: [Електронний документ]. – (http://www. cert.org/tech_tips/home_networks.html). Перевірено 26.02.2009.
14. Howland G. "How do I make games?" A Path to Game Development: [Електронний документ]. – (http://www.lupinegames. com/ articles/path_to_dev.html). Перевірено 26.02.2009.

15. ILOVEYOU: [Електронний документ]. – (http://en.wikipedia.org/wiki/ILOVEYOU). Перевірено 26.02.2009.
16. Introduction to Cryptography: [Електронний документ]. – (http://access. adobe.com/access_form.html). Перевірено 26.02.2009.
17. Kurtz, M. Handbook of Applied Mathematics for Engineers and Scientists. McGraw-Hill Inc., 1991.
18. Liberty, J. Teach yourself C++ in 21 Days: [Електронний документ]. – (http://www.cma.zdnet.com/book/c++). Перевірено 26.02.2009.
19. Liu, S and Silverman, M. A Practical Guide to Biometric Security Technology: [Електронний документ]. – (http://www.computer.org). Перевірено 26.02.2009.
20. Math Jokes Collection by Andrej and Elena Cherkaev. : [Електронний документ]. – (http://www.math.utah.edu/~cherk/mathjokes.html). Перевірено 26.02.2009.
21. Mathematics Reference Notation: [Електронний документ]. – (http:// www.alcyone.com/ max/ reference/ maths/ notation.html). Перевірено 26.02.2009.
22. Modular Arithmetic and Fermat: [Електронний документ]. – (http://www.math.princeton.edu). Перевірено 26.02.2009.
23. O'Leary T. J., O'Leary, L. I. Computing Essentials. McGraw-Hill Companies, Inc., 1997.
24. Oxford Dictionary of Computing for Learners of English. / Pyne, S. and Tuck, A. OUP, 1996.
25. Pohl, I. C++ by Dissection. Addison-Wesley, 2002.
26. Simple Cipher: [Електронний документ]. – (http://www.pbs.org/wgbh/nova/decoding/simpwave.html). Перевірено 26.02.2009.
27. Software Testing: [Електронний документ]. – (http://en.wikipedia.org/wiki/Softwaretesting). Перевірено 26.02.2009.
28. The ABBYY Lingvo 11. ABBYY Software, 2005.
29. The Five Generations of Computers: [Електронний документ]. – (http://www.webopedia.com/DidYouKnow/Hardware_Software/2002/Five Generations.asp). Перевірено 26.02.2009.
30. The Hackers' Dictionary of Computer Jargon: [Електронний документ]. – (http://www.outpost9.com/reference/jargon/jargon_toc.html). Перевірено 26.02.2009.

31. The Penguin Dictionary of Mathematics. Penguin Books, 1989
32. Ulam, S. Stefan Banach. [Електронний документ] – (http://www-history.mcs.st-andrews.ac.uk/HistTopics/Scottish_Book.html) Перевірено 26.02.2009.
33. Why is there no Nobel Prize in Mathematics? [Електронний документ] – (http://almaz.com/nobel.html). Перевірено 26.02.2009.
34. Комп'ютерний словник. Пер. з англ. В. О. Сапожкова. – Київ: Україна, 1997.

Sources of Artwork

1. http://attend.to/zzz
2. www.officiement.com
3. www.256.gr/smiles

НАВЧАЛЬНЕ ВИДАННЯ

Винник Ольга Юріївна

АНГЛІЙСЬКА МОВА
для програмістів і математиків

Редактор Оксана Чернигевич
Технічний редактор Лілія Саламін
Комп'ютерне верстання Ольги Винник
Художник-дизайнер Уляна Келеман

Здано у видавництво 16.10.2008. Підписано до друку 28.02.2009.
Формат 60×90/16. Папір офсетний. Друк на різографі.
Умовн. друк. арк. 11,5. Обл.-вид. арк. 8,5.
Дод. наклад 150 прим. Зам. 100719.

Видавництво Національного університету "Львівська політехніка"
Реєстраційне свідоцтво серії ДК № 751 від 27.12.2001 р.

Поліграфічний центр Видавництва
Національного університету "Львівська політехніка"

вул. Ф. Колесси. 2. Львів. 79000