Jolanta Maciejewska
Anna Kucharska-Raczunas

INFORMATION TECHNOLOGY

for students of technical studies

GDAŃSK 2012

PRZEWODNICZĄCY KOMITETU REDAKCYJNEGO
WYDAWNICTWA POLITECHNIKI GDAŃSKIEJ
Romuald Szymkiewicz

RECENZENT
Iwona Gajewska-Skrzypczak

PROJEKT OKŁADKI
Katarzyna Olszonowicz

Wydano za zgodą
Rektora Politechniki Gdańskiej

Oferta wydawnicza Politechniki Gdańskiej jest dostępna pod adresem
http://www.pg.gda.pl/wydawnictwo/oferta

ISBN 978-83-7348-443-6

SPIS TREŚCI

Contents

Introduction

"English for Information Technology" is a coursebook for professional English in the field of IT. The coursebook aims to provide students of technical faculties with challenging, professional material which extends their range of vocabulary, and can be used during English language courses at university (level B2 and higher). All the texts were prepared by staff of the Faculty of Electronics, Telecommunications and Informatics at Gdansk University of Technology. The texts cover a wide range of contemporary specialist issues. The language used is that of scientific papers, which will enable IT students to feel more confident when preparing mid-term projects and writing their theses. Moreover, "English for Information Technology" ensures that students will be able to function effectively in an English-speaking environment in their future work.
All the units contain phonetic exercises which allow students to practise their pronunciation of technical vocabulary. Listening comprehension is also developed through numerous recordings, requiring understanding of challenging content. This prepares students for studying abroad on various kinds of scholarships, as well as for using online lectures available from the Internet.
The course content covers approximately 30–45 hours of teaching. The units can be chosen randomly, depending on the students' interests and needs. The jokes presented before each unit can be used as warm-up exercises, subject for a lead-in discussion, or pure comprehension exercises.

All the recordings (in MP3 format) can be downloaded from the websites of **the Foreign Languages Centre (CJO)** and from **the Faculty of Electronics, Telecommunications and Informatics of Gdańsk University of Technology.**
The texts were provided by the following members of the Faculty of Electronics, Telecommunications and Informatics:
Zbigniew Czaja, Andrzej Chybicki, Marek Blok, Piotr Dalka, Marcin Kulawiak, Jakub Miler, Aleksander Jarzębowicz, Paweł Raczyński, Jan Daciuk, Krzysztof Goczyła, Marek Kubale, Mariusz Szwoch, Paweł Kaczmarek.
The authors recommend the use of "**Additional Material**", which gives extra language practice. Furthermore, studying the biographies awakens students' interest in the lives and achievements of contributors to the field of Information Technology.

Acknowledgements

The authors would like to express their thanks to Professor Marek Moszyński for his invaluable advice during their work and for co-ordinating the project.
We would also like to express our appreciation and gratitude to the authors of the texts for sharing their materials and the results of their work with us, and for their friendly co-operation.
The coursebook was funded by the Faculty of Electronics, Telecommunications and Informatics at Gdańsk University of Technology in the Faculty's 60th anniversary year.

٩(̾●̮̮̃̾•̃̾)۶ ٩(̾●̮̮̃̾•̃̾)۶ ٩(̾●̮̮̃̾•̃̾)۶

On the keyboard of life, always keep one finger on the escape button.

٩(̾●̮̮̃̾•̃̾)۶ ٩(̾●̮̮̃̾•̃̾)۶ ٩(̾●̮̮̃̾•̃̾)۶

Unit 1

Microcontrollers

LEAD-IN:

Ex. 1

Match the words 1–12 with their definitions:

1. VOLATILE	a. A communications path in a network. 2. A complete path through which an electric current can flow.
2. ENTITY	b. An apparatus for generating electric waves in a system of wireless telegraphy.
3. MICROCONTROLLER	c. The connections between and within the CPU, memory and peripherals used to carry data.
4. PERIPHERAL	d. Near the surface or outside of; external.
5. ACTUATOR	e. Something that exists as a particular and discrete unit.
6. CIRCUIT	f. A mechanism that puts something into automatic action.
7. OSCILLATOR	g. Not retaining stored information when the power supply is cut off; lasting only a short while, changeable.
8. DATA BUS	h. A microprocessor on a single {integrated circuit} intended to operate as an embedded system.
9. CHIP	i. The act or process of keeping information in a computer memory or on a magnetic tape, disk
10. ANALOGUE	j. A device in which data are represented by continuously variable, measurable, physical quantities.
11. LOOP	k. A tiny slice of material, generally in the shape of a square a few millimeters long, cut from a larger wafer of the material, on which a transistor or an entire integrated circuit is formed.
12. STORAGE	l. A closed electric or magnetic circuit through which a signal can circulate.

Ex. 2 ***T1***

Listen and repeat after the recording:

microcontroller, circuit, periphery, oscillator, to incorporate, the former, the latter, to facilitate, volatile, open-loop, closed-loop, address bus, data bus, to adhere to, to retain, entity, leakage current, actuator, inbound data

Ex. 3

Discuss the following questions with your partner:

1. What is a microcontroller?
2. What are the most common applications of microcontrollers?
3. What is a microprocessor?
4. What does semiconductor memory store?
5. What does data memory store?

TEXT-BASED TASKS:

Ex. 4

Read the text "Microcontrollers" and find the answers to the questions from Exercise 3. Discuss your answers with your partner.

MICROCONTROLLERS

We can define a microcontroller in the following way:

1. It is a digital electronic circuit with a specialized microprocessor and with necessary devices for its independent work placed in one chip.
2. Hence, it is able to work autonomically, that is it doesn't need external devices (e.g. bus controllers, system clock generators, reset circuits).
3. It was designed to work in control-measurement systems and communication systems, therefore it owns a complex communication system to connect with the surroundings.
4. It works mostly in real time.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, medical devices, remote controls, office machines, appliances, power tools and toys. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analogue components needed to control non-digital electronic systems.

A microcontroller incorporates the following components, all of which function interactively, as shown in Fig. 1.1:

- *Microprocessor*. This functions as the central processing unit (CPU). The microprocessor contains its own control unit and arithmetic and logic unit (ALU). The latter unit handles arithmetic and logical operations, whereas the former ensures the execution of instructions received from the program memory. This division of labour facilitates adaptation to a variety of practical applications through appropriate programming.
- *Input and output modules* (*I/O modules*). These handle the data transfer with the periphery. This includes input/output devices, circuits for controlling interrupts of a program, and bus systems carrying communications with other control devices, such as CAN.

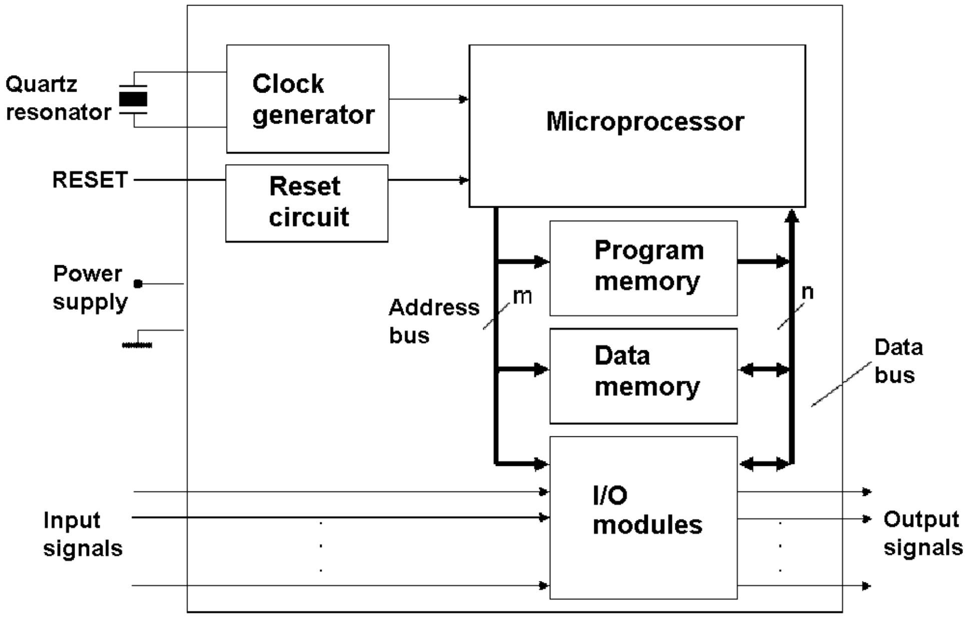


Fig. 1.1. Diagram of the microcontroller

- *Program and data memory.* This is non-volatile, permanent storage holding the program, such as the open-loop and closed-loop control algorithm, and the constant parameter sets, such as open-loop and closed-loop control parameters. Ideally suited to this task are the non-volatile memory technologies. This memory is often organized in such a way that the programme and the associated parameter sets are stored in separate memory segments. Accordingly, the term *program and data memory* is used.
- *Data memory.* This stores the data that are changed as a consequence of program execution. Because of its special characteristics, this memory segment is also termed *random access memory* (RAM). Ideally suited to this task are read/write memory technologies. Depending on the requirements of the application, volatile or non-volatile read/write memory is used.
- *Bus system.* This system interconnects the individual microcontroller components. It consists of an *address bus* and a *data bus*.
- *Clock generator.* Also known as an oscillator, this device ensures that all operations taking place within the microcontroller adhere to the same clock frequency.
- *Reset circuit.* This is used to introduce the microcontroller in an initial known state.

Semiconductor memory of the microcontroller is used to store the following:

- Data, such as I/O data, states, and intermediate results that often require rapid read and write access.
- The executable program, which, in most cases, requires permanent storage.
- Constant parameter sets, which also require permanent storage in many cases.

Ex. 5

Work in pairs. One student reads the definition of the following terms from the text above in random order:

microprocessor, *input and output modules*, *program and data memory*, *data memory*, *bus system*, *clock generator*, *reset circuit*

The other student gives the term that corresponds to each definition. Then change roles.

Ex. 6

Work in pairs. Discuss what the following abbreviations mean:

CPU, ALU, I/O, RAM, SRAM, DRAM, ROM

When you have finished, read the text below and check your answers.

Depending on their individual application, semiconductor memory uses either bit or word oriented organization, where "word" describes the logical collection of bits suitable for parallel processing by the microcontroller. Thus, the *word length* is equal to the number of bits being processed in parallel. Microcontrollers customarily accommodate word lengths of 8, 16, 32 or 64 bits. A group of 8 bits is termed a *byte* (1 byte = 8 bits).

Data memory bases on the *random access memory* (RAM), which provides direct access to any main memory location. Information can be read from and written to RAM as often as desired. The main memory uses volatile RAM, meaning that the memory contents will be lost in the event of a failure in operating power. In random access memory, a differentiation is made between static RAM (abbreviated as SRAM) and dynamic RAM (abbreviated as DRAM).
Static RAM is written to only once and retains its memory contents as long as there is a working voltage present. Because leakage currents would cause the memory contents of dynamic RAM to be lost over time, these must be refreshed periodically.

Program memory uses the *Flash* EPROM (sometimes simply termed Flash) which comprises the next development level of the EPROM and EEPROM. This memory allows the erasure, or flashing, of entire memory areas or complete memory contents through the application of electrical pulses. Once erased, the affected areas can be reprogrammed. Flash memory programming can be accomplished with the use of a programming unit. However, the decisive advantage of Flash technology lies in the fact that the Flash memory allows in-system reprogramming with the use of a reprogramming tool, even while contained in an ECU installed onboard a vehicle. For this reason, Flash technology is applied in situations where relatively large volumes of data must be committed to non-volatile storage but possibly require modification in the course of the product life cycle (e.g. serving as program or data memory in ECUs).

The *microprocessor* comprises the programmable entity handling the addressing and manipulation of data, as well as the control of the time-specific and logical execution of a program. The various memory areas provide storage for data and program instructions.

A read/write access type of memory (e.g. RAM) is required to provide storage for variable data. The memory type suited to the tasks of storing program instructions and permanent data is read-only memory (e.g. ROM). Most microprocessors also contain a small, integrated memory module holding the so-called *registers*. Their purpose is to provide rapid read and write access.

Ex. 7

Work in pairs. Write 6 true/false sentences based on the information in the text.
Make groups of four. Quiz each other.
Check your answers with the text.

FOLLOW-UP EXERCISES:

Ex. 8

Work with your partner. Decide who is A, who is B. Find Appendix 1A, 1B at the back of the book. Ask each other questions to find the missing information.

Ex. 9 ***T2***

You are going to hear a description of 4 different operating modes. Listen and note down the names of the four operating modes described in the recording. Compare your notes with your partner.

Ex. 10 ***T2***

Listen again and write down some details of the operating modes. Compare your notes with your partner. When you have finished, check your notes with the tapescript for this unit at the back of the book.

Ex. 11

Work in pairs. Without looking at your books, Student A – describe one of the operating modes and its main characteristics, without mentioning its name. Student B – listen and guess which mode is being described. Change roles.

Ex. 12

Work in groups of four. Prepare a description of another application of microcontrollers. Present it to the whole group.

USEFUL PHRASES	
It is able to work …	It was designed to work…
It is used to…	It incorporates the following components…
It functions as…	It contains…
It includes…	It consists of…
It is based on…	It comprises…

٩(̃-̮̃-)۶ ٩(̃-̮̃-)۶ ٩(̃-̮̃-)۶

Where Do Deleted Characters Go?

QUESTION: Where do the characters go when I use my backspace or delete them on my PC?

ANSWERS: The characters go to different places, depending on whom you ask:

The Catholic Church's approach to characters: the nice characters go to Heaven, where they are bathed in the light of happiness. The naughty characters are punished for their sins. Naughty characters are those involved in the creation of naughty words, such as "breast," "sex" and "contraception".

The Buddhist explanation: if a character has lived correctly and its karma is good, then, after it has been deleted, it will be reincarnated as a different, higher character. Those funny characters above the numbers on your keyboard will become numbers, numbers will become letters, and lower-case letters will become upper-case.

The 21st-century bitter cynical nihilist explanation: who cares? It doesn't really matter if they're on the page, deleted, undeleted, underlined, etc. It's all the same.

The Mac user's explanation: all the characters written on a PC and then deleted go straight to PC hell. If you're using a PC, you can probably see the deleted characters, because you're in PC hell also.

Stephen King's explanation: every time you hit the (Del) key you unleash a tiny monster inside the cursor, who tears the poor unsuspecting characters to shreds, drinks their blood, and then eats them, bones and all. Hah, hah, hah!

IBM's explanation: the characters are not real. They exist only on the screen when they are needed, as concepts, so to delete them is merely to de-conceptualize them. Get a life.

PETA's (People for the Ethical Treatment of Animals) explanation: you've been DELETING them??? Can't you hear them SCREAMING??? Why don't you go CLUB some BABY SEALS while wearing a MINK, you pig!!!!

٩[̲̅-◌-)۶ ٩[̲̅-◌-)۶ ٩[̲̅-◌-)۶

Unit 2

Mobile Technologies and Their Influence on Our Life

LEAD-IN:

Ex. 1

Match the words 1-12 with their definitions:

1. TO LAUNCH	a.	Added to or made greater in amount or number or strength.
2. ACCELEROMETER	b.	Existing or occurring together.
3. AUGMENTED	c.	A list of names, addresses, and other data.
4. CONCOMITANT	d.	The choice of roads taken to get to a place.
5. QWERTY	e.	A radio transmitter that emits a characteristic guidance signal.
6. DIRECTORY	f.	The standard English language typewriter keyboard layout.
7. ROUTE	g.	An instrument for measuring acceleration.
8. BEACON	h.	To set going; initiate.
9. MOBILE	i.	Moving or capable of moving readily.
10. INEGRATED	j.	To activate or start (a program, for example).
11. SPECIFICATION	k.	A detailed, exact statement of particulars, prescribing materials, dimensions, and quality of work for something to be built, installed, or manufactured.
12. TO INVOKE	l.	Formed into a whole or introduced into another entity.

Ex. 2 ***T3***

Repeat the following words after the recording.

ultimate, *primarily*, *functionality*, *alternative*, *to launch*, *accelerometer*, *accessible*, *to submit*, *to enable*, *augmented*, *concomitant*, *qwerty*, *innovative*, *utilization*, *infrastructure*, *particular*

TEXT-BASED TASKS:

Ex. 3

Read the "Introduction" and find the answers to the following questions:

1. What does PDA mean?
2. What role will the mobile phone have in the future?
3. What do currently released mobile phones include?
4. What multi-media formats does HTC7 allow to be played?
5. What devices are meant for business users?
6. What additional gadgets can smartphones include?

INTRODUCTION

The remarkable and concomitant progress in mobile technologies, together with the increase the wide-band Internet access, has revolutionized the way people communicate with each other. An increasing number of mobile phones and personal digital assistants (PDA) allow people to get information from any place – home, office, work or bus and share it with others. At present, there are estimated to be over 1.5 billion mobile phones with the processing power of a low-class personal computer (PC) in the world today. The range of computer-like functionalities offered by mobile devices is leading some observers to speculate that many people in the not so distant future will start to see the mobile phone as an alternative to a PC. Above all, the mobile devices are likely to become part of our life delivering several social, business, or spare time related services.

Currently released modern phones include some games, tools for social communication (e.g. Facebook, twitter) and enable us to download additional applications for other purposes. In order to satisfy customer demands, manufacturers deliver dedicated mobile phone models such as multimedia phones, business phones or smartphones targeted for gaming, Internet browsing etc.

The ultimate example of this trend is the HTC 7 Surround (see Fig. 2.1), which is primarily a portable multimedia machine but can also be used as a phone. It is equipped with sideout speakers with Dolby Mobile and SRS surround sound and integrated with Facebook® and Windows Live™ accounts. It also supports Music and Videos Hub powered by Zune®, which makes it possible to listen to music and radio and provides tools for playing multimedia formats such as .m4a, .m4b, .mp3, .wma (Windows Media Audio 9).

On the other hand, some devices aimed at business users, include a physical qwerty keyboard, large screen, push e-mail applications and are marketed primarily as business communication tools. Examples are the BlackBerry 8900 and the Nokia E63 messaging devices (see Fig. 2.2).

Fig. 2.1. HTC 7 Surround Smartphone

For a whole branch of users third generation GSM networks (3G) also allow the viewing of video content including music videos and football game highlights in VOD (video on demand) model service. With several new 3G services being launched, the 3G phones are expected to become more popular. However, it seems that the next big thing in mobile phones may be television. TV phones have stabilized their position in high-tech developed countries like Japan or Korea.

Moreover, currently released specifications force smartphone manufacturers to include several sensors and additional gadgets such as digital video cameras, accelerometers, media players, mobile internet browsers, light sensors, GPS and others.

Fig. 2.2. BlackBerry 8900 and Nokia E63 smartphones

Fig. 2.3. Augmented reality example. *Source*: Google Inc.

Ex. 4

Check the answers with your partner and then with the whole group. Refer back to the text if necessary.

FOLLOW-UP EXERCISES:

Ex. 5

Close your books. Work in groups of three and brainstorm ideas for possible applications of mobile technologies. Then compare your ideas with another group.

Ex. 6

Look at Appendix 2 at the back of the book. Read the text and, with your partner, agree on the most useful application. Prepare three arguments supporting your choice. Speak to another pair and discuss your arguments.

Ex. 7 *T4*

You are going to hear someone talking about some LBS applications. Listen to the recording and note down the functions of these applications:

- emergency services,
- navigation services,
- augmented reality.

Check your answers with your partner, and then with the whole group.

Ex. 8

Rearrange these sentences into a logical text:
When you have finished, check with your partner.

The future of the sector is focused on the development of fast wireless communication infrastructure and steady growth of technologies providing tools and services for compact devices.

One of the most popular applications of mobile technologies are so called location-based services. This particular advantage makes LBS capable of delivering specific information in accordance with what is specifically desired by the user.

Smartphones, hand-held devices and PDA are becoming part of our everyday life, making digital information accessible every time and everywhere.

They are different from more conventional paper and internet based media (guides, directories, maps etc.) because they are aware of the context in which they are invoked.

Many of the solutions presented in the paper are already provided for users worldwide, including business activities, weather information, traffic services and so on.

Ex. 9

Choose ten words from the unit which you would like to learn or find useful. Write a list, but do not show it to your partner. Prepare an article on the subject of mobile technologies, including your ten words. Tell your partner your article. Your partner will listen and write the words which he/she thinks are from your list. When you have finished compare your lists. Change roles.

Ex. 10

Read the text below and insert the FUTURE FORMS of the words given in brackets:
Future of smartphone

The next generation of smartphones 1) ……………….. (be) context-aware, taking advantage of the growing availability of embedded physical sensors and data exchange abilities. One of the main features applying to this is that the phones 2)……………….. (start) keeping track of your personal data, but adapt to anticipate the information you 3)……………….. (need) based on your intentions. There 4)……………….. (be) all-new applications coming out with the new phones, one of which is an X-Ray device that reveals information about any location at which you point your phone. One thing companies are developing is software to take advantage of more accurate location-sensing data. They describe it as wanting to make the phone a virtual mouse able to click the real world. An example of this is where you can point the phone's camera while having the live feed open and it 5)………………. (show) text with the building and save the location of the building for future use.

Ex. 11

Design your dream mobile phone. Make notes. When you have finished, work in groups of three and describe your phone to your partners. Decide which of your phones is the best.

USEFUL PHRASES	
It includes…	It is powered by…
It contains…	It supports…
It is integrated with…	It is capable of…

(-̮̃◌◌̮◌•)۶ (-̮̃◌◌̮◌•)۶ (-̮̃◌◌̮◌•)۶

Women are like a computer virus...they ENTER your life...SEARCH your pocket...SHIFT your balance ...CONTROL your life...and when you become an old version, DELETE you from the system.

(-̮̃◌◌̮◌•)۶ (-̮̃◌◌̮◌•)۶ (-̮̃◌◌̮◌•)۶

Unit 3

Information Systems

LEAD-IN:

Ex. 1

Match the words 1-12 with their definitions:

1. COAXIAL	a. The even or odd quality of the number of 1's or 0's in a binary code, often used to determine the integrity of data especially after transmission.
2. PARITY	b. Change; deformation.
3. CRYPTOGRAPHY	c. Rapid growth or increase in numbers, intensification.
4. DISTORTION	d. Having or being mounted on a common axis.
5. ENCRYPTION	e. The activity of converting data or information into code.
6. PROLIFERATION	f. A measure of the efficiency of a system, such as a code or language, in transmitting information.
7. STEGANOGRAPHY	g. The branch of cryptography in which messages are hidden inside other messages; used commonly for the process of hiding messages inside a computerized image file.
8. ENTROPY	h. The science or study of secret writing, especially code and cipher systems.
9. RECIPIENT	i. Impossible or nearly impossible to see or notice.
10. FLUCTUATION	j. A person who acquires something; one to whom something is addressed.
11. BANDWIDTH	k. Constant change; instability.
12. IMPERCEPTIBLE	l. The amount of data that can be passed along a communications channel in a given period of time.

Ex. 2 *T5*

Repeat the following words after the recording.

recipient, *qualitative*, *quantitative*, *coaxial*, *fluctuation*, *digitised*, *bandwidth*, *discrete*, *entropy*, *bit*, *parity*, *cryptography*, *to intercept*, *encryption*, *proliferation*, *metadata*, *imperceptible*, *steganography*

Ex. 3

In pairs discuss the difference between message and information. Consult with another pair when you have finished.

TEXT-BASED TASKS:

Ex. 4 ***T6***

Listen to the recording and check your ideas from Ex. 3.

Ex. 5 ***T6***

Listen again and fill in the gaps in the following sentences.
When you have finished check your answers with your partner/teacher, and then with the tapescript for this unit.

1. The goal of the Information System is to ………………………… from the sender (source) to the recipient (destination)…
2. Information Theory is the science which specifies ………………………… by basic blocks of the information/communication system.
3. In his paper Shannon for the first time introduced the qualitative and quantitative model of communication ………………………… .
4. Everybody would agree that information is transferred ……………………….. only if the recipient of the information did not already have the information to begin with.
5. Messages that convey information that is certain to happen and already known by the recipient contain ………………………… .
6. Note that information measure has nothing to do with …………………………, as the messenger should not look into the message he carries, analysing if it is important or not.

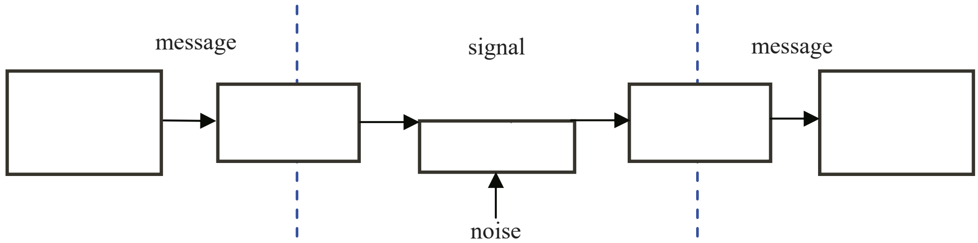
Ex. 6

Work in pairs. Student A read "Information Systems", Student B read "Information Systems and Coding." While reading, note down the subsequent steps of the model of an information system.

Student A:

When you have finished, describe the steps to your partner so that they can fill in the model in Figure 3.1.

Figure 3.1 for text A



TEXT A

INFORMATION SYSTEM

The general model of an information system originally proposed by the C. E. Shannon is presented in Fig. 3.1. If the components are appropriately interpreted then this model can be applied to any system in which information transfer occurs, e.g. conversations between humans or Internet transmissions. The information system model consists of five components: information source, transmitter, channel, receiver and destination. The information source produces a message or sequence of messages to be communicated to the destination. The transmitter operates on the message in some way to produce a signal suitable for transmission over the channel. The channel is merely the medium used to transmit the signal from transmitter to receiver, e.g. a coaxial cable used to transfer electrical signals or a piece of paper used to transfer written messages. The receiver usually performs the inverse operation of that done by the transmitter, reconstructing the message from the signal. And finally the destination is the person (or thing – e.g. computer, robot or animal) for whom the message is intended. For example, as we speak our mind generates messages in the form of words. Our voice box converts the words into signals represented by fluctuations in air pressure. Those variations in pressure are carried by the channel, which is the air present in the room. On the other side of the room our audience's ears receive the signal from the channel and we hear sounds which our brain converts into words.

Student B:

Describe the steps to your partner so that they can fill in the model in Figure 3.2.

Figure 3.2 for text B

TEXT B

INFORMATION SYSTEMS AND CODING

As everybody who has played the children's game "Chinese whispers" knows, efficient and reliable message transfer is not easy. Information system inefficiency begins at the information source. Messages are often redundant and transferring them in their original form wastes system resources. For example, when we speak, not all the words uttered are necessary for the listener to understand the message. Sometimes we already know parts of the message or the message itself is irrelevant to the listener. In other words, for efficient information transfer the message should be compressed and this is the task for source coding. Source coding takes a digitised message and tries to find the shortest code word representing that message. The message coded in this way requires the smallest transmission time and channel bandwidth. The problem is that source decoding relies on ideal transmission through a discrete channel.

In practical systems errors are unavoidable because of distortions and disturbances occurring in the channel. Therefore, if information has to be transferred reliably, additional measures have to be taken. The redundant symbols are inserted between output symbols of the source coder. These additional symbols allow the channel decoder to detect or correct errors which occurred in the channel. Error correction coding and error detection coding are collectively named error control coding. Error control coding (ECC) is present in nearly every piece of equipment we use in modern, information-based society. As examples we can cite CD or DVD discs, which employ error correction codes to protect the data embedded in the plastic disk from surface scratches, hard disk drives, digital cellular phones, packets of data transmitted over the Internet, integrated circuits whose proper work is internally secured by error control coding and many, many others. Finally, the coded message must be converted into a signal suitable for transmission over the channel with the use of a modulator.

Ex. 7

Read the other text to check the way you completed the model. Then check the figures in the key for this unit.

Ex. 8

Read the text below and put in the following words in the passive:

call, *represent*, *code*, *present*, *use* (twice), *reserve*, *need*

SOURCE CODING – THE HUFFMAN CODE

If an engineer wants to convert an inefficient source of information into an efficient one, he applies source coding. Generally speaking, the rule of entropy source coding is that the length of each code word should be approximately proportional to the information content of the message which 1) ……………….. by that code word. Therefore, the most common symbols use the shortest codes, while the longest code words 2) ……………….. for the least frequent messages. The most common entropy coding technique is Huffman coding. The Huffman code is a variable-length code with a specific method for choosing the representation for each symbol, resulting in a prefix code (which 3) …….. sometimes ………. "prefix-free code"), that is the code word representing some particular symbol is never a prefix in the code word representing any other symbol.

Table 3.1 shows an example of the Huffman code for source generating eight symbols from A to H with given probabilities. If the code with constant code word length 4) ……………….., three bits per symbol 5) ……………….., while for the Huffman code the average code length is only 2.46 bits per symbol. An example of the first twelve output symbols of that source 6) ……………….. in Table 3.2. If the Huffman code from Table 3.1. 7) ……………., this message 8) ……………….. with 29 bits, giving 2.42 bits per symbol. This value is very close to an average code length. However, in practice short source symbol sequences can vary from the expected value.

Table 3.1.

Example of Huffman code with the coding tree

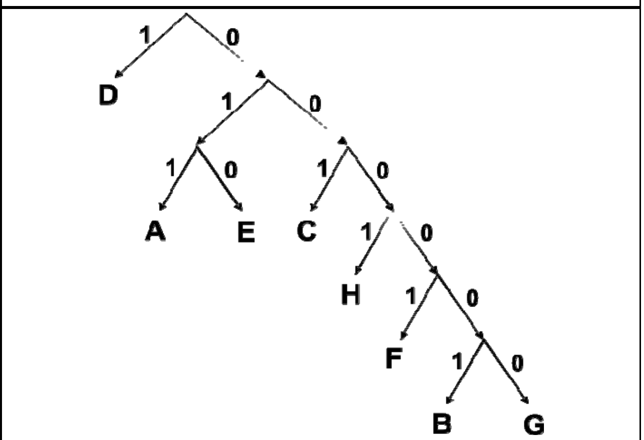
x_i	$P_X(x_i)$	code word	coding tree
A	0.18	011	
B	0.01	000001	
C	0.09	001	
D	0.40	1	
E	0.17	010	
F	0.06	00001	
G	0.01	000000	
H	0.08	0001	

Table 3.2.

Messages generated by the source and corresponding output string of the source coder from Tab. 3.1

source output:	D	A	E	D	C	D	A	E	F	D	H	D	...
source coder output:	1	011	010	1	001	1	011	010	00001	1	0001	1	...

Ex. 9

Read the text below. Work in pairs. Prepare 5 true/false sentences to the text. When you have finished, work with another pair and quiz each other.

CHANNEL CODING – THE HAMMING CODE

One of the most commonly used channel codes is the Hamming code invented by Richard Hamming. This code transforms blocks of k information bits into code words of the length n by adding $k - n$ parity bits, where $n = 2^{k-n}-1$. The minimum Hamming distance, difference in bits between any two code words, for the Hamming code is 3. This means that Hamming codes can correct single-bit errors and reliable communication is possible when the Hamming distance between the transmitted and received bit patterns is less than or equal to one.

For example, the Hamming code (7,4) uses 7-bit code words with 4 information bits and 3 parity bits and it is described by the following set of generation equations:

$$\begin{cases} c_1 = c_3 \oplus c_5 \oplus c_7 \\ c_2 = c_3 \oplus c_6 \oplus c_7 \\ c_3 = x_1 \\ c_4 = c_5 \oplus c_6 \oplus c_7 \\ c_5 = x_2 \\ c_6 = x_3 \\ c_7 = x_4 \end{cases}$$

where $\oplus$ represents modulo 2 summation, x_i – information bits and c_i – code word bits. Additional parity bits are inserted in the first, second and fourth position. For example, data bits 1000 are represented by code word 1110000 (see Tab. 3.3) which is transmitted to the receiver.

Table 3.3.

Examples of code words for the Hamming code (7,4)

c_1	c_2	$c_3 = x_1$	c_4	$c_5 = x_2$	$c_6 = x_3$	$c_7 = x_4$
1	**1**	1	**0**	0	0	0
1	**0**	0	**1**	1	0	0
0	**1**	0	**1**	0	1	0
1	**1**	0	**1**	0	0	1

The channel decoder for each received block must determine which code word was most likely transmitted. This can be achieved with the help of a syndrome vector that identifies the most likely error pattern. For the Hamming code (7,4) the syndrome bits are computed using parity check equations determined from the generation equations:

$$\begin{cases} s_1 = c_1 \oplus c_3 \oplus c_5 \oplus c_7 \\ s_2 = c_2 \oplus c_3 \oplus c_6 \oplus c_7 \\ s_3 = c_4 \oplus c_5 \oplus c_6 \oplus c_7 \end{cases}$$

If the decoder receives the string 1110010, the computed non-zero syndrome 011 indicates that an error occurred during transmission. Additionally, for this code the reversed syndrome vector is the binary number pointing at error position. In our case $110_2 = 6$ and after error correction at the sixth position we obtain the most probable transmitted code word 1110000 with information bits 1000.

FOLLOW-UP EXERCISES:

Ex. 10 ***T7***

You are going to listen to a recording about coding messages. Listen and answer the following question in your own words:

What is cryptography?

Ex. 11 ***T7***

Listen again and answer the following questions:

1. What do some users of channel coding need?
2. How long has cryptography been studied?
3. When was cryptography especially important?
4. What was Enigma?
5. How did Enigma work?
6. Who was the Enigma system broken by?
7. Who was the work continued by?
8. What was another cyphering method brought into use during WWII?
9. Can the OTP be broken?
10. Who was this fact proved by?

When you have finished, check your answers with your partner, and then with the whole group.

Ex. 12

With your partner decide who is A and B.
Sit in two groups: A and B.
Students A look at Appendix 3A
Students B look at Appendix 3B
Read your texts. In your group prepare questions to find out the missing information.
Your teacher will check if the questions are correct.

Ex. 13

Come back to your original partners. Ask and answer the questions in pairs.
Read the whole text to check whether you were right.

٩(×̯×)۶ ٩(×̯×)۶ ٩(×̯×)۶

Smoking

A man is smoking a cigarette and blowing smoke rings into the air. His girlfriend becomes irritated with the smoke and says, “Can’t you see the warning on the cigarette packet? “Smoking seriously damages your health”.

To which the man replies, “I am a programmer. We don’t worry about warnings; we only worry about errors.”

٩(×̯×)۶ ٩(×̯×)۶ ٩(×̯×)۶

Unit 4

LipMouse – Novel Multimodal Human-Computer Interaction Interface

LEAD-IN:

Ex. 1

Match the words 1-12 with their definitions:

1. ITERATION	a. A facial expression in which the lips are tightly pulled together and pushed outward.
2. TO ATTENUATE	b. A computational procedure in which a cycle of operations is repeated, often to approximate the desired result more closely.
3. PUCKERED	c. The strength at which a stimulus is perceived.
4. THRESHOLD	d. The quality of never making an error.
5. INCONVENIENCE	e. The point of interaction or communication between a computer and any other entity.
6. RELIABILITY	f. Relating to a nerve or the nervous system.
7. INTERFACE	g. The quality of not being useful or convenient.
8. NEURAL	h. To reduce in force, value, amount, or degree; weaken.
9. INTUITIVE	i. The condition of being unable to perform as a consequence of physical or mental unfitness.
10. DISABILITY	j. Produced by man; not occurring naturally; made in imitation of a natural product, esp. as a substitute; not genuine.
11. CALIBRATION	k. Spontaneously derived from or prompted by a natural tendency.
12. ARTIFICIAL	l. The act of checking or adjusting (by comparison with a standard) the accuracy of a measuring instrument.

Ex. 2 ***T8***

Repeat the following words after the recording:

intuitive, *disabilities*, *utilize*, *acceleration*, *parameters*, *threshold*, *calibration*, *iterative*, *to attenuate*, *label*, *inconvenience*, *gesture*, *algorithm*, *reliability*, *efficient*, *sufficient*, *neural*, *neutral*, *crucial*, *puckered*, *Artificial Neural Network* (*ANN*)

TEXT-BASED TASKS:

Ex. 3 ***T9***

You are going to hear someone talking about a new IT solution. Listen to the recording and answer the question:

What is HCI?

Ex. 4 ***T9***

Listen again. What are the two main types of HCI solutions and what are their characteristics? Consult your answer with your partner, and then with the whole group.

Ex. 5

Read the text below and find the answer to the following questions:

1. What is the total effectiveness of the lip gesture recognition?
2. How many recognition errors appear every two seconds?
3. How many times was each person asked to perform a typical calibration procedure?
4. How many face recordings were collected to determine the effectiveness of the lip gesture recognition?
5. When is ANN trained?
6. When is the ANN able to recognize lip gestures made by the user during current session?
7. What is used by the ANN to classify gestures made by the user?
8. Who are the target users of LipMouse?
9. What is the purpose of calibration?
10. What is the main task of the LipMouse?

INTERFACE DESCRIPTION

LipMouse is a name of the patent-pending, contactless, vision-based, human-computer interface that enables the user to work on a computer using movements and gestures made with his/her mouth only. LipMouse is an application running on a standard PC computer. It requires only one hardware component: a display-mounted, standard web camera that captures images of the user's face. The main task of the LipMouse is to detect and analyse images of the user's mouth region in a video stream acquired from a web-camera. All movements of mouth (or head) are converted to movements of the screen cursor. Various parameters regarding threshold, speed and acceleration of the cursor movement may be set according to the user's preferences.
LipMouse also detects three mouth gestures: opening the mouth, sticking out the tongue and forming puckered lips (Fig. 4.1). Each gesture may be associated with an action, which may be freely chosen by a user. Possible actions include clicking or double-clicking various mouse buttons, moving mouse wheel – both horizontally and vertically, and others. Many actions may be defined as single or continuous ones. Single actions are executed only once,

in the very moment when a new gesture is detected; continuous actions are executed as long as the gesture considered is performed. For example, opening mouth gesture may be associated with an action executing a single left mouse button click in the moment, when the mouth is opened, or with an action that keeps the left mouse button pressed as long, as a user keeps his/her mouth open.

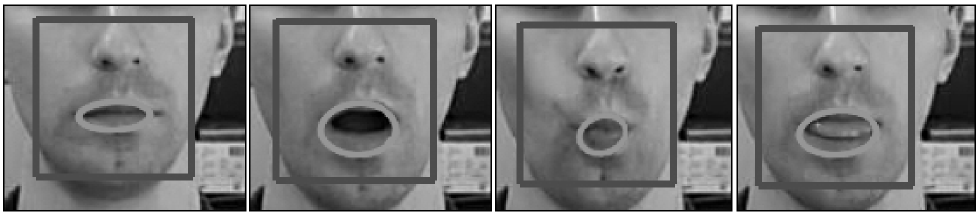


Fig. 4.1. Starting from the left: neutral face expression and three gestures detected by LipMouse: mouth opening, forming puckered lips and showing the tongue

Before the user starts working with LipMouse, a short calibration, lasting about 30 seconds, needs to be executed. During the calibration, the user is asked to perform some head movement and gestures according to the instructions seen on the screen. The purpose of the calibration is to tune LipMouse to detect gestures made by the user in the current lighting conditions.

The target users for the tool engineered are people who, for any reason, cannot or do not want to use traditional input devices. Therefore LipMouse is a solution enabling severely disabled and paralysed people to use a computer and communicate with the surrounding world. No user adaptation, such as for example placing marks on the face, is required in order to successfully work with LipMouse.

In the case of the LipMouse, lip shape and lip region image features are used by the ANN to classify gestures made by the user. The ANN is trained during the calibration stage. After the training process is completed, the ANN is able to recognize lip gestures made by the user during current session.

It is also crucial to minimize false-positive rate of detection of all three, real gestures of the LipMouse in order to prevent execution of actions not meant by the user. False-negative rate is less important – if a gesture is not recognized in some frames, it will be recognized in succeeding frames when the user moves his head a little or changes face expression. In order to minimize the number of false-positives, post-processing of the ANN output is performed in order to determine the reliability of classification. If the reliability is not sufficient, no gesture is recognized.

EXPERIMENTS AND RESULTS

In order to determine the effectiveness of the lip gesture recognition, face recordings of 102 persons were collected. Each person was asked to perform a typical calibration procedure twice. The first iteration was used to train ANN and the second iteration was used to obtain accuracy of lip gesture classification. Each video frame from the recording was classified independently. Results of this experiment are presented in Tab. 4.1.

Table 4.1.

Results of lip gesture classification experiment

Gesture	Image frames	Number of errors	Percentage of correct classifications [%]
Neutral (no gesture)	6120	310	94.9
Mouth opening	6120	463	92.4
Forming puckered lips	6120	723	88.2
Sticking out the tongue	6120	530	91.3
All gestures	24480	2026	91.7

Total effectiveness of the lip gesture recognition is over 90%, which means that on average three recognition errors appear every two seconds of the algorithm working. However, due to the advanced ANN output post-processing, the majority of errors emerge when the neutral gesture is recognized instead of other three gestures. These errors do not pose much inconvenience to the user and are attenuated further by means of simple time-averaging of lip gesture detection results.

Experiments carried out show that the effectiveness of the LipMouse algorithm is sufficient for comfortable and efficient usage of a computer by anyone who does not want or cannot use a traditional keyboard and mouse.

FOLLOW-UP EXERCISES:

Ex. 6 ***T10***

You are going to hear a description of the scheme of the LipMouse algorithm. While listening, draw the scheme. Compare with your partner.

When you have finished, check the scheme in the key for this unit.

Ex. 7

Put the following words together to make word partnerships which appear in the recording. Check with your partner.

1. vision	a) user
2. patent	b) devices
3. display	c) mounted
4. web	d) people
5. screen	e) buttons
6. mouse	f) network
7. neural	g) conditions
8. training	h) pending
9. lighting	i) set
10. target	j) camera
11. input	k) based
12. disabled	l) cursor

Ex. 8

Read the text in the Tapescript for Unit 4 and find the word partnerships.

Ex. 9

Work individually. Go through all the texts and write 10 true/false sentences.

Ex. 10

Work in groups of three. Quiz one another with your sentences.

Ex. 11

Work in groups of four. Brainstorm ideas for more computer applications, similar to the one presented in this unit. Choose the most interesting one and present it to the whole group. As a group, vote for the most interesting application.

USEFUL PHRASES:
The main task/goal of the application is…
The target users are people who…
It requires…
The solution enables…
Possible actions include…
It is also crucial to…
The purpose of …is to…

٩(̃•̮-̃)۶ ٩(̃•̮-̃)۶ ٩(̃•̮-̃)۶

Virus Alert:
To prevent the spread of the worm:

1. Don’t use the powerline.
2. Don’t use batteries either, since there are rumours that this virus has invaded most major battery plants and is infecting the positive poles of the batteries. (You might try hooking up just the negative pole.)
3. Don’t upload or download files.
4. Don’t store files on floppy disks or hard disks.
5. Don’t read messages. Not even this one!
6. Don’t use serial ports, modems, or phone lines.
7. Don’t use keyboards, screens, or printers.
8. Don’t use switches, CPUs, memories, microprocessors, or mainframes.
9. Don’t use electric lights, electric or gas heat or air-conditioning, running water, writing, fire, clothing or the wheel.

I'm sure if we are all careful to follow these 9 easy steps, this virus can be eradicated.

٩(̃•◌̮◌̃-)۶ ٩(̃•◌̮◌̃-)۶ ٩(̃•◌̮◌̃-)۶

Unit 5

Geographic Information Systems

LEAD-IN:

Ex. 1

Match the words 1-12 with their definitions:

1.MATRIX	a.	A set of bits that represents a graphic image. Each bit or group of bits corresponds to a pixel in the image.
2. GRID	b.	A set of horizontal lines composed of individual pixels, used to form an image on a CRT or other screen.
3. BITMAP	c.	A range of grey shades from white to black, as used in a monochrome display or printout.
4. RASTER	d.	A rectangular array of circuit elements usually used to generate one set of signals from another.
5. NODE	e.	A network of horizontal and vertical lines that provide coordinates for locating points on an image.
6. GRAYSCALE	f.	A type of geometry comprised of planar three-dimensional rings and triangles, used in combination to model objects that occupy discrete area or volume in three-dimensional space (spheres and cubes, or real-world objects like buildings and trees).
7. MULTIPATCH	g.	A small high-speed memory that improves computer performance.
8. CACHES	h.	An interconnection point on a computer network.
9. TO RETRIEVE	i.	The quantity of something per unit measure, especially per unit length, area, or volume.
10. DENSITY	j.	Any of a set of two or more numbers used to determine the position of a point, line, curve, or plane in a space.
11. COORDINATE	k.	The distance, measured in degrees on the map.
12. LONGITUDE	l.	Obtain from a storage device; as of information on a computer.

Ex. 2

Make sure you understand the following words. Divide them into two groups – one connected with computing, and the other one – with geography:

retrieval, *latitude*, *density*, *matrix*, *raster*, *depth*, *grayscale*, *multipatch*, *grid*, *cache*, *coordinate*, *pixel*, *elevation*, *bitmap*, *curve*, *surface*, *longitude*, *node*, *floodplains*

Work with your partner. One of you gives the definition of a word, and the other person says the word.

Ex. 3 ***T11***

Repeat the following words after the recording:

retrieval, *latitude*, *density*, *matrix*, *raster*, *depth*, *grayscale*, *multipatch*, *grid*, *cache*, *coordinate*, *pixel*, *elevation*, *bitmap*, *curve*, *surface*, *longitude*, *node*, *floodplains*

TEXT-BASED TASKS:

Ex. 4 ***T12***

You are going to listen to the recording about Geographic Information System. While listening answer the following questions (listen twice if necessary):

a) What can geographic information refer to?
b) What do the geographic coordinates consist of?
c) What is Geographic Information System?
d) What is spatial analysis?
e) Why has GIS been created?
f) What does GIS operate on?
g) What is Geodatabase?

Discuss the answers with your partner and then with the whole group.

Ex. 5

Read the tapescript at the back of the book and check your answers.

Ex. 6

Work in groups of three. Each of you will read a different text:
student A read text A "Raster",
student B read text B "Vector"
student C read text C "Triangulated Irregular Network".

While you read write 5 true/false sentences to your text. When you have finished student A – give your sentences to student B, student B – give your sentences to student C and student C – give your sentences to student A. Read the next text corresponding with the sentences you've got , decide if the sentences are true or false. Change again so that each of you reads all 3 texts and works with all true/false sentences.
Finally, discuss all your answers in your group of three. If necessary, refer back to the texts.

TEXTS

FORMATS OF GEOGRAPHIC DATA

Geographic Information Systems, in general, process three types of data:

TEXT A

RASTER

Raster data, in a broad sense, consists of a matrix of colour values, referred to as pixels. Every colour value is encapsulated in 1, 4, 8, 16, 24, 32, 48, or 64 bits. This is referred to as image bit depth. The larger the amount of bits, the more information a pixel may store, but also the bigger the size of the whole image. When the colour is represented in less than 16 bits, the image quality greatly degrades. In order to conserve space and maintain image quality, such pixels can represent grayscale or indexed colour. In case of larger bit depths, the contents of a pixel are evenly divided between red, green, blue and alpha components of a single colour. This is commonly abbreviated as RGB for red, green and blue or RGBA for all four components. The alpha channel, which contains information regarding image transparency, may also be stored in a separate grayscale bitmap called “mask”.

Raster images commonly represent satellite images or aerial photographs. However, instead of colours, the pixel values in the image matrix may also represent continuous data (e.g. height or temperature), discrete data (rainfall measurements or land use) or a null value (which means that no data is available for the specific point they represent). Each raster image is complimented with information regarding its placement on the surface of the Earth. This usually consists of the geographic coordinates of the upper left pixel of the image, the size of a pixel in geographic units (i.e. the spatial resolution of the image), as well as image rotation in the x and y axis.

The simplest example of raster data is a bitmap, however it may be stored in various formats. These include the standard file-based structure of TIF or JPEG, as well as binary large object (BLOB) data stored directly in relational database management systems. The latter, when properly indexed, typically allow for quicker data retrieval, but at the cost of convenience of data storage and exchange.

TEXT B

VECTOR

In vector format, spatial objects are represented by points, lines and polygons oriented in geographical space. Points are commonly used to describe discrete points of interest on the map, with the exact type of represented geography being determined by the scale of the map. For example, on a large scale map (e.g. of a country) single points may represent cities. On a small scale map points may represent wells or peak elevations.

Lines and polylines are most often used for linear features such as rivers, roads, railroads, trails, and topography curves. Polylines are formed by a series of interconnected points, which give the impression of smoother geometry. Lines represent distance, and can overlap or intersect one another.

Geographical features that cover a particular area, e.g. buildings or lakes, are commonly represented by polygons. Similar to lines, polygons may intersect and overlap one another. Because polygons cover an area, this functionality allows to perform simple spatial analysis e.g. of land use and may help to design e.g. the placement of perimeter fencing.

All vector data types may have non-spatial attributes, such as character strings, numbers and dates. For example, a polygon data type, which describes buildings, may contain attributes such as address, number of storeys, amount of rooms in a storey, number of residents, date of erection, date of last renovation etc. These attributes may be used to perform spatial queries, for example to quickly find all buildings which are taller than ten storeys. Vector data may also be freely styled using these attributes, for example lakes may be coloured depending on their depth.

The most common format used for storage of vector data, the Shapefile, can contain basic features, as well as their variations. The latter include multipoints and multipatches (features containing multiple points and multiple polygons, respectively), as well as three-dimensional features (similar to the regular ones, but containing height/depth data) and features containing an additional default attribute meant for storage of a user-defined measurement.

TEXT C

TRIANGULATED IRREGULAR NETWORK

Information regarding a three-dimensional model of terrain is most often represented in the form of a Triangulated Irregular Network (TIN). A TIN is a graph representation of the Earth surface (or sea bottom), where irregularly distributed nodes (with coordinates x, y and z) represent the most important changes in terrain topography. The lines connecting nearby nodes are arranged in a network of non-overlapping triangles. TINs may be produced manually or derived from the elevation data extracted from satellite images. A TIN is different from the equally popular Digital Elevation Model (DEM) in that whereas a DEM is represented by a raster (and thus has regularly distributed nodes), a TIN is essentially a list of nodes accompanied by lists of their neighbours. Because of this TIN files are much smaller than DEMs, but at the same time they are more complicated, and thus more difficult to handle.

Each type of geographic data has its advantages and disadvantages. Raster datasets are often very large and thus their manipulation requires a lot of resources. They also have a fixed resolution, so zooming in past a certain scale reveals no additional details. Vector and TIN representation of an area will virtually always be smaller than a corresponding raster image. Vector data is also easier to manipulate (e.g. reproject) and (as a de facto series of numbers) can be naturally integrated into relational databases. In addition, vector and graph data are much easier to analyse, especially in case of networks such as roads, power grids, telecommunication networks, etc. Vector data is also simpler to update and maintain (e.g. when a new building is erected), whereas in such cases a raster image needs to be completely reproduced.

Having said that, raster data is much easier to overlay or draw with partial transparency. Raster data is also more detailed than vector data at similar resolution (as it is basically a photograph). It also renders much faster. At the same time, production of vector maps requires much more human labour than generation of raster images of the same area. Therefore the combination most commonly found in a GIS is a satellite raster image for background, with layers of vector features representing the objects of interest accompanied by non-spatial attributes.

FOLLOW-UP EXERCISES:

Ex. 7

Brainstorm possible applications of GIS in groups of two.
Do not look at the texts below. Check your ideas with another pair.

Ex. 8

Read the paragraphs of the text below and rearrange them in a logical order:

WORKSTATION GIS

A. A relatively new application of workstation GIS is related to creation and visualization of three-dimensional city models. Such detailed building models are created from satellite stereo pair imagery, where the same area is photographed twice (from two adjacent angles) by a passing satellite. The resulting stereographic image is converted to 3D model data, which may be used for modelling noise, planning the placement of mobile phone antennas, as well as for the purposes of marketing and promotion of the city.

B. Marine districts also use GIS to decide how to allocate funds for building sea defences, as well as for monitoring the flow of vessels in the coastal region. GIS is also commonly used by transportation companies for planning distribution routes, analysis of accident patterns and optimizing fuel usage.

C. Incident Management Teams may use a GIS to maintain an inventory of floodplains and resource caches, while municipalities use it for planning the location of industrial parks and green spaces, as well as designing the layout of new districts. Spatial analysis via GIS allows to easily establish the percentage of land used in each category, population density levels by district, or the average distance to a nearest hospital in the area of the city.

D. Standalone GIS is most often used for the purpose of cartography, administration or decision support. Local government authorities may use GIS to manage the location and dimensions of land parcels. In this context feature attributes contain data such as references to owners, land use and tax values. Another common application of GIS is in the inventory, management and planning of infrastructure, e.g. concerning the distribution of electricity, supply of fresh water, management of sewerage systems and layout of telecommunication networks.

E. A GIS inventory of environmental hazards in relation to vital resources such as groundwater, may assist in the analysis of contaminant spread rates, cumulative pollution levels, as well as estimation of potential years of life lost in a particular area due to environmental hazards.

Ex. 9

Work with your partner. Match the words to make expressions. How many possibilities can you think of?

a. spatial	1. coordinates
b. local	2. transportation
c. spatial	3. images
d. geographic	4. routes
e. public	5. maps
f. street	6. users
g. satellite	7. services
h. walking	8. network
i. remote	9. analysis
j. mapping	10. Data
k. navigation	11. navigation
l. voice	12. system

Ex. 10

Read the text below and find the expressions in the text.

SERVER GIS

GIS software used for creating and editing of spatial data, as well as performing its spatial analysis is usually confined to a single workstation. If the data is shared between several users of a single institution, it is usually placed in a remotely accessible geodatabase on the local network.

However, if a GIS is to be used by many geographically distributed users, it is referred to as a Web GIS. A Web GIS usually consists of a Geodatabase, a Map Server and a Web browser-based Client.

Popular examples of a Web GIS are on-line mapping services such as Google Maps or OpenStreetMap. Such services offer limited but specialised GIS functionality to remote users through a minimalistic DHTML interface. In addition to browsing several types of maps (e.g. satellite images, street maps and terrain data), popular Web GIS servers offer features such as resolving addresses and calculation of driving and walking routes. A relatively new addition to Google Maps is its Transit service, which helps to design a complete journey (by car, on foot/ferry or using public transportation) between two addresses in selected countries in the world. Other popular functionalities include geolocation of photography (by means of linking it with specific geographic coordinates, referred to as "geotagging") and overlaying user-defined points of interest on the main map window.

MOBILE GIS

A special type of GIS software, referred to as "mobile GIS" may be found in embedded devices such as car navigation systems and Smartphones. Because such devices come with GPS receivers, the primary functionality of mobile GIS is to show the user's position on a vector map of the area. Other applications include route discovery and voice navigation for driving, walking and biking. Modern versions of these devices assume a working wireless internet connection, and thus additionally offer the display of satellite maps (e.g. Google Maps mobile), as well as geocoding services.

Ex. 11

Work in pairs. Cover the text and retell it to your partner using as many expressions as possible.

Ex. 12

Work in groups of three. Discuss a project of a new application. Present it to the whole group. Vote for the most innovative application.

USEFUL PHRASES: It consists of… It is related to… The simplest example is… The most common format used is… The primary functionality is to… Other applications include… Information is most often represented in the form of… This system may be used for the purpose of…

=^.^= =^.^= =^.^=

The Programmer and the Engineer

A Programmer and an Engineer were sitting next to each other on an airplane. The Programmer leans over to the Engineer and asks if he wants to play a fun game. The Engineer just wants to sleep so he politely declines, turns away and tries to sleep. The Programmer persists and explains that it's really an easy game. He explains, "I ask a question and if you don't know the answer you pay me $5. Then you ask a question and if I don't know the answer I'll pay you $5." Again the Engineer politely declines and tries to sleep.

The Programmer, now somewhat agitated, says, "O.K., if you don't know the answer you pay me $5 and if I don't know the answer I pay you $50!" Now, that got the Engineer's attention, so he agrees to the game. The Programmer asks the first question, "What's the distance from the Earth to the Moon?" Then Engineer doesn't say a word and just hands the Programmer $5.

Now, it's the Engineer's turn. He asks the Programmer, "What goes up a hill with three legs and comes down on four?" The Programmer looks at him with a puzzled look, takes out his laptop computer, looks through all his references and after about an hour wakes the Engineer and hands the Engineer $50. The Engineer politely takes the $50 turns away and tries to return to sleep.

The Programmer, a little miffed, asks, "Well what's the answer to the question?" Without a word, the Engineer reaches into his wallet, hands $5 to the Programmer, turns away and returns to sleep.

=^.^= =^.^= =^.^=

Unit 6

Information Systems Engineering – a Road to a Successful IT Project

LEAD-IN:

Ex. 1

Match the words 1-12 with their definitions:

1. SCALABILITY	a. The ability to find or follow something.
2. DEPLOYMENT	b. A typical or stereotypical example.
3. TRACEABILITY	c. To distribute evenly.
4. DISPERSE	d. A layer or level.
5. PARADIGM	e. The act of decreasing or reducing something.
6. CONSTITUENT	f. A component part; ingredient.
7. TIER	g. Organization, arrangement, positioning.
8. MITIGATION	h. The quality of being measurable.
9. SCOPE	i. The area covered by a given activity or subject.
10. ATTRIBUTE	j. A restriction or limitation.
11. PREREQUISITE	k. A property, quality or feature.
12. CONSTRAINT	l. Something that is required in advance.

Ex. 2 ***T13***

Repeat the following words after the recording:

scope, *attribute*, *scalability*, *prerequisite*, *deployment*, *to elicit*, *to trigger*, *to validate*, *stakeholder*, *unambiguous*, *cross-reference*, *traceability*, *dispersed*, *paradigm*, *constituent*, *encapsulate*, *tier*, *debugged*, *constraint*, *agile*, *mitigation*, *countermeasures*

Ex. 3

In pairs discuss the features of a successful IT project.
Share your ideas with the whole group.

Ex. 4

Read the questions and discuss the answers in groups of three.

a. What is the success of any IT project?
b. What does the information systems engineering focus on?

c. What is the difference between the information systems engineering and the project management?
d. What is the system scope?
e. What is the key challenge in IT business?
f. What does “the quality of a software-based system” mean?

TEXT-BASED TASKS:

Ex. 5 T14

You are going to listen to the recording about designing IT projects. Listen and note down the answers to the questions from Ex. 4.
Compare your answers with another group.

Ex. 6 T14

Listen again and complete your notes. Check with the whole group.
When you have finished check your notes with the tapescript at the back of the book.

Ex. 7

Read the text ‘Systems engineering’ and note down the subsequent steps in systems engineering and the main information about them.
Check with your partner.

SYSTEMS ENGINEERING

Systems engineering follows an organized process to understand the users’ problem, find the solution, implement it with software and finally deploy and operate it. The process of systems engineering can be divided into several disciplines:

business modelling,
requirements engineering,
architectural design,
detailed design,
implementation,
testing,
deployment.

Business modelling aims at capturing the customer’s business which the system is intended to support and, based on that, elicit the users’ needs to be met. The customer’s business is decomposed into business processes, which, by employing people in specific roles, perform defined activities processing input resources into output products and providing value to the entire business. The structure of activities within a business process is specified by the process workflow driven by business rules. The process is triggered by events and may generate new events itself. A business model documents all these elements of the customer’s business. Nowadays, it is often expressed with Business Process Modelling Notation (BPMN).

Having identified the actual and target business model, system engineers (system analysts in particular), analyse it to identify, document and validate the requirements for a business information system. At this stage it is crucial to identify and involve all individuals and parties which are entitled to define the shape of future system (so called system stakeholders). Requirements gathered from stakeholders have to be analysed, as usually several problems occur (e.g. lack of sufficient detail, conflicting requirements), and validated through additional communication with related stakeholders. Requirements also need to be documented in an unambiguous way, not only with their contents but also priorities, cross-references or traceability to sources. Commonly used representations of requirements include structured text (predefined tables) and semi-formal notations combining text and graphical elements. An example of the latter could be “use case diagram” from Unified Modelling Language (UML) which visualises how frequently used system’s functions ('use cases') depend on each other.

The next step in systems engineering is the design of the system’s architecture, which defines the overall structure of the system often dispersed among different hardware and physical locations. Among different architectures we can distinguish stand-alone and distributed systems. Stand-alone systems process all input on a single device, be it a personal computer, a mobile phone or a networking device. Stand-alone systems for PCs are often named desktop applications. They run within the operating system of the host machine. Distributed systems process user input on multiple networked machines playing different roles depending on the architecture type. So called client-server architecture employs servers as main processing units, and clients as input-output units. Depending on the volume of processing on the client machines we distinguish thin, thick and rich clients. Peer-to-peer architecture (P2P) consists of a number of identical nodes balancing the processing equally over the entire system. Modern architectures follow service oriented paradigm (SOA) or cloud processing. Distributed systems run over the Internet are simply called internet applications. Selection of adequate hardware and system architecture is crucial to reduce the risk of project failure.

Once the overall architecture of a system is decided, the constituent subsystems must be designed in details with chosen technologies and programming languages. Most of them currently follow the object oriented paradigm, which structures the system into objects communicating with other objects through explicit interfaces. Each object is created as an instance according to its specification – its class. A class defines the object’s attributes and methods which provide its behaviour. Objects encapsulate their behaviour to conceal the processing details and provide for better abstraction. Classes can also inherit properties and behaviour from other classes (called parent classes). Then they can be overridden with custom behaviour of a child class preserving the interface of the parent class, which satisfies the polymorphism principle. Classes are grouped into packages and components. In UML, class diagram presents a system’s classes with their relationships. An example class diagram for an e-learning university system, where teachers give courses on a certain subject to students and grade them is shown in Fig. 6.1.

Systems engineering has elaborated many design patterns to promote simple but effective designs. Among them, the most common are MVC (model-view-controller), layers (also named tiers), both separating data storage, processing and presentation, and ORM (object-relational mapping) connecting an object oriented code to a relational database.

The classes or other pieces of system structure like scripts are implemented in source code with Java, C#, PHP, Python, Ruby or other programming language. Programmers write their code in an Integrated Development Environment (IDE) like Eclipse, Netbeans, Visual Studio and other. IDE often comes with a development framework, a set of reusable libraries, architecture and code patterns and meta-code solutions like tagging. Source code is compiled or otherwise converted into an executable build that can be run in a target environment.

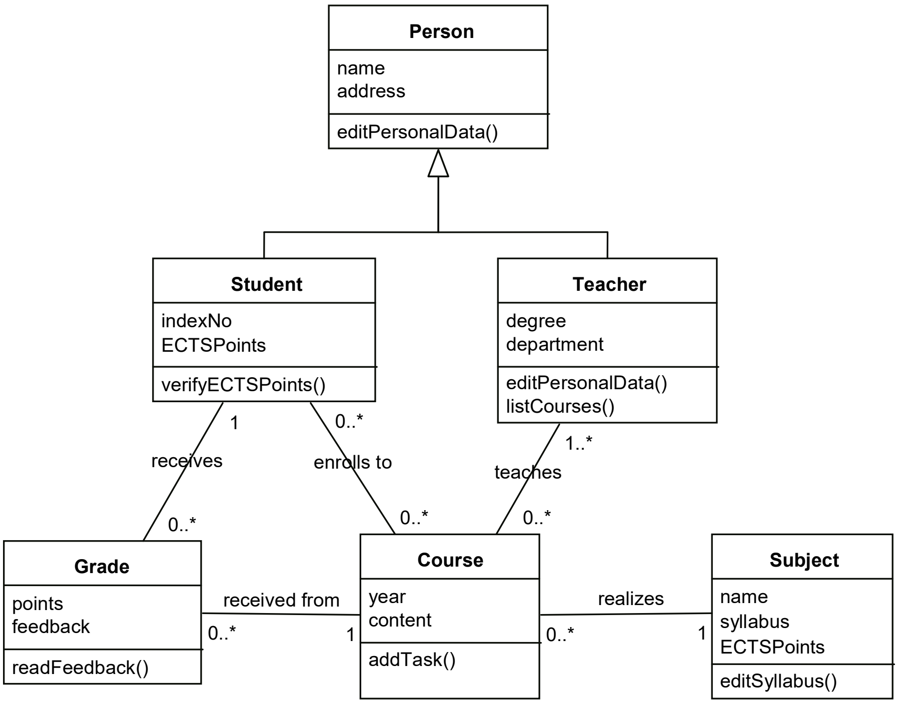


Fig. 6.1. Class diagram of an e-learning system for a university

Implementation of software is inevitably tied to software testing. Testing is aimed at detecting defects in the code (so called "bugs") by running its executable representation in a series of test cases. Numerous techniques enabling detection exist, but they can be grouped into two main approaches: black-box testing which focuses on inputs and outputs of the tested system without any knowledge of its internal structure and white-box testing which utilises the knowledge of code structure for test cases preparation. Many tests can be automated, where test cases are implemented as test scripts and run within a testing environment which makes sure test results are recorded, compared to expected values etc. Automation also allows to reduce costs of testing and to avoid dull work (e.g. regression testing – running all test cases after some modifications to the system were made).
Tested and debugged system can be considered 'a product' and deployed in the user environment. Deployment requires setting up necessary tools and libraries such as a database management system, the installation of particular subsystems on target machines, and finally integration and tuning of the entire system. Successful deployment also covers user training, where users are familiarized with system functions and interfaces.

FOLLOW-UP EXERCISES:

Ex. 8

In pairs, looking at your notes, describe the details of each step. Change roles. Use the following phrases:

USEFUL PHRASES:	
To begin with…	Having done that…
At this stage…	The next step is…
Once this is done…	Finally…

Ex. 9

Read the text below and put the following words into the gaps.
ATTENTION: there are more words than you need.

management, *problems*, *defects*, *product*, *quality*, *produce, software*, *performance*, *development*, *techniques*, *users, manage*

QUALITY ASSURANCE

Quality of a system does not only mean lack of 1) ....and resulting failures. It also includes the suitability of the system to the needs of its 2) .... right scope of the functionality, sufficient guarantees related to non-functional characteristics like performance, security or maintainability and finally an ergonomic user interface. 3) ....is rarely a feature that can be added last minute, it has to be introduced into the system from the early phases of 4) .... Therefore, quality is not only attributed to the final 5) .... (delivered system), but also to its earlier representations like requirements or UML models. Moreover, quality of the development process, developers and tools should also be assured.
Quality assurance uses several 6) .... One of them is 7) .... testing mentioned before, however more specialised tests are applied to verify particular quality attributes, for example 8) .... tests to check how the system works with large portions of data or numerous users, penetration tests to simulate security threats or usability tests to detect 9) .... with the user interface. Testing is limited to executable code only, hence different techniques have to be applied for other artefacts such as models or documentation. The simplest is a peer review but much more formalised techniques like software inspections could also be employed. The quality of the development process and its elements is closely related to the domain of project 10) ....

Ex. 10

Read the text "Project management" and find the answers to the following questions.

1. What does it mean to deliver the product efficiently?
2. What does project management do?
3. What is the most essential success factor of IT projects?
4. What is methodology of IT projects?
5. What is the difference between plan-driven and risk-driven methodologies?

6. What areas does project management cover?
7. What does human resource management deal with?
8. What competences does effective team management require?
9. What does configuration management provide for?
10. What does risk management aim at?
11. What are the chronological steps in risk management?
12. What strategies does risk mitigation employ?

PROJECT MANAGEMENT

A successful IT project must deliver the product efficiently, which means keeping the project within assumed time and budget constraints. To achieve this, the project must be *managed*. Project management creates advantageous environment for systems engineering to provide customers with effective IT solutions.

Selection of appropriate development methodology is one of the most essential success factors of IT projects. Methodology is a set of principles and procedures driving the process of software development. Plan-driven methodologies focus on well-defined processes with detailed schedules to follow, while risk-driven ones organize the project phases around risk reduction. On the contrary, agile methodologies emphasize the creativity and responsibility of individuals within the process, promote running product over detailed designs, and address rapidly changing user requirements.

Project management covers a set of specific areas such as time management, budgeting, human resource management, configuration management, and risk management. Time management uses schedules often visualized in Gantt charts to track project progress and maintain it to meet the deadline. An example Gantt chart is presented in Fig. 6.2.

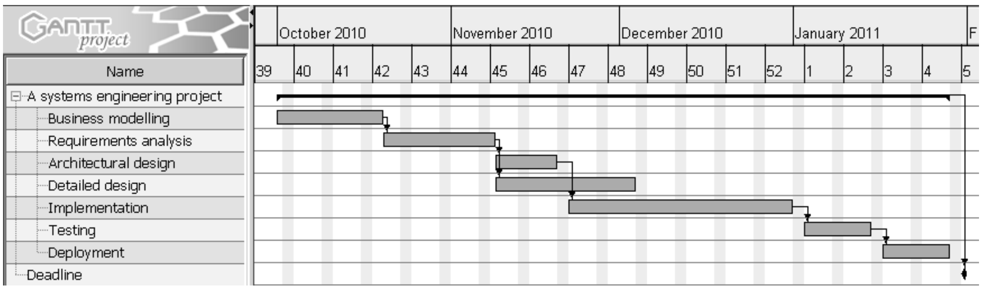


Fig. 6.2. Example Gantt chart

Human resource management includes staffing, training, task delegation, motivation, and monitoring of the project team. Nowadays, information systems are mostly developed in teams, whose size varies from several to even thousands of people. Effective team management requires many social ("soft") competencies like communication skills, fast decision-making, empathy, intuition, as well as presentation and negotiation skills.

Configuration management supports system development in maintaining all resources like source code, executables, documentation, test cases and scripts. It provides for versioning of resources with repositories such as Subversion or CVS, controlled resource sharing,

document formatting and naming conventions and, last but not least, management of resource configurations into correct product packages.
Risk management aims at reducing the risk of project failure with an organized process comprising risk identification, analysis and mitigation. First, risks with their contributing factors should be identified using a check-list, questionnaire or brainstorming. Then, the risks are estimated in terms of likelihood and impact, and a number of biggest risks are qualified to mitigation, while the rest is only monitored. Risk mitigation employs various strategies such as reduction, avoidance, transfer or acceptance. Based on selected strategy, detailed countermeasures are planned and integrated into overall project plan. Finally, the risk is continuously monitored and, if necessary, identified and analysed again till the project termination.

Ex. 11

Discuss your answers with your partner. Then check with the whole group.

Ex. 12

Work in pairs. Prepare a summary for this unit, including 5 factual mistakes.
Present your summary to another pair and let them find and note down the mistakes.
Change roles.

The Man And The Dog

There is an OLD story about the data centre of the future.

This data centre runs 24/7 with only a man and a dog.

The man's job is to feed the dog.

The dog's job is to make sure the man does not touch the computer.

Unit 7

Ubiquitous and Pervasive Processing

LEAD-IN:

Ex. 1

Match the words 1-12 with their definitions:

1. UBIQUITOUS	a. Existing or being everywhere, or in all places, at the same time; omnipresent.
2. AUGMENTED	b. Any electronic device that changes one form of energy into another, such as a microphone, which changes sound waves into electrical signals.
3. ACTUATOR	c. A label assigned to identify data in memory. d. A sequence of characters in a markup language used to provide information, such as formatting specifications, about a document.
4. TRANSDUCER	e. A device responsible for actuating a mechanical device, such as one connected to a computer by a sensor link
5. DEPLOYMENT	f. Compile (a computer program) again or differently.
6. INTEROPERABILITY	g. Transforming a mechanical, electrical, or computer system from a packaged form to an operational state. h. All of the activities that make a software system available for use.
7. TO RECOMPILE	i. The ability of software and hardware on multiple machines from multiple vendors to communicate.
8. TAG	j. Added to or made greater in amount or number or strength; made greater in size or amount or degree.
9. VENDOR	k. The mechanical advantage gained by employing a lever; the power to accomplish something; strategic advantage.
10. TRAPDOOR	l. A standard procedure for regulating data transmission between computers.
11. PROTOCOL	m. A hinged or sliding door in a floor, roof, or ceiling.
12. LEVERAGE	n. A person who sells something.

Ex. 2 ***T15***

Repeat the following words after the recording:

ubiquitous, *augmented*, *actuator*, *interoperability*, *suite*, *transducer*, *calibration*, *heterogeneous*, *vendor*, *deployment*, *trapdoor*, *to track*, *authentication*, *protocol*, *gesture*, *to encounter*, *implementation*, *to recompile*, *appropriate*, *to enable*, *access*, *leverage*

Ex. 3

Work in pairs. Discuss what examples of ubiquitous systems technology you know in everyday life. Consult with another pair.

TEXT-BASED TASKS:

Ex. 4

Complete the gaps with the following words:
ATTENTION: there are more words than you need.

standards, *user*, *intelligence*, *decision*, *environment*, *intelligent, sensors*, *system*, *computers*, *academic*, *information*, *compute, actuators*, *context*

The ubiquitous systems technology covers a wide range of 1) .... , architectures and working solutions in both 2) .... and industrial environments. Generally, a ubiquitous system is distributed, aware of 3) ....context and autonomous. Such systems intend to hide human-system interaction by automated 4) ....making and augmenting the environment with autonomic devices for context recognition. 5) ....may handle many actions and interactions in the environment, which requires some elements of artificial 6) .... Ubiquitous systems use 7) ....that monitor the environment, for example measure temperature or detect user movement, and pass the 8) .... to processing modules. Additionally, the systems use 9) .... that perform environment actions, for example turn the light on/off or open the door. Usually, a ubiquitous 10) .... monitors the environment, detects contexts and performs environment actions depending on the 11) .... Typically, the processing does not involve explicit human system interaction, although the 12) ....influences the system by his behaviour.

Ex. 5

Read the text below and answer the questions.

1) What does the IEEE 1451 suite of standards specify?
2) What is the primary focus of the IEEE 1451 suite?
3) What does Transducer Electronic Data Sheet describe?
4) What do web-based standards cover?
5) How are devices usually integrated?

Sensors interoperability is the most important aspect of the technology in the context of this paper. The IEEE 1451 suite of standards specifies how sensors should integrate with ubiquitous environments and how their data should be processed. The primary focus of the suite is to enable sensor-to-network plug-and-play integration. The standards generalize sensors and actuators to transducers that represent any device that is integrated in a ubiquitous system. Standards define Transducer Interface Module that represents a smart transducer capable of automated integration. Additionally, Transducer Electronic Data Sheet describes transducer giving sensor type, measurement range, calibration data and user information.
Web based standards are very interesting and cover, amongst other things: Sensor Web Enablement and IEEE 1451 Web Services Interface. The solutions take advantage of gen-

eral purpose Web standards, but there is a lack of standards for sensor description, data messaging and sensor location. Some research papers propose other solutions that have not yet been considered as standards. For example Device Profile for Web Services and the Domonet Middleware are proposed as mechanisms for integration of heterogeneous platforms. Other important standards concern Transducer Markup Language (XML-related solution), Common Alerting Protocol, Sensor Markup Language and Sensor Data Model.
Typically, devices are integrated by wireless communication with numerous standards available. WiFi, DECT and 3G are usually used in mobile phone networks. Standards of RFID device communication cover lower and higher level standards that concern video processing (HAVi), device discovery (Jini, SSDP, SLP) and service management. We leverage the knowledge about existing standards to integrate concrete services in a test platform. We rely on the interface supplied by vendors of devices and implement custom services that supply a common Web services interface for the devices.

Ex. 6

Discuss the answers with your partner and then check with the whole group.

FOLLOW-UP EXERCISES:

Ex. 7

Read the text below. Put the words in brackets into the proper form – active or passive. Then check with the whole group.
What devices are described?

DESCRIPTION OF DEVICES

The door lock service is responsible for electronic manipulation of a trapdoor in a laboratory. The use of automated processing 1)………………..(enable) to extend functionality of a traditional trapdoor. Enhanced security is one of the most important factors that characterises the device, as user identification and access rules may easily 2)…………………(configure) without a physical access to the door. Additionally, electronic door locks may be managed without any key by combining with devices for biometric information processing such as fingerprint reading, RFID or video recognition.
In our experiment, a dedicated device (BERA lock) 3)………………..(manage) physical operations on the trapdoor such as opening, closing, locking and unlocking. The device 4)……………….. (equip) with a beeper that emits sound when a user has access to the laboratory (when trapdoor is in the "close" position). The sound effect stops after opening the door handle. The device is networked using ATMega server through the Ethernet wire. It may 5)………………..(control) using a Web services-available service
Radio-frequency identification (RFID) 6)………………..(use) typically to identify objects by attaching to them an electronic tag that communicates with a base terminal using electromagnetic waves. The technology widely 7)………………..(use) in tracking people, devices and vehicles. In a person tracking system, for example, a ubiquitous environment may 8)………………..(unlock) door if a relevant person 9)………………..(detect) in front of the door.
In our experiment, the RFID authentication service is responsible for processing of RFID devices. CAENRFID 949EU IP65 UHF stand alone reader with HUBER+SUHNER AG

806–960 MHz antenna 10)………………..(use) for user authentication. We used typical RFID infrastructure: a receiver is waiting for users having RFID identifiers. When a user with an identifier 11)……….(appear), the receiver reads the identification and passes it on to the service. The RFID device 12)…………………(connect) using serial port.

The Automatic file upload and download services 13)………………..(enable) automatic file download or upload if a user is in the right place at the right time, considering his actual schedule of work (for example in a lecture room at the beginning of a lecture). The service authenticates the PDA device automatically in the first step of processing. If the authentication is successful, the user 14)…………………..(give) access to his directories on an FTP server and methods for file transfer. The Linksys WRT53GL v.1.1 router was used to enable integration of the PDA devices with the server. Wireless enabled PDA devices automatically try to identify themselves in a wireless network emitted by the access point. Regular file transmission 15)……………………(realise) using the FTP protocol.

In the gesture recognition service we used the Logitech QuickCam Pro5000 Internet camera with AForge API to recognize gestures of an observed user. Background setup was the first step required for appropriate image processing. After the setup the user's figure is identified by the application, and gesture recognition is done using the AForge API.
An exemplary scenario covered integration of gesture recognition and Power Point presentation management. If a user raises his right hand during a presentation, the system displays the next page of the presentation.

Generally, although the services seemed operational, sensor installation and service deployment were unexpectedly difficult in some cases. The difficulties resulted from detailed implementation issues such as unknown installation procedure of a service or statically coded network location (port number, IP address). It was necessary to analyse the source code of a service, correct and recompile it to fit the target platform. Development documentation gave important information that simplified the process. In some cases, it was sufficient to follow the documentation for a successful installation of a service. The deployment of the corrected services in the IIS server did not encounter difficulties.

Ex. 8

Read the text again. Note the characteristic features of the devices.

Ex. 9

Work in pairs. Looking at your notes describe the different devices to each other.
When you have finished, look back at the text to check the details.
Change roles.

Ex. 10

Work in pairs. Write 10 true/false sentences to the text.

Ex. 11

Work with another pair. Read your sentences in a group of 4 and decide if they are true or false.

Ex. 12

Design a device that you would need in your everyday life.
Share your idea with the whole group.

USEFUL PHRASES:
The most important aspect is…
Additionally…
The primary focus is to enable…
It would be typically used to…
It enables…
It may handle…
The solution takes advantage of…

:⊲) :⊲) :⊲)

A Wish

A programmer is walking along a beach and finds a lamp. He rubs the lamp, and a genie appears. “I am the most powerful genie in the world. I can grant you any wish, but only one wish.”

The programmer pulls out a map, points to it and says, “I’d want peace in the Middle East.”

The genie responds, “Gee, I don’t know. Those people have been fighting for millennia. I can do just about anything, but this is likely beyond my limits.”

The programmer then says, “Well, I am a programmer, and my programs have lots of users. Please make all my users satisfied with my software and let them ask for sensible changes.”

At which point the genie responds, “Um, let me see that map again.”

:⊲) :⊲) :⊲)

Unit 8

Fundamentals of Robotics

LEAD-IN:

Ex. 1

Match the words 1-12 with their definitions:

1. CONTRIVANCE	a. The under part of an automobile, consisting of the frame (on which the body is mounted) with the wheels and machinery.
2. GRIPPER	b. To cause to slope, as by raising one end; incline.
3. TO TILT	c. Straight; consisting of a straight line or lines; bounded by straight lines.
4. RECTILINEAR	d. Projecting horizontal beam fixed at one end only.
5. GANTRY	e. The joint, or the region of the joint.
6. CANTILEVER	f. A device that is very useful for a particular job.
7. WRIST	g. A mechanical device that grasps and holds.
8. CHASSIS	h. A bridgelike framework used to support a travelling crane, signals over a railway track.
9. COUNTERPART	i. Either of two opposed hinged parts in a mechanical device.
10. PIVOTAL	j. Imaginary, not genuine or authentic; assumed; false.
11. JAW	k. One that has the same functions and characteristics as another.
12. FICTITIOUS	l. Being of vital or central importance; crucial.

Ex. 2 ***T16***

Repeat the words after the recording:

fictitious, *counterpart*, *contrivance*, *wrist*, *pneumatic*, *interpolation*, *cylindrical*, *gripper*, *pivotal*, *robust*, *rigid*, *assembly*, *chassis*, *versatile*, *joint*, *jaw*, *to mount*, *gear*, *actuator*, *to tilt*, *rectilinear*, *gantry*, *cantilever*

TEXT-BASED TASKS:

Ex. 3 ***T17***

You are going to listen to the history of robots. Listen carefully for the following information:
Who is the father of the term “robot”?

Ex. 4 ***T17***

Listen to the recording again and decide if the following sentences are true or false:

1. The term „Robot” was first used in a poem called „Rossum’s Universal Robotts”.
2. The play was written by Karel Čapek in 1923.
3. Karel Čapek predicted the rise of a powerful robot industry.
4. In 1948, an influential artificial intelligence research paper, “Cybernetics” was published by Norbert Wiener.
5. Today’s robots are both mechanically and structurally close to handling the same tasks as their fictitious counterparts.
6. Contemporary robots can navigate in space on their own.
7. Robots can only follow human commands.
8. Some robots are able to carry out conversation.

Ex. 5

Read the text below. Write 10 questions to the text.

One of the most important robot components is a manipulator. Manipulator is a fancy name for a mechanical arm. A manipulator is an assembly of segments and joints that can be conveniently divided into three sections: the wrist, usually consisting of one to three segments and joints; and a gripper or other means of attaching or grasping. Industrial robots are stationary manipulators whose base is permanently attached to the floor or a stand. Trying to classify manipulators one has to take into account six categories:

— arm geometry: rectangular; cylindrical; spherical; jointed-arm (vertical); joined-arm (horizontal).
— degrees of freedom (DOF): robot arm; robot wrist.
— power sources: electrical; pneumatic; hydraulic or any combination.
— type of motion: slow motion; joint-in-interpolation; straight-line interpolation; circular interpolation; any-line interpolation.
— path control: limited sequence; point-to-point; continuous path; controlled path.
— intelligence level: low-technology (no feedback control); high technology (feedback control, servo).

Actually, the six or more joints of the manipulator can be in any order, but there are only a few combinations of joint order and segment length that work effectively. They almost always end up being divided into arm and wrist. A good example of a manipulator is the human arm, consisting of a shoulder, upper arm, elbow and wrist. The three twisting motions of arm segments give the position of the wrist. The next three twisting motions of wrist joints, which give orientation, commonly labelled as pitch, roll and jaw for tilting up/down, twisting and bending left/right, respectively.

On a mobile robot, the chassis can often substitute for one or two of the degrees of freedom, usually fore/aft and sometimes to jaw the arm left/right, significantly reducing the complexity of the manipulator.

The three general layouts for three-DOF arms are called Cartesian, cylindrical and polar (or spherical). They are named according to the shape of the volume that manipulator can reach and orient the gripper into any position – the work envelope. They are all useful but some are better for a particular robot than others. Some use all sliding joints, some use only pivoting joints, some use both. Pivoting joints are usually more robust than sliding joints but, with special design, or extending can be used effectively for particular types of task.

Ex. 6

Work in pairs. Quiz each other with your questions. Do NOT look back at the text. When you have finished, check if you remembered correctly.

Ex. 7

Read the text and fill the gaps with the following words:
ATTENTION: there are more words than you need.

items, *environment*, *actuator*, *gripper*, *jaws*, *type*, *knowledge*, *actuate, kinematic*, *object*, *screws*, *base*, *environmental, task*

THE GRIPPER

The end of the manipulator 1) …................... chain is the part the user or robot uses to affect the 2) …........................ .. For this reason it is commonly called an end-effector, but it is also called a 3) ….....................as it is a very common 4) …............ for it to perform, when mounted on a robot. It is often used to pick up different 5) …............. for the robot to carry, some can turn 6) …..................... or door-knobs or manipulate various tools. Closing too tightly on an 7) …............... and crushing it is a major problem with autonomous grippers. For these reasons, gripper design requires as much 8) …................... as possible of the range of items the gripper will be expected to handle. Some objects require grippers that have many 9) …............., but in most cases, grippers have only two jaws. There are several basic types of gripper geometries. The most basic 10) …................ has two simple jaws geared together so that turning the 11) …................ of one turns the other. This pulls the two jaws together. The jaws can be moved through a linear 12) …............... or can be directly mounted on a motor gearbox output shaft, or driven through a right angle drive which places the drive motor further out of the way of the gripper. Examples of two simple design grippers are shown in Fig. 8.1.

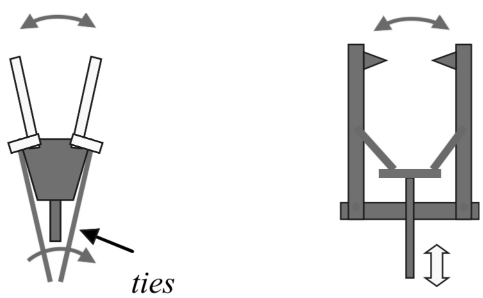


Fig. 8.1. Two types of two fingers grippers a) tie driven gripper, b) reciprocating lever gripper

Ex. 8

Work in groups of three. Decide who is A, B, and C.
Each of you reads one text (A, B or C).
When you have finished reading, in your groups three, retell the text you have read giving as many details as possible. Other students listen and take notes.
Finally, read the the other texts to check whether your notes were correct.

TEXT A

RECTANGULAR LAYOUT

Cartesian or rectangular layout robot looks like a three dimensional XYZ coordinate system. In fact, that is how the working end moves around the work envelope. There are two basic layouts based on how the arm segments are supported, gantry and cantilevered. Kinematic chain for rectangular layout robot has been presented in Fig. 8.2. Gantry type robots have some important advantages:

- they can obtain large work envelope, its stroke along the X-axis and Y-axis can be increased easily;
- their linear movement allows for simpler controllers;
- they have high degree of mechanical rigidity, accuracy, and repeatability due to their structure;
- they can carry heavy loads because the weight-lifting capacity does not vary at different locations within the work envelope.

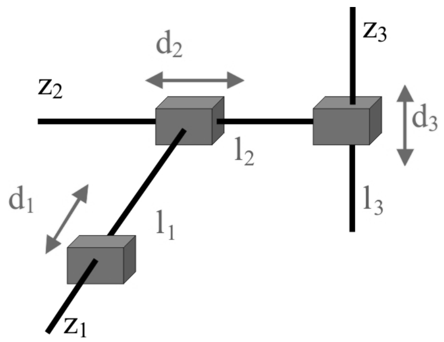


Fig. 8.2. Gantry robot kinematic chain

TEXT B

CYLINDRICAL LAYOUT

The second type of manipulator work envelope is cylindrical. Cylindrical robots usually incorporate a rotating base with the first segment able to telescope or slide up and down, carrying a horizontally telescoping segment. While they are very simple to picture and the work envelope is fairly intuitive, they are hard to implement effectively because they require two linear motion segments both of which have moment loads in them caused by the load at the end of the upper arm. In the basic layout, the control code is fairly simple, i.e. the angle of the base, height of the first segment, and extension of the second segment are easy to describe using cylindrical coordinate system. Kinematic chain for cylindrical layout robot and its work envelope shape have been presented in Fig. 8.3.

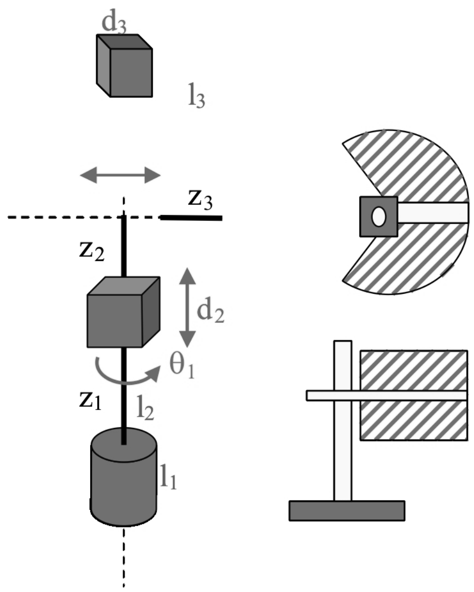


Fig. 8.3. Cylindrical layout robot kinematic chain and its work envelope shape example

Cylindrical layout robots have some important advantages:

- their vertical structure conserves floor space;
- their deep horizontal reach is useful for far-reaching operation;
- their capacity is capable of carrying large payloads.

Unfortunately those kinds of robots have also some disadvantages:

- their overall mechanical rigidity is lower than that of rectilinear robots because their rotary axis must overcome inertia;
- their repeatability and accuracy are also lower in the direction of rotary motion.

TEXT C

SPHERICAL LAYOUT

The third and most versatile geometry is the spherical type. In this layout, the work envelope can be thought of as being all around. In reality, though, it is difficult to reach everywhere. There are several ways to layout an arm with this work envelope. The most basic has a rotating base that carries an arm that can pitch up and down (shoulder). The next segment is placed at the end of the first one and connected with it using rotary joint (elbow). The work envelope for that type of robot is like a section of sphere. The first motion corresponds to a base rotation about a vertical axis. The second motion corresponds to a shoulder pitch and the third motion corresponds to an elbow rotation. A spherical-coordinated robot provides a larger work envelope than the rectilinear or cylindrical one. Kinematic chain for spherical layout robot and its work envelope shape have been presented in Fig. 8.4.

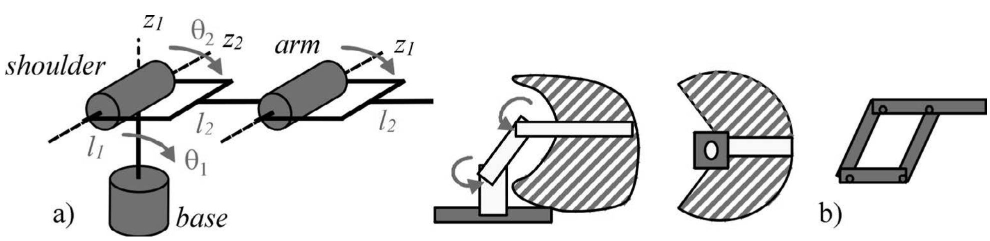


Fig. 8.4. Spherical layout robot kinematic chain and its work envelope shape example (a). An alternate elbow construction – better for large loads (b)

FOLLOW-UP EXERCISES:

Ex. 9

In pairs, read the following sentences carefully and put them in the logical order to make a concise text. When you have finished, check your version with the key.

- The move toward open architecture controls is relatively recent but follows the trend in computers.
- An industrial robot is easy, but putting it into an intelligent work cell requires much more than the robot
- Open architecture control refers to software designs that can use or be used with products from a variety of manufacturers.
- Important accessories such as process tools, safety devices, programmable logic controllers, computers and simulation programs are needed to make robots easier to use.
- The control system is the set of logic and power functions that allows the automatic monitoring and control of mechanical structure and permits it to communicate with the other equipment and user in the environment

Ex. 10

Fill in the conditional sentences with the proper form of the verbs given. When you have finished check your answers with the whole group.

1. What would you do if you __________ (buy) a new game, and it __________ (not work)?
2. If he succeeds with the project, I _________ (tell) you.
3. If they hadn't prepared that program on time, I __________ (let) you know.
4. If I won a lottery, I __________ (invest) in a new website.
5. If they had been free, they __________ (finish) their work earlier.
6. It __________ (be) silly if we tried to do it our way.
7. Would you mind if I __________ (use) your mobile?
8. If we __________ (start) the work earlier, we wouldn't be in such a hurry right now.
9. If they were sensible, they __________ (not ask) that question before the project was completed.
10. I __________ (not open) that mail if it had contained a virus.

Ex. 11

Work in groups of three. Design a project of a robot. Present it to the group. Vote for the most innovative one.

USEFUL PHRASES: One of the most important components would be… A good example of …… could be… This robot would look like… Some robots require… It should incorporate… It would have the following advantages/functions…

|☺ |:-) |☺

Surprise

"Knock, knock."

"Who's there?"

very long pause….

"Java."

|☺ |:-) |☺

Unit 9

Vector Space Model in Text Retrieval

LEAD-IN:

Ex. 1

Match the words 1-12 with their definitions:

1. COMMODITY	a.	To recover or make newly available (stored information) from a computer system.
2. RETRIEVE	b.	To overwhelm with a large number or amount.
3. QUERY	c.	The extent to which information is available in respect to any specified area of interest.
4. ORTHOGONAL	d.	Advantage; benefit.
5. COSINE	e.	Mutually independent; well separated.
6. QUOTIENT	f.	The result of the division of one number or quantity by another.
7. DELUGE	g.	A question; an inquiry.
8. COVERAGE	h.	In a right triangle, the ratio of the length of the side adjacent to an acute angle to the length of the hypotenuse.
9. STEM	i.	An order of succession; an arrangement.
10. SUFFIX	j.	The main part of a word to which affixes are added.
11. SEQUENCE	k.	Anything that is added at the end of something else.
12. TO COMPENSATE	l.	To undo the effect of a disadvantage; make up for shortcomings.

Ex. 2 ***T18***

Repeat the following words after the recording:

commodity, *to retrieve*, *query*, *fair*, *to compensate*, *length*, *orthogonal*, *cosine*, *logarithm*, *quotient*, *stem*, *suffix*, *root*, *stemmer*, *deluge*, *coverage*

TEXT-BASED TASKS:

Ex. 3

Read the following sentences and put them in a logical order to make a concise text:

- Additional problems may arise from the situations where texts are written in a foreign language, or when long documents match a query, so that they have to be summarized.
- Even when the data they had been looking for was there, it was not easy to find.

- One of the most important commodities in people's life is information. It does not mean the task of finding a specific piece of data is very easy.
- The problem is that there is so much information that rejecting irrelevant items and ranking the relevant ones is more difficult than finding anything that might be related to a *query*.
- There is a large difference between the situation twenty years ago and now for people who are looking for a specific piece of information.
- There were catalogues printed on small cards, but their quality was disputable.
- With the advent of the Internet, things have changed.
- Twenty years ago, they would have to go to a bookstore or to a library, and not only to a single one.
- People are faced with a deluge of information.
- Its importance is constantly growing.
- It does not mean the task of finding a specific piece of data is very easy.

Ex. 4 ***T19***

Listen to the recording and check the whole text from Exercise 3.

Ex. 5

Work individually. Read the following text and fill the gaps with the words from the list. When you have finished check with your partner, then with the whole group.

document, *information retrieval*, *documents*, *picture*, *knowledge*, *sequence*, *universe*, *sentence*, *languages*, *text retrieval*, *document retrieval*, *query*

Finding a piece of information that matches a query is called 1) ………………. While a 2) ………….. may be worth a thousand words, and a film is a 3) ………………… of pictures possibly with sound, much of the knowledge in the 4) ………….. is stored as written texts. The texts are written in natural 5) ………., such as English, Polish, Arabic, Chinese, etc. A part of information retrieval concerned with finding written texts is called 6) ………. or 7) ………….

Let us consider the problem of finding documents on a given subject. A user should pose a 8) ………. stating what she or he wants to find. It may be a sentence or two describing the subject. A text retrieval system should understand the query. It should also understand every 9) ………. in its collection. Then it should match the query with relevant 10) …………. Unfortunately, understanding text is not as easy as one might think. 11) ………. of meaning of words may be insufficient for understanding a 12) …………. . Humans use their knowledge of the world to complete the task. That knowledge is hard to code.

Ex. 6

Work in pairs. Decide who is A, and who is B. Student A – read text A and note down key information. Student B – do the same with text B.

SPIS TREŚCI

TEXT A

If understanding a query or a document is hard to perform, we have to turn to simpler methods. We may treat a query as a document, and a document – as a *bag of words*, i.e. a set of words without their context. Suppose the query was:

Who invented the telephone?

It seems obvious that the documents containing the answer most probably also contain the word **telephone**. They would also probably contain the word **invented**. So a solution might be to select all documents containing words present in the query. Technically, this means associating each document (and a query, as we treat a query as another document) with a vector that has a place for each word that may occur in the whole collection of documents. We would put 1 in that place if the document contains the word, and 0 otherwise. Given the vector $q = (q_1, ..., q_N)$ for the query, and $p_i = (p_{i1}, ..., p_{iN})$ for an i-th document, a measure of similarity is:

$$sim(q, p_i) = \sum_{j=1}^{N} P_{ni} q_j \, p_{i,j}$$

This formula, however, does not take into account differences between documents containing the same words. A document on the invention of the telephone probably mentions **telephone** and **invented** many times while other documents may contain single instances of those words. It is, therefore, better to count the words instead of merely note their presence. Each of the vectors would now contain a count of words, and the similarity measure formula would stay the same.

There is a fair chance that a sufficiently long story, like e.g. a book (but not a historical one) would contain many passages with the word **telephone**, the word **invented**, and many other words that can occur in other queries. The longer the document, the greater the chance that a particular term is there. Our similarity measure favours longer documents. To compensate for that, the vectors should be *normalized* for their length. It means that each *weight* in the vector (each vector position) should be the count for the word divided by the *length of the vector*. The length is calculated as the sum of the squares of the counts.

Normalization produces vectors of length 1 in an N-dimensional space. Our similarity measure is a cosine of angle between the two vectors. When the cosine is equal to one, the vectors are equal. When the cosine is zero, the vectors are orthogonal, which also means they are totally independent of each other. That is why the model is called the *space vector model*.

We have focused so far on the words **telephone** and **invented**. The query, however, contains two other words: **who** and **the**. Obviously, they are far less important in our query, but how a system could know that? Common words are less important than keywords, as they appear in most documents. Words that appear in fewer documents seem to be much more associated with the subject of those documents. An *inverted document frequency* idf_j is a measure of the importance of a word. It is calculated as a logarithm of a quotient of the number of documents where the word occurs and the number of documents in the collection. A *term frequency* tf_j is the normalized count for the word. The weight of a word j would now be $tf_j * idf_j$.

TEXT B

Certain words occur in all documents. They include determiners, prepositions, pronouns. The idf measure for them would therefore be zero. Such words are called *stopwords*. Holding a place for them in the vectors would be a waste of space, and including them in computations would be a waste of time. A *stoplist* lists all stopwords, i.e. words that should be excluded from the vectors.

The word **invented** is the simple past or past participle form of the *lexeme* to **invent**. The most relevant document may contain a phrase:

This led Graham Bell to invent the telephone.

Another possibility might be the phrase:

That led to the invention of the telephone.

Invention is not the same lexeme as to invent, but it is related. To expose that relation, *stemmers* cut off suffixes of words leaving their *stems*, or as in the case of **invention** – their *roots*. This not only improves the search, it also reduces the size of the indexes containing our vectors.

A **telephone** is the name of a device. Many people would use the word **phone** for the same device. In other words, **phone** is a synonym of **telephone**, or **telephone** and **phone** are synonymous. A document on the invention of the telephone may contain the word **phone** instead of **telephone**. Extending the query with a list of synonyms may help finding relevant documents.

How do we know that a certain technique is better than another one in text retrieval? A *recall* is the number of relevant documents returned by the text retrieval system divided by the number of relevant documents in the collection. A *precision* is the number of relevant documents returned by the system divided by the number of the documents returned by the system. Coverage is the number of documents for which the system can find the similarity measure divided by the total number of documents in the collection. For example, documents containing images, maps, movies, etc. cannot be handled by the pure text retrieval system, so they lower the coverage. A better recall usually means lower precision, and vice versa.

Is the space vector model equivalent to the ideal system? Imagine that the most relevant document has the phrase:

Graham Bell filed a patent for the telephone.

FOLLOW-UP EXERCISES:

Ex. 7

Work in pairs. Looking only at your notes retell the text you've read to your partner. Your partner should take notes. Student A should start.

Ex. 8

When you have finished, read both texts carefully. Complete your notes with important details that your partner missed.

Ex. 9

With your partner take turns and try to answer the questions without looking back at the unit:

1. What is the difference in the way you get information now and 20 years ago?
2. What is the problem with information now?
3. What is the relation between information retrieval and text retrieval?
4. What does the term *bag of words* mean?
5. What are stopwords?
6. What is a recall?
7. How does stemming influence recall and precision?
8. What is the role of synonyms in searching for information?
9. What is coverage?

Ex. 10

Discuss your answers to the questions with the whole class.

Ex. 11

Look at ADDITIONAL MATERIAL (Internet of the Future) at the back of the book – Do Exercise 1 and 2.

(◕‿◕✿)(◕‿◕✿)(◕‿◕✿)

Are Computers Men or Women?

A language teacher was explaining to her class that in French, nouns, unlike their English counterparts, are grammatically designated as masculine or feminine.

“House” in French is feminine – “la maison,” “pencil” in French is masculine “le crayon.”

One puzzled student asked, “What gender is computer?” The teacher did not know, and the word was not in her French dictionary.

So, for fun, she split the class into two groups appropriately enough, by gender and asked them to decide whether “computer” should be a masculine or a feminine noun.

Both groups were required to give four reasons for their recommendation.

The men’s group decided that computer should definitely be of the feminine gender (“la computer”), because

- no one but their creator understands their internal logic,
- the native language they use to communicate with other computers is incomprehensible to everyone else,
- even the smallest mistakes are stored in long term memory for possible later review,
- as soon as you make a commitment to one, you find yourself spending half your paycheque on accessories for it.

The women’s group, however, concluded that computers should be masculine “le computer”) because:

1. In order to do anything with them, you have to turn them on.
2. They have a lot of data but still can’t think for themselves.
3. They are supposed to help you solve problems, but half the time they ARE the problem.
4. As soon as you commit to one, you realise that if you had waited a little longer you could have gotten a better model.

(◕‿◕✿)(◕‿◕✿)(◕‿◕✿)

Unit 10

Introduction To Game Design and Development

LEAD-IN:

Ex. 1

Match the words 1-12 with their definitions:

1. TO IMPLEMENT	a) A false or mistaken view, opinion, or attitude.
2. TO IMMERSE	b) Not based on fact; unreal.
3. TO RECUR	c) Logical; consistent and orderly.
4. COHERENT	d) To put into practical effect; carry out.
5. HANDICAP	e) To engage wholly or deeply; to throw into.
6. NOTIONAL	f) To attack in such a manner that no escape or defence is possible.
7. CHECKMATE	g) To happen again.
8. MISCONCEPTION	h) Disadvantage, harm.
9. PARTICIPANT	i) The end of something in time; the conclusion; result.
10. SUBSET	j) A set whose members are all contained in another set.
11. PACE	k) Someone who takes part in an activity.
12. TERMINATION	l) The rate of speed at which a person, animal, or group walks or runs.

Ex. 2 ***T20***

Repeat the words below after the recording:

interaction, *interface*, *therapy*, *disobliging*, *approach*, *participant*, *notional*, *subset*, *to implement*, *virtual reality*, *to immerse*, *doubtful*, *pace*, *to advance*, *genre*, *quantifiable*, *unambiguous*, *checkmate*, *coherent*, *termination*, *figurine*, *novice*, *to recur*, *rigorous*, *handicap*, *misconception*

Ex. 3

Work in pairs. Discuss the questions below with your partner:

1. What are video games?
2. What can video games be used for?
3. What is game design?
4. What kind of skills does game design require?
5. What does the process of game design need?
6. Is it possible for one person only to design a complicated game?
7. What is the most important thing in game design?

TEXT-BASED TASKS:

Ex. 4 ***T21***

You are going to hear a text about video games. Listen to the recording and note down the answers to the questions from Ex. 3.

Discuss your answers with your partner. When you have finished, look at the tape-script at the back of the book and check your answers.

Ex. 5

Read the "Introduction" below and decide whether the following sentences are true or false according to the text:

1. Humans play games on the basic level.
2. A pretender knows that the notional reality is different from the real world.
3. Games do not include rules.
4. Playing with puzzles is more structured than playing games.

INTRODUCTION

Constantly growing popularity of video games is the effect of natural human desire to play. Generally, **play** is a wide category of non-essential, and usually recreational, human activities that are often socially significant as well. **Pretending**, a very important element of playing, means the mental ability to establish a notional reality, which the pretender knows is different from the real world. The pretender can create, abandon, or change it at will

Humans can entertain themselves by playing on three different levels, depending on their growing maturity. On the basic level humans play **toys**, which come with neither *rules* about the right way to play with them, nor with a particular *goal* to be achieved. Even toys depicting some real objects give only a disobliging suggestion for the way and goal of a play. Giving a toy a distinct goal to achieve brings it to the next level on which the toy becomes a **puzzle**. Puzzles usually have no rules telling us how to achieve the goal. Many approaches are allowed though some of them may turn out to be fruitless. **Games** include not only rules but also at least one goal, unlike toys and puzzles, which have none or only one of these attributes. Though playing a game still requires pretending, it is far more structured than playing with toys and puzzles, so in effect it requires more maturity. Playing **multi-player games** also requires some social cooperation of the players.

There have been many efforts to define the word *game* over the years. Despite their differences they have some elements in common, which are: *rules*, *goal*, *play* and *pretending*. This may lead to a non-rigorous definition of a game:

A **game** is a type of play activity, conducted in the context of a pretended reality, in which the participant(s) try to achieve at least one arbitrary, non-trivial goal by acting in accordance with rules.

Ex. 6

Check the sentences with your partner, and then with the whole group.

Ex. *7*

Work in groups of three. Decide who is A, B or C.

Student A read text A, student B read text B, student C read text C. While you read, underline the key information. When you have finished, write a short summary of your text.

TEXT A

TRADITIONAL AND VIDEO GAMES

Video games are a subset of a universe of all games that are mediated by a computer. The term computer covers here all *processor devices* such as personal computers, video and hand-held consoles, mobile and smart phones, and others. A game designer should be able to design any kind of games – traditional, card or video ones. However, designing video games requires taking into account some of their specific demands and advantages over traditional games.

Firstly, video games usually **hide the rules**, as they do not require written ones. Though the rules still exist, it is the machine that implements and enforces them for the players and controls all the game mechanics. In many games players do not even have to know the rules exactly, though they still need instructions how to play. The disadvantage of this attitude is that players cannot optimize their actions until they become familiar with all the rules by playing the game. The preferred way to help the player is to give him some kind of hints instead of making him work out the rules by means of trial and error, which can be very frustrating.

The second important difference is the possibility of **presenting a game world**. As a game world is usually fictional, creation of the pretended reality takes place in the imagination of the players. Traditional games can help this process with colourful boards, cards, figurines, booklets and other accessories, whereas video games can immerse the player far deeper, presenting game worlds very close to those known from television or films. Modern video games use interactive multimedia presentation with realistic pictures, animations, movies, music, dialogues, and sound effects. Some game designers go a little further, creating very realistic *augmented reality* and *virtual reality* games. In augmented reality games computers are used together with real-world activities to play the game. It is possible for a player to use cellular phones, video cameras, GPS devices, as well as the Internet applications. In contrast, virtual reality games try to separate the player from the real world immersing him or her as deep as possible in a fictional, virtual world. This goal can be achieved by "surrounding" the player's senses with specialized hardware such as VR *head mounted display* (HMD), earphones, *VR gloves*, *haptic devices* and others.

As the third significant difference, video games bring **artificial intelligence** (AI) into the world of games. First of all, this means that a computer can play the role of the other player, so it is possible to play alone the games formally designed for two or more players. Although the social correctness of this attitude may be doubtful, it offers almost all kinds of games for a single player. The AI components in video games are responsible for creating artificial opponent(s) that could play as competently as human ones.

And the last meaningful difference is the possibility of **setting the pace** by a computer. This means that in many games it is the machine that decides about the action speed and advances. Of course, there are still many games in which the player has enough time to make a move, but controlling the pace by a machine brings to life new genres of games such as real time, action, etc.

TEXT B

FUNDAMENTAL CONCEPTS OF GAMES

Goal. Each game should have at least one goal that can be arbitrarily defined by a game designer. As each game should be a challenge, the object of a game should be non-trivial. A goal is usually quantifiable (e.g. find all diamonds) but it can also be not quantifiable (e.g. manage the Zoo without going bankrupt) or even not achievable in never-ending games such as *Space Invaders* or *Breakout*.

Termination Condition is a simple or complex rule that determines when the game is over. The termination condition usually belongs to one of the three classes:

- *victory condition*, that is an unambiguous situation in the game, in which one or more players are declared to be winners, such as checkmate in *Chess*;
- *loss condition*, that is an unambiguous situation indicating which player has lost, e.g. losing money and resources in simulators such as *SimCity* or *Theme Park*
- *time out* (*limit*), that is a defined period of time which passed in real world or within the game world (days, years, epochs etc.), e.g. four quarters in a basketball sport game.

Rules are definitions and instructions that the players agree to accept for the duration of the game. Rules should be *comprehensible* for all players, *coherent*, e.g. with no conflicts among them, and *unambiguous* to prevent arguments. Rules serve several functions and they are used to establish such elements as:

- *the goals* of the game;
- *the semiotics* of the game, that is the meaning and relationships of various abstract symbols in the game and parallels to the real world symbols;
- *the sequence* of play, that is the progression of activities that make up the game;
- *the gameplay* – challenges and actions of players;
- *the termination conditions*;
- *meta-rules* which are rules that define when other rules may be changed or when some exceptions are allowed.

Gameplay consists of *challenges* to be overcome to achieve the goal of the game and allowed *actions* used to address these challenges**Błąd! Nie można odnaleźć źródła odwołania.**. A **challenge** is simply any task set for the player that is non-trivial to accomplish and requires mental or physical effort. Challenges may be categorized using several aspects:

- *simple* versus *complex*;
- *unique* (e.g. level's boss), *recurring* (e.g. numerous enemies) and *continuing* (e.g. avoiding being hit);
- *direct* (leading to the goal) and *optional* (e.g. practice, better score);
- *explicit* (defined by rules) versus *non-explicit* (e.g. in logical games);
- *easy* (for novice or youngest players) versus *difficult* (for veterans and hard-core players).

Actions are any activities that can be performed by a player within a game. They are formally divided into *accepted* actions (required and optional) and *not accepted* ones such as game bugs or cheating. The latest tendency is to give a player the largest possible set of actions allowing each action that is not prohibited.

Fairness is the guarantee that, at the beginning of the game, all players in a multi-player game have equal chances of winning. It is not an essential element of a game but a culturally constructed notion. To achieve fairness, the game should be *symmetrical* or well-*balanced* with proper *handicaps* for particular players.

TEXT C

DESIGN PROCESS

There are many factors that influence designing and producing of an enjoyable and commercially successful game. These factors include a fundamental concept, or an idea an experienced development team but also marketing, distribution and others. But there is also one additional factor that can be the key to success. This key is **the player-centric approach** in game design. It is the philosophy of design which obliges a designer to entertain a class of representative players and to empathize with them. E*ntertainment* has priority over the creative expression of the designer and *empathization* means to mentally place the designer in the position of a representative player, make design decisions from a player's point of view, and understand the player's preferences, motivations, and reactions.

There are two fundamental misconceptions that must not occur during the design process. The first misconception is considering the designer as a typical player instead of defining him or her as a representative member of the target group of customers on the market. The second misconception is treating the player as the designer's opponent, which leads to designing too difficult games. There are also other "temptations" during the design process that could prevent games from reaching "harmony". The first example is the creation of *market-driven games* that are built for a particular market including certain elements in their design to increase sales within that market. The second one may be *designer-driven* games, in which a designer controls each aspect of game creation, ignoring play-testing and team work.

The game design process consists of three stages:

- *concept stage*, at which fundamental assumptions concerning the game concept, audience, genre etc. are agreed;
- *elaboration stage* means iterative adding of the design details and refining decisions through prototyping and play testing;
- *tuning stage* covers only small adjustments to polish the game, but no new features can be added.

Though all stages are very important, the concept stage may determine the success or failure of the final result, which makes it so crucial and important in the design process.

Ex. 8

In your groups of three tell the other students what your text is about. Student A should start. You can use your summaries if necessary.

Ex. 9

Read the two texts you haven't read before to check whether you managed to understand your friends correctly.

FOLLOW-UP EXERCISES:

Ex. 10

Using the jumbled words make up correct sentences. Use all the words.

1. can themselves by on three maturity levels depending playing humans on their growing entertain different

2. level toy distinct giving a toy a goal to brings it to the on which the becomes a puzzle next achieve
3. cooperation games playing players multi-player requires social of the
4. games are a of a computer universe of all that are video mediated by a subset games
5. place as a game is fictional creation imagination world of the reality takes in the of the players pretended usually
6. defined game have at one goal that can be by a game designer each least arbitrarily should
7. players are definitions game and instructions that the agree to for the duration of the rules accept
8. performed activities actions are any can be by a within player a game that
9. successful there many and factors are that designing producing an enjoyable and commercially game influence of
10. occur fundamental there design are two that not during must the process misconceptions

Ex. 11

Check your sentences with your partner. When you have finished, find them in the unit texts.

Ex. 12

Work in a group of 3 or 4. Plan a game suitable for young adults. Your game should fulfil the criteria mentioned in the unit.
Present your game to the group. The group listens and decides whether the criteria were met.

USEFUL PHRASES: The rules would be… The goal would be… It can be played on different levels:… It is possible for the player to use… The challenge would be to… Interactive multimedia presentation can be used to show…

(◑‿◐) (◑‿◐) (◑‿◐)

Programmers' Wisdom

Every program has at least one bug and can be shortened by at least one instruction – from which, by induction, one can deduce that every program can be reduced to one instruction which doesn't work.

(◑‿◐) (◑‿◐) (◑‿◐)

Unit 11

Databases and Database Management Systems

LEAD-IN:

Ex. 1

Match the words 1-12 with their definitions:

1. TUPLE	a. The act of putting one thing into another.
2. TABULAR	b. Add by linking or joining so as to form a chain or series.
3. REDUNDANCY	c. An ordered set of *n* elements.
4. INSERTION	d. Acting together; at the same time.
5. CONCATENATE	e. Occurring or appearing again or repeatedly.
6. CONCURRENCY	f. A property, quality, or feature belonging to or representative of a person or thing.
7. ATTRIBUTE	g. Organized as a table or list.
8. RECURRENT	h. Unnecessary repetition of information.
9. DOMAIN	i. A territory over which rule or control is exercised. The set of all possible values of an independent variable of a function.
10. UNAUTHORISED	j. An act of breaking rules.
11. VIOLATION	k. A line of text serving to indicate what the passage below is about; a block of data on a tape or disk providing information about the size, location, etc., of a file.
12. HEADER	l. Not having official authority or sanction; not allowed to do something.

Ex. 2 ***T22***

Repeat the following words after the recording:

tuple, *efficient*, *domain*, *tabular*, *attributes, identifier*, *integrity*, *implicitly*, *redundancy*, *normalisation*, *insertion*, *violation*, *convenient*, *querying*, *concatenate*, *unauthorized*, *failure*, *concurrency*, *rigid*, *composite*, *recurrent, header*

TEXT-BASED TASKS:

Ex. 3

Read the text and complete the gaps with the following words:
ATTENTION: there are more words than you need.

system, *applications*, *databases*, *systematically, processing, apply, software*, *related*, *base*, *foundations*

Modern computer (1) ………. require efficient and reliable (2) ………. of large amounts of information. To fulfil this demand, (3) ………. are used. Informally speaking, a *database* is a collection of (4) ………. information organised in accordance with some *logical model*. This collection is managed by a specially dedicated (5) ………. called *Database Management System* (DBMS). If there are multiple databases on a single computer (6) ………. (which is normal), then we have a *database system*. Currently, the most widely used database model is a *relational model*. Its origin dates back to the beginning of the 70s of the last century, when E.F. Codd announced mathematical (7) ………. of this model called the *relational algebra*. This model became the (8) ………. for building *relational databases* (RDBs) and computer systems that manage them.

Ex. 4 ***T23***

Listen to the recording. Define in one sentence what is being described.

Ex. 5 ***T23***

Listen again and answer the following questions:

1) What is an essential element of database according to Codd's theory?
2) What is the restriction on mathematical relation imposed by Codd?
3) What is the first normal form?
4) What does the picture present?
5) What information about students does the database contain?
6) What do the rows of the table represent?
7) What do the columns of the table represent?

Check your answers with your partner, and then with the whole group.

Ex. 6

Work in pairs. Decide who is A and B.
Student A read text A.
Student B read text B.
Note down the most important information.

TEXT A

A database relation must also satisfy a condition that does not refer to mathematical relations. A database relation must have a *primary key* that uniquely identifies relation tuples. A primary key is typically a single attribute of the relation (a *simple primary key*). So it is the case in the relation STUDENTS, where the attribute Album_No is the identifier (and the primary key). There are also relations in which a primary key consists of several attributes. Such a key is called a *composite primary key*. Take for example the set of all cars. If you would like to create a database relation that stores information about the cars, then if we are confined to only one country, the primary key could simply be a registration number of a car. However, if this relation were to keep data about cars from various countries, the registration number would not be enough, and the primary key should be extended with some other attribute, for example, the symbol of the country.
But let us return to our example of the college students. Now imagine that in our database we want to store not only personal data of students, but also their grades they had obtained for different subjects. To do this we must create a new relation which we can call GRADES, that could look like the following figure.

GRADES

Album_No	Course	Mark	Semester
143879	Maths	4.0	1
143879	English	5.0	1
143879	Programming	4.5	2
654889	Maths	3.0	1
654889	Programming	3.5	2
324672	Physics	5.0	1

TEXT B

Note that tuples in the relation GRADES must be somehow related to tuples in the relation STUDENTS, so that it is known which student is associated with a specific grade. The attribute Album_No in the relation GRADES serves this purpose, with the same meaning as in the relation STUDENTS (actually, this attribute could be called differently in GRADES, because in a relational database the same name does not necessarily mean the same thing). Such a linking attribute (or a set of attributes) is called a *foreign key*. The value of a foreign key must be equal to some value of the primary key in the relation that it refers to. Otherwise the database would lose an important feature called *data integrity*. In this way we created a *relationship* between the two relations. Note also that the attribute Album_No in the relation GRADES is not a good primary key for this relation, because its values, for obvious reasons, can be repeated in different tuples. However, if we attach the attribute Course to the attribute Album_No, we obtain a good composite primary key (we implicitly assume that the relation GRADES stores only final grades and that the names of subjects in different semesters are different).

Ex. 7

Retell your text to your partner without looking back at it. You can use your notes.

Ex. 8

Work in pairs. Read the text below and write 10 true/false sentences to the text.
When you have finished, work with another pair and quiz the other students using your sentences.
Finally check with the text.

At this point you could ask the following question: Why not include the names of students in the relation GRADES? After all, if we wanted to print a report on achievements of a student with his or her name (as it is usual), it would be much easier for us to have the names in the relation GRADES because we would need to retrieve data from only one relation, not from two relations. You are right, as indeed it would be easier, but in that case a very important requirement for "well-formed" relational databases would not be fulfilled. According to this requirement, a relational database should have no redundancy. In other words, any piece of information should be recorded only once. If we put names of students into GRADES, the information that a student has been assigned a specific album number would be repeated as many times as the number of the student's grades is, and additionally this information would be stored in the relation STUDENTS. You might ask further: Does it really matter? The answer is yes, it does, for two reasons. First, to include the names of students into GRADES would be a waste of valuable resource, which is the memory occupied by the database. But perhaps the second reason is more important. Imagine that you accidentally mistyped the name of the student, for example, Kovalski instead of Kowalski was entered into STUDENTS and, even worse, the mistake was repeated in the relation GRADES and perhaps in other relations in the database. After the discovery of this mistake we need to make corrections in all the places where this name is stored. So we have to remember these places, but there can be a lot of them. If you forget even one, the database will contain conflicting information, which means that it loses its integrity, which in most cases is catastrophic.

Eliminating redundancy in a relational database is called *normalisation* of a database. Insertion of the names of students into the relation GRADES would cause a violation of the so-called *second normal form* (2NF). There are other normal forms that refer to other undesirable phenomena that may occur in relational databases.

Databases would not be so useful without the possibility of efficient and convenient access to the data stored. For this a standard language called SQL (Structured Query Language) was devised and it is devoted to querying relational databases. SQL allows you to fetch data from one or more relations in a manner consistent with the relational algebra. Thus it is possible to use SQL to fetch specific tuples from a relation (*Selection*) or specific attributes in these tuples (*Projection*), to fetch tuples from several identically built relations (*Union*), to fetch only those tuples from one relation that do not exist in another relation (*Minus*), and concatenate tuples from two relations (*Join*). For example, an SQL statement that prints a report on all grades of Jan Kowalski (Album_No 143879) from our database, accompanied by the surname and the name of the student, in chronological order, and within one semester, alphabetically by names of subjects, may look like this:

```
SELECT DEGREES.Album_No, STUDENTS. Surname, STUDENTS. Name,
      DEGREES.Course, DEGREES.Mark, DEGREES.Semester
 FROM DEGREES JOIN STUDENTS
      ON DEGREES.Album_No = STUDENTS.Album_No
 WHERE DEGREES.Album_No = “143879”
 ORDER BY DEGREES.Semester, DEGREES.Course
```

Executing SQL statements is one of the major tasks for DBMS. This is not an easy task when you consider the fact that relations in a database may contain millions of tuples, and an impatient user wants to have the results of a query immediately! DBMS also performs other tasks. The most important of them, in addition to storing and retrieving data, are: protection of data against unauthorized access, preserving data integrity in case of failures, appropriate concurrency support in case several users try to obtain access to the same data at the same time, and some other, more technical tasks.

Relational model is the most common, but not the only logical model used in databases. Another model is the object-oriented model. Databases organised according to this model are called object-oriented databases (OODBs). Other databases called object-relational databases (ORDBs), that extend the relational model with some object-oriented features, are becoming more and more popular. They depart from the rigid rules of the first normal form and introduce complex (non-atomic) values, such as collections, lists, tables and records, new data types, such as multimedia data types, they are also able to store pieces of code called *stored procedures*, etc. In the near future, development of database theory and practice will definitely run in that direction. This will make databases even more useful and more powerful tools for implementation of today’s data-intensive computer applications.

Databases are also a subject of very intensive and challenging theoretical studies. Algorithms for query processing, extension of relational models towards recurrent relations, methods of indexing of various kinds of information (including spatial and temporal data) and mobile database systems are topics that continuously form an important area of contemporary scientific research of computer scientists.

FOLLOW-UP EXERCISES:

Ex. 9

Explain the following abbreviations with your partner. When you have finished check in the text from exercise 8.

1. DBMS
2. RDB
3. OODB
4. ORDB
5. SQL
6. 2NF

Ex. 10

Choose the appropriate form of the word in brackets (present perfect or past simple:

1. All samples (failed/have failed) under the first test condition
2. Recent studies (have shown/showed) that this solution is the most feasible one.
3. Research conducted in the 1990s (has shown/showed) that more powerful computers were needed to perform these calculations.
4. The studies (were not completed/have not been completed) yet.
5. Two years ago researchers (demonstrated/have demonstrated) the link between memory capacity and the speed of picture processing.
6. Over the past six years there (were/have been) several exciting developments in the field of mechatronics.
7. Group 1 (completed/have completed) the first assignment, but Group 2 (didn't start/haven't started) yet.
8. In the past five years several manufacturers (developed/have developed) the technology of speech recognition.
9. Few researchers (investigated/have investigated) this possibility so far.

Ex.11

Work in groups of three.
Design a relational database for passenger cars in Europe. Assume that each car has one owner, but a person may own several cars.
Present your ideas to the whole group.

@(ᵕ.ᵕ)@ @(ᵕ.ᵕ)@ @(ᵕ.ᵕ)@

Right Click

Tech Support: "I need you to right-click on the Desktop."
Customer: "Ok."
Tech Support: "Did you get a pop-up menu?"
Customer: "No."
Tech Support: "Ok. Right click again. Do you see a pop-up menu?"
Customer: "No."
Tech Support: "Ok, sir. Can you tell me what you have done up until this point?"
Customer: "Sure, you told me to write 'click' and I wrote 'click'."

@(ᵕ.ᵕ)@ @(ᵕ.ᵕ)@ @(ᵕ.ᵕ)@

Unit 12

Computational Problems and Algorithms

LEAD-IN:

Ex. 1

Match the words 1-12 with their definitions:

1. STRING	a. Broken – not continuous.
2. HALTING	b. A constant number that serves as a measure of some property or characteristic.
3. PRIMALITY	c. A positive or negative whole number or zero.
4. COEFFICIENT	d. A number used to divide another.
5. DIVISOR	e. Able to be done or put into effect; possible.
6. FEASIBLE	f. An expression of two or more terms.
7. POLYNOMIAL	g. A set of consecutive characters.
8. INTEGER	h. The property of being a prime number.
9. PLANE	i. Close or similar.
10. ROUTE	j. The property (of a problem or difficulty) that makes it impossible to work out.
11. APPROXIMATE	k. A road, course, or way for travel from one place to another.
12. UNSOLVABILITY	l. A surface containing all the straight lines that connect any two points on it. A flat or level surface.

Ex. 2 ***T24***

Repeat the words below after the recording;

sequence, *string*, *combinatorial*, *unsolvability*, *primality*, *divisor*, *plane*, *convex*, *integer*, *coefficient*, *exponentially*, *feasible*, *algorithm*, *procedure*, *polynomial*, *halting*, *proof*, *approximate*, *route*, *efficient*, *proportional*, *matrices*, *explicitly*

Ex. 3

Discuss the following questions with your partner:

1. What is an algorithm?
2. What are the synonyms of the term “algorithm”?
3. What does it mean that a problem is solvable algorithmically?
4. What is computability theory about?
5. What is “halting problem”?
6. What is a “semidecidable problem”?
7. Is it possible to write an optimal chess-playing program?
8. Is there an infinite number of ways of arranging chess pieces on the board?

9. What could such a program consider?
10. What is the subject of study of “computational complexity”?

TEXT-BASED TASKS:

Ex. 4

Read the text below and check your answers from Exercise 3.

The term “algorithm” is a central notion of this unit. Informally, an *algorithm* is a finite sequence of well-defined operations, which transforms any proper string of input data into a proper string of output data and halts in a finite time. In practice, when you see the word “algorithm” you may think: recipe, procedure, program, method, strategy, technique or computation.

In this unit we consider algorithmic problems of combinatorial nature. To say that a problem is solvable *algorithmically* (or that it is *algorithmic*) means that a computer program can be written which will produce the correct answer for any input if we let it run long enough and allow it as much storage space as it needs. In the 1930’s, before the advent of computers, mathematicians worked hard to formalize and study the notion of an algorithm. Much of the emphasis in the early work in this field, called *computability theory*, concentrated on describing or characterizing problems that could be solved algorithmically and exhibiting some problems that could not be. One of the important negative results was A.M. Turing’s proof of unsolvability of the so-called “halting problem”. The *halting problem* involves deciding whether an arbitrarily given algorithm (or computer program) will eventually halt (rather than get into an infinite loop) while working on a given input. A computer program that solves this problem cannot exist. Moreover, in the class of unsolvable problems there exists a subclass of *semidecidable problems*, for which “yes” can be obtained in a finite time, while there is no such guarantee in the case of negative answer.

Although computability theory has obvious and fundamental implications for computer science, knowing that a problem can be solved theoretically on a computer is not sufficient to tell us whether it is practical to do so. For example, it is possible to write an optimal chess-playing program. There is only a finite number of ways of arranging the chess pieces on the board, and a game must terminate after a finite number of moves. The program could consider each of the computer’s possible moves, each of its opponent’s possible responses, each of its possible responses to those moves, and so on, until each sequence of possible moves reaches an end. The number of distinct arrangements of pieces on the board is about 10^{50} (10 to the power of 50), and a program that would examine them all would take many billion years to run. It is not surprising that such programs are not used! Similarly, it is not hard to give an example of a problem that needs, say 10^{80} (10 to the power of 80) operations to do. For example, from the very definition of primality that an integer n is prime if it has no divisor between 2 and $\sqrt{n}$ (square root of n), one can evolve a simple brute force test for primality: just check whether any integer between 2 and $\sqrt{n}$ actually divides n. If n is a 160-digit prime number, then we would have to do about 10^{80} divisions at worst. This would take longer than the remaining lifespan of our universe, even on an impossibly fast computer! Note that the last number estimates the number of all atoms in the observable universe.

There are numerous problems with practical applications that have algorithmic solutions, but for which the time and storage requirements are much too great for the corresponding programs to be of practical use. Clearly, the time and space requirements of a computer program are of practical importance. They are the subject of theoretical study in the area of computer science called *computational complexity*.

Ex. 5

Work in pairs. Student A – cover the text, student B – ask your partner the questions from exercise 3 and check the answers. Change roles.

Ex. 6 ***T25***

You are going to hear a text about combinatorial problems. Listen to the recording and note down the categories of combinatorial problems.

Ex. 7 ***T25***

Listen again. Note down the key information concerning the categories.
Compare your notes with your partner and then with the tapescript at the back of the book.

Ex. 8

Work in pairs. Student A – read text A, student B – read text B below. While you read note down the key information.

TEXT A

All combinational problems can be divided into five categories as follows.

1. *Nonalgorithmic problems*. These problems cannot be solved by algorithms. An example of such a problem is the halting problem. Another interesting example is the so-called *tiling problem*.

   Example *Infinite tiling*

   Suppose we are given infinitely many identical floor tiles, each shaped like a regular hexagon.
   The whole plane can be tiled with hexagons, i.e. we can cover the plane with no empty spaces left. This can also be done if the tiles are identical rectangles, but not if they are regular pentagons. This fact was already known to Pythagoras. Suppose we are given a certain polygon, not necessarily regular and not necessarily convex, and suppose we have infinitely many identical tiles in that shape. Can the whole plane be tiled with that shape, or not? That question has been proved to be computationally unsolvable.

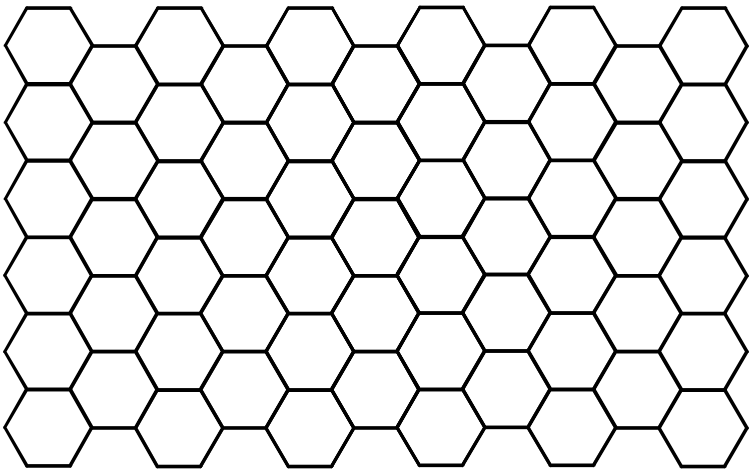


Fig. 10.1. An example of infinite tiling

Yet another example is the problem of solving Diophantine equations. A *Diophantine equation* is a problem of finding integer solution of the algebraic equation with integer coefficients. For instance, $x^n+y^n = z^n$ (x to the power of n plus y to the power of n equals z to the power of n) has infinitely many integer solutions for $n = 2$ (n equals 2), but has no such solution for any $n > 2$ (n greater than 2) which follows from the famous Andrew Wile's proof of the *Fermat*'*s Last Theorem*. In 1900 David Hilbert asked whether there exists a "finitary way" to solve the problem of Diophantine equations. In 1970 it was shown that this problem is computationally unsolvable.

2. *Presumably nonalgorithmic problems*. For such problems no finite algorithm is known, but there is no proof that such an algorithm cannot exist. Many examples of such problems come from open Diophantine problems. Another interesting example is the so-called *Collatz problem*.

Example *Collatz problem*

Given a computer procedure as follows:

```
procedure Collatz;
begin
     read(k),
     repeat
          if even(k) then k := k/2 else k := 3*k+1
     until k = 1
end;
```

The question is: for a given integer k determine whether this procedure terminates.

So far the problem has not been solved, though the question has been answered in affirmative for all k £ $5 \leq 10^{18}$. Therefore, the following decision problem: given integer k, determine whether the procedure *Collatz* loops indefinitely, might be nonalgorithmic.

3. *Nonpolynomial problems*. These problems cannot be solved by algorithms running in time bounded by a polynomial in the size of the input. A good example of such a problem is listing mathematical objects whose number grows exponentially. For instance, for a given integer *n* we might wish to generate all arrangements of *n* elements.

Example *Permutations*

```
procedure Permute(k,n);
begin
    for each possible position in the list L of size n do
    begin
        insert k into the given position;
        if k = n then write (L)
            else Permute (k+1,n);
        remove k from the given position
    end
end;
```

Since there are *n*! such arrangements, the running time of *Permute* must grow faster than any polynomial in *n*.

TEXT B

4. *Presumably nonpolynomial problems*. No algorithm is known to solve these problems in polynomial time, though on the other hand, there is no formal proof that such an algorithm cannot exist. Nevertheless, computer scientists are generally pessimistic about the possibility of designing any algorithm of this kind. Most of the presumably exponential problems belong to the class of NP-complete problems. Perhaps one of the best known NP-complete problems is the Traveling *Salesperson Problem* (*TSP*). In this problem, a traveller wishes to visit a number of cities and return to the starting point in such a way that each city is visited exactly once, and the total distance covered is as short as possible. Although there are several ad hoc procedures which can be used to find approximate solutions, a brute force solution effectively involves looking at all possible routes and choosing the shortest. This approach is feasible if there are 10 cities, since the number of possible routes is then 9!/2 = 181440, and a computer working through these at the rate of one million routes per second will find the best route in about 0.18 seconds. On the other hand, if there are 20 cities, then the number of possible routes is about $6 \cdot 10^{16}$, which would take almost 2000 years to work through! Of course, there are more efficient methods but they still require checking a portion of *n*! possible routes. In theory, it may be possible that the *TSP* can be solved by a polynomial-time algorithm. However, in practice, most computer scientists are pessimistic about the possibility of inventing such an algorithm.

5. *Polynomial problems*. Such problems have algorithms that run in time bounded by a polynomial in the size of input. For example, *n* items can be sorted into order in the worst-case in time proportional to $n^2/2$ (e.g. *bubble sort*). Another example is a multiplication of square matrices that can be done in time proportional to n^3 though there are much faster algorithms. It is interesting to note that for some problems there is a polynomial-time algorithm even though we do not know it explicitly.

FOLLOW-UP EXERCISES:

Ex. 9

In the same pairs retell your partner the most important information from your text. Refer to your notes if necessary. Change roles.

Ex. 10

Read the other text and check if you understood your partner's version correctly.

Ex. 11

Individually prepare 8 true/false sentences for both texts (4 for each text).
Make groups of 4. Say your sentences, and your partners will have to decide whether they are true or false.

Ex 12

Make questions to the underlined words in the sentences. When you have finished, check with the whole group.

1. In this unit we consider algorithmic problems of combinatorial nature.
2. It is possible to write an optimal chess-playing program.
3. It is possible to write an optimal chess-playing program.
4. There is only a finite number of ways of arranging the chess pieces on the board.
5. The program could consider each of the computer's possible moves.
6. Much of the emphasis in the early work in this field concentrated on describing or characterizing problems.
7. This will produce the correct answer for any input if we let it run long enough and allow it as much storage space as it needs.
8. Computability theory has obvious and fundamental implications for computer science.
9. This computer program is connected with this issue.
10. More advanced computer programs do not exist.

Ex. 13

Work in groups of four. Design a project (an idea) of an algorithm. Make assumptions, discuss the subsequent steps, input and output. Draw a diagram of your algorithm. Present it to the whole group.

USEFUL PHRASES:	
The program could consider…	A good example is/would be….
It is interesting to know that…	On the other hand…
In theory it may be possible that…	However, in practice…
Yet another example is…	

Appendices

Appendix 1A

The *input and output modules* (I/O modules) provide the means for accepting external information and passing data to peripheral devices. To facilitate adaptation of their functionalities to the application they are serving, I/O modules provide 1) ____________________ . Typical examples of I/O modules are analogue-digital converters used for data input, as well as 2) ____________________ and digital-analogue converters to handle data output. Timers are employed for counting external pulses or measuring intervals between events. Communications with external components and/or other microcontrollers can be implemented by means of serial and parallel interfaces. One example of such communication is 3) ____________________ with extraneous microcontrollers by means of the CAN bus. Depending on the requirements of a given application, 4) ____________________ may be integrated in a microcontroller.
The microprocessor processes the inbound data entering via the input modules and controls the 5) ____________________ . Its registers provide storage for operands, results, and addresses.

The input/output modules provide for the input of external signals and the influencing of manipulated variables by means of 6) ____________________ signals. In this way, the I/O modules comprise a link between the microprocessor and its environment. In addition to a connection to the internal data bus of the microcontroller, each I/O module features external connections termed *pins*, which are suitable for connecting 7) ____________________ , for example.

Its principal tasks may be divided as follows:
- communications with the internal data bus of the microcontroller,
- communications with the environment,
- data storage,
- watchdog functions and timer control,
- fault recognition.

Input/output module types are differentiated as follows:
- *Isolated I/O.* Two separate address areas accommodate the microprocessor memory and the memory designated for the I/O modules. Because only special instructions can be used for I/O modules, their programming is subject to 8) ____________________ .
- *Memory mapped I/O.* Microprocessor and I/O modules share a memory area with a common address range. 9) ____________________ has the advantage of allowing the large number of instructions dedicated to addressing microprocessor memory to be used for the I/O modules. One drawback of the occupation of address space does exist, especially affecting microprocessors utilizing a word length of 4 or 8 bits. By contrast, state-of-the-art microprocessors handling word lengths of 16 or 32 bits operate only in conjunction with memory mapped I/O architectures.

Appendix 1B

The 1) ____________________ provide the means for accepting external information and passing data to peripheral devices. To facilitate adaptation of their functionalities to the application they are serving, I/O modules provide limited programming options. Typical examples of I/O modules are analogue-digital converters used for data input, as well as pulse width modulation modules and digital-analogue converters to handle data output. 2) ____________________ are employed for counting external pulses or measuring intervals between events. Communications with external components and/or other microcontrollers can be implemented by means of 3) ____________________ . One example of such communication is digital data communications with extraneous microcontrollers by means of the CAN bus. Depending on the requirements of a given application, additional functions may be integrated in a microcontroller.

The microprocessor processes the 4) ____________________ entering via the input modules and controls the flow of data. 5) ____________________ provide storage for operands, results, and addresses.

The input/output modules provide for the input of external signals and the influencing of manipulated variables by means of output signals. In this way, the I/O modules comprise a 6) ______________________ between the microprocessor and its environment. In addition to a connection to the internal data bus of the microcontroller, each I/O module features external connections termed *pins*, which are suitable for connecting sensors and actuators, for example.

Its principal tasks may be divided as follows:

- communications with the internal data bus of the microcontroller,
- communications with the environment,
- data storage,
- watchdog functions and timer control,
- fault recognition.

Input/output module types are differentiated as follows:

- *Isolated I/O.* Two separate address areas accommodate the 7) ____________________ and the memory designated for the I/O modules. Because only special instructions can be used for I/O modules, their programming is subject to severe limitations.
- *Memory mapped I/O.* Microprocessor and I/O modules share a 8) ___________________ with a common address range. This arrangement has the advantage of allowing the large number of instructions dedicated to addressing microprocessor memory to be used for the I/O modules. One drawback of the occupation of address space does exist, especially affecting microprocessors utilizing a word length of 4 or 8 bits. By contrast, state-of-the-art microprocessors handling 9) _____________________ operate only in conjunction with memory mapped I/O architectures.

Appendix 2

Applications of mobile technologies – Location Based Services

Location Based Services (LBS) have been considered the most potential part of wireless value-added services in the context of mobile technologies applications. LBS are the kind of services accessible with mobile devices through the mobile network. They utilize the ability to use the location of the mobile device. LBS utilize wireless connections like GSM or Wi-Fi and geographic information (GIS) to provide a mobile user with information that depends on a particular geographical location of the mobile terminal.

From a historical point of view location based information is not a new idea. For several years the position of specific information was delivered to mass-audience via posters (e.g. of concerts, meetings, events) or by traffic signs, which submit navigational information.

However, the innovative character of LBS relies on providing two-way communication and interaction. The user can define what kind of information he needs, his preferences and his position. This helps the provider to deliver information filtered by the appropriate criteria specified by the user.

In this context several elements can be distinguished in LBS systems architecture, namely:

- **Mobile Devices:** These include PDAs, Mobile Phones, Laptops, navigation units of cars or a toll boxes for road pricing in trucks.
- **Communication Network** that enables transferring information between the user and the service provider. This is usually accomplished by using GSM or Wi-Fi wireless network.
- **Positioning Component** usually obtained by the built-in Global Positioning System (GPS) receiver. Other methods of positioning include the utilization of WLAN (wireless local area networks) stations, active badges or radio beacons.
- **Service and Application Provider,** which consists of the set of programmed functions delivered by developing company service. Here examples can be a route finding service, yellow pages browsing, with respect to position or searching for specific information on objects of user interest (e.g. assortment of the nearby shops).
- **Data and Content Provider**: Service providers usually do not store and maintain all the information that can be requested by users. Therefore the dedicated database from the maintaining authority (e.g. mapping agencies) or business and industry partners (e.g. yellow pages, traffic companies) is often used as a part of LBS infrastructure.

Appendix 3A

Steganography and digital watermarking

A recent rapid growth of the Inernet, together with availability of inexpensive digital recording and storage devices has created (1) ……………….. in which it became very easy to obtain, replicate and distribute digital content without any loss in quality. This has become a great concern to the multimedia content (music, video, and image) publishing industries, because technologies or techniques that could be used to protect (2) .... for digital media, and prevent unauthorized copying are in a developing state.

(3) ……………….. has its limitations in protecting intellectual property rights because once a message content is decrypted, there is nothing to prevent an authorized user from illegally replicating digital content. (4) …............................... was obviously needed to help establish and prove ownership rights, track content usage, ensure authorized access, to facilitate content authentication and to prevent illegal digital content replication.

This need attracted attention in the early (5) .... from the research community and industry leading to a creation of a new information hiding form, called digital watermarking. The basic idea of digital watermarking is to create a metadata containing information about the digital content to be protected, and hide it within that content. The information to hide, the metadata, can have very different formats e.g. it may be formatted as a character string or a binary image pattern. The embedded watermark should be (6) ...., and it should be robust enough to survive not only most common signal distortions, but also distortions caused by malicious attacks.

The digital watermarking is a special kind of steganography. Steganography is a kind of secret writing i.e. steganography aims to hide (7) …........................by embedding them into ordinarily looking cover objects, while steganalysis aims at detecting stego objects containing hidden messages. (8) ……………….. contain a lot of redundancies such as natural noises and quantized errors whose changes make no significant impact on their perceptual and statistical properties. These have led people to research digital media for steganography, and in particular, (9) .... have been researched for steganography.

The Least Significant Bit (LSB) embedding is a well-known steganographic method that is the way of replacing secret (usually encrypted) message bits with the least significant bits of sample values in digital media. This method is simple but very easy to attack by setting the LSB's of the message to random binary values so more advanced methods are used.

Appendix 3B

Steganography and digital watermarking

A recent rapid growth of the Internet, together with availability of inexpensive digital recording and storage devices has created an environment in which it became very easy to obtain, replicate and distribute digital content without any loss in quality. This has become a great concern to the multimedia content (music, video, and image) publishing industries, because technologies or techniques that could be used to protect (1) ……………….. for digital media, and prevent (2) .... are in a developing state.
Encryption has its limitations in protecting intellectual property rights because once a message content is decrypted, there is nothing to prevent an authorized user from illegally replicating digital content. Some other technology was obviously needed to help establish and prove ownership rights, track content usage, ensure authorized access, to facilitate content authentication and to prevent illegal digital content replication.

This need attracted attention in the early 1990s from the research community and industry leading to a creation of (3) ……………….., called digital watermarking. The basic idea of digital watermarking is to create a metadata containing information about the (4) …………………, and hide it within that content. The information to hide, the metadata, can have very different formats e.g. it may be formatted as (5) ……………….. . The embedded watermark should be imperceptible, and it should be robust enough to survive not only most common signal distortions, but also distortions caused by (6) ....

The digital watermarking is a special kind of steganography. Steganography is (7) ……………….., i.e. steganography aims to hide the existence of secret messages by embedding them into ordinarily looking cover objects, while steganalysis aims at detecting stego objects containing (8) .... Digital media contain a lot of redundancies such as natural noises and quantized errors whose changes make no significant impact on their perceptual and statistical properties. These have led people to research digital media for steganography, and in particular, digital images have been researched for steganography.

The Least Significant Bit (LSB) embedding is a well-known steganographic method that is the way of replacing secret (usually encrypted) message bits with the least significant bits of sample values in digital media. This method is simple but very easy to attack by setting the LSB's of the message to random binary values so (9) .... are used.

Internet of the Future

Ex. 1

As a class divide into two groups A and B.
As – sit together, look at text A. Write correct questions to get information for the gaps in your text.
Bs – do the same with your text.
Do not look at each other's versions. Your teacher will check your questions.
When you have finished, make pairs A and B. In turns ask each other your questions and fill in the gaps. Finally look at both texts to check your answers.

Version A

Architecture of recent Internet was developed more than 1) ……….. ago. Its first generation dated 1969 was designed by researchers for research. During the first 20 years, works on competing networking technologies, e.g. IBMs SNA, Xerox's XNS, or Digitals DECnet could be observed. This first phase that lasted till 2) ……… may be referred to as Internet 1.0.

In 1989, industry started to adapt the Internet for 3) ……….. use. A number of issues (including security, traffic management, congestion control, scalability, and others), became more important than before. However, the most urgent problem was the shortage of Ipv4 addresses. Some 4) ……….. (including private addresses, network address translation – NAT) were introduced, but the 5) ……… was not as visible as desired. Another important issue that remained unsolved referred to a proper provisioning the requested level of Quality of Service (QoS) including bandwidth, transmission delay, and jitter. This second period, called Internet 2.0 phase, lasted for the next 6) ……….. years.

Internet has become an 7) …………of our life and business. After the next 20 years, we are now entering a new phase, called Internet of the Future, or Internet 3.0. There are about 40 ongoing projects all over the world aimed at designing the new architecture. The general idea is to 8) …………. while keeping the best ideas from the past.

Researchers are sure that important aspects that should be taken into account are as follows. First, Internet 3.0 should be 9) ………., i.e. enabling the users (e.g. companies, government, and citizens) to set the respective rules to protect the information. Another important issue is mobility. End-users are becoming more and more mobile, which now causes problems related with addressing/naming.

Another problem is energy-efficient 10) ……….. . Some solutions, including utilization of the sleeping mode of wireless nodes, are known. However, it should be also applied to wired network devices. In the current Internet architecture, the IP address identifying the host changes every time the node 11) ………….. to a different location. This in turn causes the problems related to systems reachability. To solve it, the identity of a system should be separated from its IP address, and should not change while moving to another location.

Most of the current traffic is of a client-server type. The disadvantage is the common utilization of central servers redirecting the user to servers offering the same content but located closer to the 12) …………. . As a result, it takes too much time as well as causes scalability and performance problems. Faster location of the nearest service is necessary.

Other performance and scalability problems of the current IP-based Internet are the result of the unified control, management, and data plane. It means that data and 13) ………….. are recently sent using the common infrastructure (e.g. IP links). This often brings out serious 14) ………… . Generalized Multiprotocol Label Switching Architecture (GMPLS) is an example of a successful attempt to separate the considered planes. Researchers are sure that such separation is a must.

End-users require some guarantees on transmission delay maximum values and throughput. What is more, according to differentiated characteristics of the used 15) ………., these requirements are differentiated as well. Therefore, next generation Internet should offer differentiated levels of service. In particular, this may be achieved by providing a complete 16) ………… from other services e.g. by means of physical separation of utilized resources.

Last but not least aspect refers to the person-to-person communications. Even though Internet was originally designed to enable computer-to-computer communications, the real aim is often to reach a person, but not a computer. However, this implies assigning the addresses to humans…

Version B

Architecture of recent Internet was developed more than 40 years ago. Its first generation dated 1) ………. was designed by researchers for research. During the first 2) ……….. years, works on competing networking technologies, e.g. IBMs SNA, Xerox's XNS, or Digitals DECnet could be observed. This first phase that lasted till 1989 may be referred to as Internet 1.0.

In 1989, industry started to adapt the Internet for commercial use. A number of issues (including security, traffic management, congestion control, scalability, and others), became more 3) ……….. than before. However, the most urgent problem was the 4) ………. of Ipv4 addresses. Some solutions (including private addresses, network address translation – NAT) were introduced, but the improvement was not as visible as desired. Another important 5) ………… that remained unsolved referred to a proper provisioning the requested level of Quality of Service (QoS) including bandwidth, transmission delay, and jitter. This 6) ………..period, called Internet 2.0 phase, lasted for the next 20 years.

Internet has become an indispensable part of our life and business. After the next 20 years, we are now entering a new phase, called 7) …………, or Internet 3.0. There are about 40 ongoing projects all over the world aimed at 8) ………… The general idea is to start with new solutions while keeping the best ideas from the past.

Researchers are sure that important aspects that should be taken into account are as follows. First, Internet 3.0 should be secure, i.e. enabling the users (e.g. companies, government, and citizens) to set the respective rules to protect the information. Another important issue is 9) …………. . End-users are becoming more and more mobile, which now causes problems related with addressing/naming.

Another problem is energy-efficient communications. Some solutions, including 10) ……….. of the sleeping mode of wireless nodes, are known. However, it should be also applied to wired network devices. In the current Internet architecture, the IP address identifying the host changes every time the node moves to a different location. This in turn 11) ………… the problems related to systems reachability. To solve it, the identity of a system should be separated from its IP address, and should not change while moving to another location.

Most of the current traffic is of a client-server type. The disadvantage is the common utilization of central servers redirecting the user to servers offering the same content but located closer to the requesting node. As a result, it takes too much time as well as causes 12) …………. and performance problems. Faster location of the nearest service is necessary.

Other performance and scalability problems of the current IP-based Internet are the result of the 13) …………, management, and data plane. It means that data and control packets are recently sent using the common infrastructure (e.g. IP links). This often brings out serious security problems. Generalized Multiprotocol Label Switching Architecture (GMPLS) is an example of a 14) ………….. to separate the considered planes. Researchers are sure that such separation is a must.

End-users require some 15) ………….. on transmission delay maximum values and throughput. What is more, according to differentiated characteristics of the used applications, these requirements are differentiated as well. Therefore, next generation Internet should offer differentiated levels of service. In particular, this may be achieved by providing a complete isolation from other services e.g. by means of physical separation of utilized resources.

Last but not least aspect refers to the 16) ……….. communications. Even though Internet was originally designed to enable computer-to-computer communications, the real aim is often to reach a person, but not a computer. However, this implies assigning the addresses to humans…

Ex. 2

Discuss the following with your partner without looking at the text.

1. Enumerate the phases of the Internet.
2. Characterize in brief the first phase of the Internet.
3. Discuss what Internet 2.0 was designed for.
4. Enumerate the most important issues to be solved when designing Internet 3.0.
5. What do you think about the future of person-to-person communications over the Internet?

Biographies

Ex. 1

BIOGRAPHY QUIZ

Work in pairs. Scan all the biographies and find answers to the following questions. Check your answers in the key at the back of the book.
When you have finished quiz another pair.

1. Who is considered the "Father of the computer"?
2. Who invented the first widely used high-level programming language FORTRAN?
3. Who was the only legitimate child of the poet Lord Byron?
4. Who was the lead programmer of computer games "Doom" and "Quake"?
5. Who is called the 'Father of super computing'?
6. Who is credited with inventing the World Wide Web?
7. Which of the scientists is of Dutch origin?
8. Who is the co-author of "Perceptrons" – the foundational work in the analysis of artificial neural networks?
9. Who is the father of the analysis of algorithms?
10. Who popularized the asymptotic notation?
11. Which of the scientists was of Hungarian origin?
12. Who is the pioneer of the application of operator theory to quantum mechanics?
13. Who is the creator and developer of the C++ programming language?
14. Which of the scientists used to be a famous hacker?
15. Who initiated the development of the Linux kernel?
16. Who is the creator of the C programming language and the key developer of the Unix operating system?
17. Which of the scientists was sponsored by the Nazi German Government?
18. Who is the designer of the Pascal programming language?
19. Which of the scientists worked at the British code breaking centre during the Second World War?
20. Which of the scientists wrote a paper on the chemical basis of morphogenesis?

Ex. 2

In groups of four prepare 10 more questions to the texts. Quiz another group.

Ex. 3

Fill in the table with the missing information about famous scientists:

NAME	NATIONALITY	CONTRIBUTION
David Huffman		
Charles Babbage		
John Backus		
Ada Lovelace		
John Carmack		
Seymour Cray		
Tim Berners-Lee		
Edsger Dijkstra		
Marvin Minsky		
Donald Knuth		
John von Neumann		
Bjarne Stroustrup		
Linus Torvalds		
Dennis Ritchie		
Konrad Zuse		
Niklaus Wirth		
Alan Turing		

Ex. 4

Work in groups of three. Each of you chooses one of the scientists. Describe the scientist you have chosen to your partners WITHOUT MENTIONING THE NAME. Start with the most general information. Your partners will guess the name. Change roles.

Ex. 5

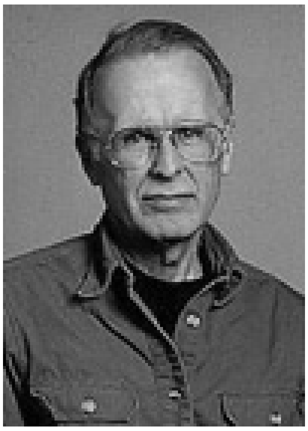
Work in pairs. Design a profile of a perfect scientist using your imagination and creativity. Work with another pair and compare your ideas.

Charles Babbage

Charles Babbage, FRS (26 December 1791 – 18 October 1871) was an English mathematician, philosopher, inventor, and mechanical engineer who originated the concept of a programmable computer. Considered a „father of the computer”, Babbage is credited with inventing the first mechanical computer that eventually led to more complex designs.

Parts of his uncompleted mechanisms are on display in the London Science Museum. In 1991, a perfectly functioning difference engine was constructed from Babbage's original plans. Built to tolerances achievable in the 19th century, the success of the finished engine indicated that Babbage's machine would have worked. Nine years later, the Science Museum completed the printer Babbage had designed for the difference engine, an astonishingly complex device for the 19th century.

John Backus



John Warner Backus (3 December 1924 – 17 March 2007) was an American computer scientist. He directed the team that invented the first widely used high-level programming language (FORTRAN) and was the inventor of the Backus-Naur form (BNF), the almost universally used notation to define formal language syntax. He also did research in function-level programming and helped to popularize it.

The IEEE awarded Backus the W.W. McDowell Award in 1967 for the development of FORTRAN. He received the National Medal of Science in 1975, and the 1977 ACM Turing Award "for profound, influential, and lasting contributions to the design of practical high-level programming systems, notably through his work on FORTRAN, and for publication of formal procedures for the specification of programming languages." In 1953 Backus developed the language Speedcoding, the first high-level language created for an IBM computer. Programming was very difficult, and in 1954 Backus assembled a team to define and develop Fortran for the IBM 704 computer. Fortran was the first high-level programming language to be put to broad use.

Backus made another, critical contribution to early computer science: during the latter part of the 1950s Backus served on the international committees which developed ALGOL 58 and the very influential ALGOL 60, which quickly became the *de facto* worldwide standard for publishing algorithms. Backus developed the Backus-Naur Form (BNF), in the UNESCO report on ALGOL 58. It was a formal notation able to describe any context-free programming language, and was important in the development of compilers. This contribution helped Backus win the Turing Award.

Sir Tim Berners-Lee

Sir Timothy John "Tim" Berners-Lee (born 8 June 1955), also known as "**TimBL**", is a British physicist, computer scientist and MIT professor, credited for his invention of the World Wide Web, making the first proposal for it in March 1989. On 25 December 1990, with the help of Robert Cailliau and a young student at CERN, he implemented the first successful communication between a Hypertext Transfer Protocol (HTTP) client and server via the Internet.
Berners-Lee is the director of the World Wide Web Consortium (W3C), which oversaw the Web's continued development. He is also the founder of the World Wide Web Foundation, and is a senior researcher and holder of the 3Com Founders Chair at the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). He is a director of The Web Science Research Initiative (WSRI), and a member of the advisory board of the MIT Centre for Collective Intelligence In 2004, Berners-Lee was knighted by Queen Elizabeth II for his pioneering work. In April 2009, he was elected as a member of the United States National Academy of Sciences, based in Washington, D.C.
In 1989, CERN was the largest Internet node in Europe, and Berners-Lee saw an opportunity to join hypertext with the Internet: „I just had to take the hypertext idea and connect it to the Transmission Control Protocol and domain name system ideas and – ta-da! – the World Wide Web." He wrote his initial proposal in March 1989, and in 1990, with the help of Robert Cailliau, produced a revision which was accepted by his manager, Mike Sendall. He used similar ideas to those underlying the ENQUIRE system to create the World Wide Web, for which he designed and built the first Web browser, which also functioned as an editor (WorldWideWeb, running on the NeXTSTEP operating system), and the first Web server, CERN HTTPd (short for Hypertext Transfer Protocol daemon). The first web site built was at CERN, and was first put on line on 6 August 1991.

John Carmack

John D. Carmack II (born 20 August 1970) is an American game programmer and the co-founder of id Software. Carmack was the lead programmer of the id computer games *Wolfenstein 3D*, *Doom*, *Quake*, their sequels and the Commander Keen series of games. Though Carmack is best known for his innovations in 3D graphics, he is also a rocketry enthusiast and the founder and lead engineer of Armadillo Aerospace.
Softdisk, a computer company in Shreveport, Louisiana, hired Carmack to work on Softdisk G-S (an Apple IIGS publication), uniting him with John Romero and other future key members of id Software such as Adrian Carmack (not related). Later, this team would be placed by Softdisk in charge of a new, but short-lived, bi-monthly game

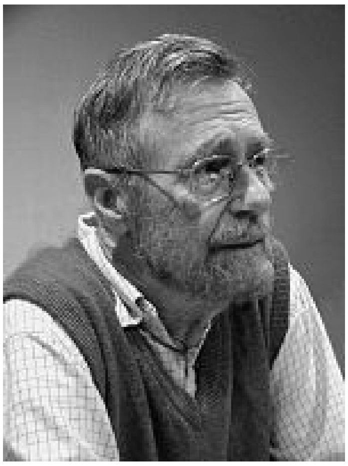
SPIS TREŚCI

subscription product called *Gamer's Edge* for the IBM PC (MS-DOS) platform. In 1990, while still at Softdisk, Carmack, Romero, and others created the first of the Commander Keen games, a series which was published by Apogee Software, under the shareware distribution model, from 1991 onwards. Afterwards, Carmack left Softdisk to co-found id Software, where he remains.
He has pioneered or popularised the use of many techniques in computer graphics, including „adaptive tile refresh" for Commander Keen, raycasting for *Hovertank 3-D*, *Catacomb 3-D*, and *Wolfenstein 3-D*, binary space partitioning which *Doom* became the first game to use, surface caching which he invented for *Quake*, Carmack's Reverse (formally known as z-fail stencil shadows) which he devised for *Doom 3*, and MegaTexture, used in Enemy Territory: Quake Wars. While he was not the first to discover Carmack's Reverse, he developed it independently without knowing of the prior research done on the subject. Carmack's engines have also been licensed for use in other influential first-person shooters such as *Half-Life*, *Call of Duty* and *Medal of Honor*.

Seymour Roger Cray

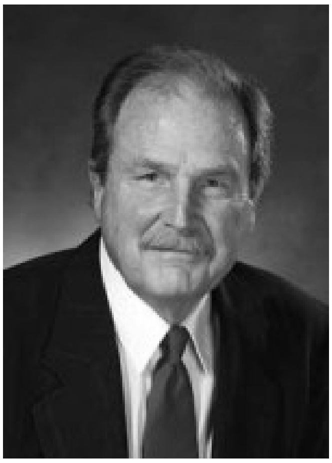
Seymour Roger Cray (28 September 1925 – 5 October 1996) was a U.S. electrical engineer and supercomputer architect who designed a series of computers that were the fastest in the world for decades, and founded Cray Research which would build many of these machines. Called „the father of supercomputing," Cray has been credited with creating the supercomputer industry. Cray was born in 1925 in Chippewa Falls, Wisconsin to Seymour R. and Lillian Cray. He graduated from Chippewa Falls High School in 1943 before being drafted for World War II as a radio operator. He saw action in Europe, and then moved to the Pacific theatre where he worked on breaking Japanese codes. On his return to the United States he received a B.Sc. in Electrical Engineering at the University of Minnesota, graduating in 1949. He also was awarded a M.Sc. in applied mathematics in 1951.
In 1950, Cray joined Engineering Research Associates (ERA) in Saint Paul, Minnesota. ERA had formed out of a former United States Navy lab that had built codebreaking machines, a tradition ERA carried on when such work was available. ERA was introduced to computer technology during one such effort, but in other times had worked on a wide variety of basic engineering as well.
The 6600 was the first commercial supercomputer, outperforming everything then available by a wide margin. While expensive, for those that needed the absolutely fastest computer available there was nothing else on the market that could compete. When other companies (namely IBM) attempted to create machines with similar performance, he increased the challenge by releasing the 5-fold faster CDC 7600.

Edsger Wybe Dijkstra



Edsger Wybe Dijkstra (11 May 1930 – 6 August 2002) was a Dutch computer scientist. He received the 1972 Turing Award for fundamental contributions to developing programming languages, and was the Schlumberger Centennial Chair of Computer Sciences at The University of Texas at Austin from 1984 until 2000.
Shortly before his death in 2002, he received the ACM PODC Influential Paper Award in distributed computing for his work on self-stabilization of program computation. This annual award was renamed the Dijkstra Prize the following year, in his honour.
Among his contributions to computer science are the *shortest path-algorithm*, also known as *Dijkstra*'*s algorithm*; *Reverse Polish Notation* and related *Shunting yard algorithm*; the THE multiprogramming system, an important early example of structuring a system as a set of layers; *Banker's algorithm*; and the semaphore construct for coordinating multiple processors and programs. Another concept due to Dijkstra in the field of distributed computing is that of self-stabilization – an alternative way to ensure the reliability of the system. Dijkstra's algorithm is used in SPF, Shortest Path First, which is used in the routing protocols OSPF and IS-IS.
He died in Nuenen on August 6, 2002 after a long struggle with cancer. The following year, the ACM (Association for Computing Machinery) PODC Influential Paper Award in distributed computing was renamed the Dijkstra Prize in his honour.

David Huffman



David Albert Huffman (9 August 1925 – 7 October 1999) was a pioneer in the computer science field.
Throughout his life, Huffman made significant contributions to the study of finite state machines, switching circuits, synthesis procedures, and signal designs. However, David Huffman is best known for the invention of Huffman code, a highly important compression scheme for lossless variable length encoding. It was the result of a term paper he wrote while a graduate student at the Massachusetts Institute of Technology (MIT), where he earned a D.Sc. degree on a thesis named *The Synthesis of Sequential Switching Circuits*, advised by Samuel H. Caldwell (1953).
„Huffman codes" are used in nearly every application that involves the compression and transmission of digital data, such as fax machines, modems, computer networks, and high-definition television (HDTV), to name a few. Huffman made important contributions in many other areas, including information theory and coding, signal designs for radar and communications applications, and design procedures for asynchronous logical circuits. As an outgrowth of his work on

SPIS TREŚCI

the mathematical properties of "zero curvature Gaussian" surfaces, Huffman developed his own techniques for folding paper into unusual sculptured shapes, and proved an important special case of Kawasaki's theorem on flat-foldability which he called the "critical π condition"; his work was an important precursor to the field of computational origami.

Donald Ervin Knuth

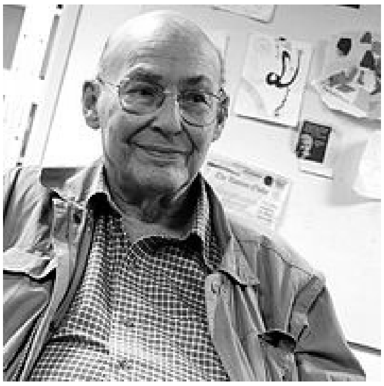
Donald Ervin Knuth (born in the USA, 10 January 1938) is a computer scientist and Professor Emeritus at Stanford University.
He is the author of the seminal multi-volume work *The Art of Computer Programming*. Knuth has been called the „father" of the analysis of algorithms. He contributed to the development of the rigorous analysis of the computational complexity of algorithms and systematized formal mathematical techniques for it. In the process he also popularized the asymptotic notation.
In addition to fundamental contributions in several branches of theoretical computer science, Knuth is the creator of the TeX computer typesetting system, the related METAFONT font definition language and rendering system, and the Computer Modern family of typefaces.
As a writer and scholar, Knuth created the WEB/CWEB computer programming systems designed to encourage and facilitate literate programming, and designed the MMIX instruction set architecture.

Ada Lovelace

Augusta Ada King, Countess of Lovelace (10 December 1815 – 27 November 1852), born **Augusta Ada Byron**, was an English writer chiefly known for her work on Charles Babbage's early mechanical general-purpose computer, the analytical engine. Her notes on the engine include what is recognised as the first algorithm intended to be processed by a machine; as such she is sometimes portrayed in popular culture as the „World's First Computer Programmer".
She was the only legitimate child of the poet Lord Byron (with Anne Isabella Milbanke), but had no relationship with her father, who died when she was nine. As a young adult she took an interest in mathematics, and in particular Babbage's work on the analytical engine. Between 1842 and 1843 she translated an article by Italian mathematician Luigi Menabrea on the engine, which she supplemented with a set of notes of her own. These notes contain what is considered the first computer program—that is, an algorithm encoded for processing by a machine. Though Babbage's engine was not built until 1989–91, Lovelace's notes are important in

the early history of computers. She also foresaw the capability of computers to go beyond mere calculating number-crunching, while others, including Babbage himself, focused only on these capabilities.

Marvin Minsky



Marvin Lee Minsky (born 9 August 1927) is an American cognitive scientist in the field of artificial intelligence (AI), co-founder of Massachusetts Institute of Technology's AI laboratory, and author of several texts on AI and philosophy.

Minsky's inventions include the first head-mounted graphical display (1963) and the confocal microscope (1957, a predecessor to today's widely used confocal laser scanning microscope). He developed, with Seymour Papert, the first Logo "turtle". Minsky also built, in 1951, the first randomly wired neural network learning machine, SNARC.

Minsky wrote the book *Perceptrons* (with Seymour Papert), which became the foundational work in the analysis of artificial neural networks. This book is the centre of a controversy in the history of AI, as some claim it to have had great importance in driving research away from neural networks in the 1970s, and contributing to the so-called AI winter. That said, few of the mathematical proofs present in the book, which are still important and interesting to the study of perceptron network", were ever countered. He also founded several other famous AI models. His book "A framework for representing knowledge" created a new paradigm in programming. While his "Perceptrons" now is more historical than practical book, the theory of frames is in wide use. Minsky was an adviser on the movie *2001: A Space Odyssey* and is referred to in the movie and book.

John Von Neumann

John von Neumann in the 1940s

John von Neumann (28 December 1903 – 8 February 1957) was a Hungarian American mathematician who made major contributions to a vast range of fields, including set theory, functional analysis, quantum mechanics, ergodic theory, continuous geometry, economics and game theory, computer science, numerical analysis, hydrodynamics, and statistics, as well as many other mathematical fields. He is generally regarded as one of the greatest mathematicians in modern history. The mathematician Jean Ieudonné called von Neumann "the last of the great mathematicians", while Peter Lax described him as possessing the most "fearsome technical prowess" and "scintillating intellect" of the century. Von Neumann was a pioneer of the application of operator theory to quantum mechanics, in the development of functional analysis, a principal member of the Manhattan Project and the Institute for Advanced Study in Princeton (as one of the few originally appointed), and a key figure in the development of game theory and the concepts of cellular automata, the universal constructor, and the digital computer.

Dennis MacAlistair Ritchie

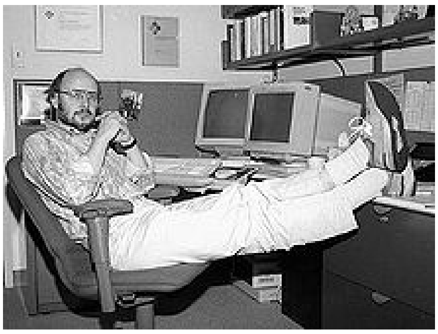
Dennis MacAlistair Ritchie (born 9 September 1941) is an American computer scientist notable for developing C and for having influence on other programming languages, as well as operating systems such as Multics and Unix. He received the Turing Award in 1983 and the National Medal of Technology 1998 on April 21, 1999. Ritchie was the head of Lucent Technologies System Software Research Department when he retired in 2007.

Ritchie is best known as the creator of the C programming language and a key developer of the Unix operating system, and as co-author of the definitive book on'C, The C Programming Language, commonly referred to as *K&R* (in reference to the authors Kernighan and Ritchie).

Ritchie's invention of C and his role in the development of Unix alongside Ken Thompson have placed him as an important pioneer of modern computing. The C language is still widely used today in application and operating system development, and its influence is seen in most modern programming languages. Unix has also been influential, establishing concepts and principles that are now well-established precepts of computing.

Ritchie was elected to the National Academy of Engineering in 1988 for "development of the "C" programming language and for co-development of the UNIX operating system."

Bjarne Stroustrup



Bjarne Stroustrup (born 30 December 1950 in Århus, Denmark) is a computer scientist, most notable for the creation and the development of the widely used C++ programming language. He is currently Professor and holder of the College of Engineering chair in Computer Science at Texas A&M University.

Stroustrup began developing C++ in 1979 (then called "C with Classes"), and, in his own words, "invented C++, wrote its early definitions, and produced its first implementation... chose and formulated the design criteria for C++, designed "all its major facilities, and was responsible for the processing of extension proposals in the C++ standards committee." Stroustrup also wrote what many consider to be the standard text for the language, *The C++ Programming Language*, which is now in its third edition. The text has been revised twice to reflect the evolution of the language and the work of the C++ standards committee.

SPIS TREŚCI

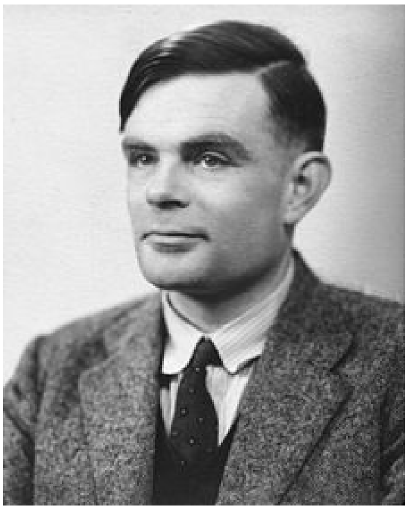
Linus Torvalds

Linus Benedict Torvalds (born 28 December 1969 in Helsinki, Finland) is a Finnish software engineer and hacker, best known for having initiated the development of the Linux kernel. He later became the chief architect of the Linux kernel, and now acts as the project's coordinator. He also created the revision control system Git.

His interest in computers began with a Commodore VIC-20 After the VIC-20 he purchased a Sinclair QL, which he modified extensively, especially its operating system. He programmed an assembly language and a text editor for the QL, as well as a few games. He is known to have written a *Pac-Man* clone named *Cool Man*. On January 5, 1991 he purchased an Intel 80386-based IBM PC and spent a month playing the game *Prince of Persia* before receiving his MINIX copy, which in turn enabled him to begin work on Linux.

His personal mascot is a penguin nicknamed Tux, which has been widely adopted by the Linux community as the mascot of the Linux kernel.

Alan Turing



Alan Mathison Turing (23 June 1912 – 7 June 1954), was a British mathematician, logician, cryptanalyst and computer scientist. He was highly influential in the development of computer science, providing a formalization of the concepts of "algorithm" and "computation" with the Turing machine, which played a significant role in the creation of the modern computer.

During the Second World War, Turing worked for the Government Code and Cypher School at Bletchley Park, Britain's codebreaking centre. For a time he was head of Hut 8, the section responsible for German naval cryptanalysis. He devised a number of techniques for breaking German ciphers, including the method of the bombe, an electromechanical machine that could find settings for the Enigma machine. After the war he worked at the National Physical Laboratory, where he created one of the first designs for a stored-program computer, the ACE.

Towards the end of his life Turing became interested in mathematical biology. He wrote a paper on the chemical basis of morphogenesis, and he predicted oscillating chemical reactions such as the Belousov-Zhabotinsky reaction, which were first observed in the 1960s.

Niklaus E. Wirth

Niklaus Emil Wirth (born 15 February 1934) is a Swiss computer scientist, best known for designing several programming languages, including Pascal, and for pioneering several classic topics in software engineering. In 1984 he won the Turing Award for developing a sequence of innovative computer languages.

From 1963 to 1967 he served as assistant professor of Computer Science at Stanford University and again at the University of Zurich. Then in 1968 he became Professor of Informatics at ETH Zürich, taking two one-year sabbaticals at Xerox PARC in California (1976–1977 and 1984–1985). Wirth retired in 1999.

Wirth was the chief designer of the programming languages Euler, Algol W, Pascal, Modula, Modula-2, Oberon, Oberon-2, and Oberon-07. He was also a major part of the design and implementation team for the Lilith and Oberon operating systems, and for the Lola digital hardware design and simulation system. He received the *ACM Turing Award* for the development of these languages and in 1994 he was inducted as a Fellow of the ACM. He designed the simple programming language PL/0 to illustrate compiler design. It has formed the basis for many university compiler design classes.

His article *Program Development by Stepwise Refinement*, about the teaching of programming, is considered to be a classic text in software engineering. In 1975 he wrote the book "Algorithms + Data Structures = Programs", which gained wide recognition and is still useful today.

Konrad Zuse

Konrad Zuse in 1992

Konrad Zuse (22 June 1910 Berlin – 18 December 1995 Hünfeld near Fulda) was a German engineer and computer pioneer. His greatest achievement was the world's first functional program-controlled Turing-complete computer, the Z3, which became operational in May 1941. He received the Werner-von-Siemens-Ring in 1964 for the Z3. Much of his early work was financed by his family and commerce, but after 1939 he was given resources by the Nazi German Government.

Zuse's S2 computing machine is considered to be the first process-controlled computer. In 1946 he designed the first high-level programming language, Plankalkül. Zuse founded one of the earliest computer businesses on 1 April 1941 (*Zuse Ingenieurbüro und Apparatebau*). This company built the **Z4**, which became the world's first *commercial* computer.

Due to World War II Zuse's work went largely unnoticed in the UK and the US. Possibly his first documented influence on a US company was IBM's option on his patents in 1946. In the late 1960s, Zuse suggested the concept of a Calculating Space (a computation-based universe).

Working in his parents' apartment in 1936, his first attempt, called the Z1, was a floating point binary mechanical calculator with limited programmability, reading instructions from a perforated 35 mm film. In 1937 Zuse submitted two patents that anticipated a von Neumann architecture. He finished the Z1 in 1938. The Z1 contained some 30,000 metal parts and never worked well, due to insufficient mechanical precision. The Z1 and its original blueprints were destroyed during WWII.

Bill Gates and Steve Jobs – Entrepreneurs of the IT Business

Ex.1

Work in pairs. Decide who the sentences below refer to – Bill Gates or Steve Jobs?

1. He launched one of the largest industries of the past decades while still in his early twenties.
2. He enrolled in the Hewlett-Packard Explorer Club.
3. At school he came to know Paul Allen, a classmate with similar interests in technology who would eventually become his business partner.
4. His early experiences with computers included debugging (eliminating errors from) programs.
5. In 1975 he joined a group known as the Homebrew Computer Club.
6. He was responsible for what was probably the first computer virus, a program that copies itself into other programs and ruins data.
7. In 1976 he and Steve Wozniak formed their own company.
8. He dropped out of Harvard in 1975, ending his academic life and beginning his career as a software designer.
9. The foundation he created has donated $4 billion since its start in 1996.
10. In 1986 he purchased a small company called Pixar from filmmaker George Lucas.
11. In November 1997 he announced that his company would sell computers directly to users over the Internet and by telephone.
12. On January 13, 2000, he handed off day-to-day management of his company to a friend and right-hand man Steve Ballmer.
13. In September of 1997 he was named interim CEO of Apple.
14. His occupation was listed as “entrepreneur” in the “high tech” business.

Ex. 2

Divide into two groups A and B. Students A – sit in one group, Students B – sit in another group. Students A – read the text about Steve Jobs, students B – read the text about Bill Gates.

While you read underline interesting facts about the person.

TEXT A

Steve Jobs Biography

Steve Jobs

With his vision of affordable personal computers, Steve Jobs launched one of the largest industries of the past decades while still in his early twenties. He remains one of the most inventive and energetic minds in American technology.

Steven Jobs was born February 24, 1955, in San Francisco, California, and was adopted by Paul and Clara Jobs.

In 1961 the family moved to Mountain View, California. This area, just south of Palo Alto, California, was becoming a centre for electronics. Electronics form the basic elements of devices such as radios, televisions, stereos, and computers. At that time people started to refer to the area as "Silicon Valley." This is because a substance called silicon is used in the manufacturing of electronic parts.

Jobs enrolled in the Hewlett-Packard Explorer Club. There he saw engineers demonstrate new products, and he saw his first computer at the age of twelve. He was very impressed, and knew right away that he wanted to work with computers.

While in high school Jobs attended lectures at the Hewlett-Packard plant. On one occasion he boldly asked William Hewlett (1931–2001), the president, for some parts he needed to complete a class project. Hewlett was so impressed he gave Jobs the parts, and offered him a summer internship at Hewlett-Packard.

In 1975 Jobs joined a group known as the Homebrew Computer Club. One member, a technical whiz named Steve Wozniak (1950–), was trying to build a small computer. Jobs became fascinated with the marketing potential of such a computer. In 1976 he and Wozniak formed their own company. They called it Apple Computer Company, in memory of a happy summer Jobs had spent picking apples.

Jobs had realized there was a huge gap in the computer market. At that time almost all computers were mainframes. They were so large that one could fill a room, and so costly that individuals could not afford to buy them. Advances in electronics, however, meant that computer components were getting smaller and the power of the computer was increasing.

Jobs and Wozniak redesigned their computer, with the idea of selling it to individual users. The Apple II went to market in 1977, with impressive first year sales of $2.7 million. The company's sales grew to $200 million within three years. This was one of the most phenomenal cases of corporate growth in U.S. history. Jobs and Wozniak had opened an entirely new market — personal computers.

By 1980 the personal computer era was well underway. Apple was continually forced to improve its products to remain ahead, as more competitors entered the marketplace. Apple introduced the Apple III, but the new model suffered technical and marketing problems. It was withdrawn from the market, and was later reworked and reintroduced.

Jobs continued to be the marketing force behind Apple. Early in 1983 he unveiled the Lisa. It was designed for people possessing minimal computer experience. It did not sell well, however, because it was more expensive than personal computers sold by competitors. Apple’s biggest competitor was International Business Machines (IBM). By 1983 it was estimated that Apple had lost half of its market share (part of an industry's sales that a specific company has) to IBM.

In 1984 Apple introduced a revolutionary new model, the Macintosh. The on-screen display had small pictures called icons. To use the computer, the user pointed at an icon and clicked a button using a new device called a mouse. This process made the Macintosh very easy to use. The Macintosh did not sell well to businesses, however. It lacked features other personal computers had, such as a corresponding high quality printer. The failure of the Macintosh signalled the beginning of Jobs’s downfall at Apple. Jobs resigned in 1985 from the company he had helped found, though he retained his title as chairman of its board of directors.

Jobs soon hired some of his former employees to begin a new computer company called NeXT. Late in 1988 the NeXT computer was introduced at a large gala event in San Francisco, aimed at the educational market. Initial reactions were generally good. The product was very user-friendly, and had a fast processing speed, excellent graphics displays, and an outstanding sound system. Despite the warm reception, however, the NeXT machine never caught on. It was too costly, had a black-and-white screen, and could not be linked to other computers or run common software.

NeXT was not, however, the end of Steve Jobs. In 1986 Jobs purchased a small company called Pixar from filmmaker George Lucas (1944–). Pixar specialized in computer animation. Nine years later Pixar released *Toy Story*, a huge box office hit. Pixar later went on to make *Toy Story 2* and *A Bug*’*s Life*, which Disney distributed, and *Monsters*, *Inc.* All these films have been extremely successful. *Monsters*, *Inc.* had the largest opening weekend ticket sales of any animated film in history.

In December of 1996 Apple purchased NeXT Software for over $400 million. Jobs returned to Apple as a part-time consultant to the chief executive officer (CEO). The following year, in a surprising event, Apple entered into a partnership with its competitor Microsoft. The two companies, according to the *New York Times*, “agreed to cooperate on several sales and technology fronts.” Over the next six years Apple introduced several new products and marketing strategies.

In November 1997 Jobs announced Apple would sell computers directly to users over the Internet and by telephone. The Apple Store became a runaway success. Within a week it was the third-largest e-commerce site on the Internet. In September of 1997 Jobs was named interim CEO of Apple.

In 1998 Jobs announced the release of the iMac, which featured powerful computing at an affordable price. The iBook was unveiled in July 1999. This is a clam-shaped laptop that is available in bright colours. It includes Apple’s AirPort, a computer version of the cordless phone that would allow the user to surf the Internet wirelessly. In January 2000 Jobs unveiled Apple's new Internet strategy. It included a group of Macintosh-only Internet-based applications. Jobs also announced that he was becoming the permanent CEO of Apple, spearheading the advent of the iPod, iPhone and iPad.

He eventually resigned as CEO in August 2011, while on his third medical leave. He was then elected chairman of Apple's board of directors. His occupation was listed as "entrepreneur" in the "high tech" business.

Steve Jobs was truly a computer industry visionary.

Unfortunately he died on October 5, 2011.

adapted from
http://www.notablebiographies.com/Ho-Jo/Jobs-Steve.html

TEXT B

Bill Gates Biography

Bill Gates

Microsoft cofounder and chief executive officer Bill Gates has become the wealthiest man in America and one of the most influential personalities in the ever-evolving computer industry.

William H. Gates III was born on October 28, 1955, in Seattle, Washington.

Although Gates’s parents had a career in law in mind for their son, he developed an early interest in computer science and began studying computers in the seventh grade at Seattle's Lakeside School. Lakeside was a private school chosen by Gates's parents in the hope that it would be more challenging for their son's intellectual drive and curiosity. At Lakeside, Gates came to know Paul Allen, a classmate with similar interests in technology, who would eventually become his business partner. Immediately, Gates and Allen realized the potential of the young computer industry.

Gates's early experiences with computers included debugging (eliminating errors from) programs for the Computer Centre Corporation's PDP-10, helping to computerize electric power grids for the Bonneville Power Administration, and founding with Allen a firm called Traf-O-Data while still in high school. Their small company earned them twenty thousand dollars in fees for analysing local traffic patterns.

While working with the Computer Center’s PDP-10, Gates was responsible for what was probably the first computer virus, a program that copies itself into other programs and ruins data. Discovering that the machine was connected to a national network of computers called Cybernet, Gates invaded the network and installed a program on the main computer that sent itself to the rest of the network's computers, making it crash (became damaged). When Gates was found out, he was severely punished, and he kept away from computers for his entire junior year at Lakeside.

Gates entered Harvard University in 1973 and pursued his studies for the next year and a half. Gates dropped out of Harvard in 1975, ending his academic life and beginning his career as a software designer. At this time, Gates and Allen cofounded Microsoft. They wrote programs for the early Apple and Commodore machines. One of Gates’s most significant opportunities arrived in 1980, when IBM approached him to help with their personal computer project, code name Project Chess. Gates developed the Microsoft Disk Operating System, or MS-DOS. An operating system is a type of software that controls the way a computer runs. Not only did he sell IBM on the new operating system, but he also convinced the computer giant to allow others to write software for the machine. By the early 1990s Microsoft had sold more than one hundred million copies of MS-DOS, making the operating system the all-time leader in software sales. For his achievements in science and technology, Gates received the Howard Vollum Award in 1984 from Reed College in Portland, Oregon.

Gates's competitive drive and fierce desire to win has made him a powerful force in business, but it has also consumed much of his personal life. In the six years between 1978 and 1984, he took a total of only two weeks' vacation

Many criticize Gates not just for his success, but because they feel he tries to unfairly, and maybe even illegally, dominate the market. As a result of Microsoft's market control, the U.S. Department of Justice brought an antitrust lawsuit (a lawsuit that is the result of a company being accused of using unfair business practices) against the company in 1998, saying the company had an illegal stronghold on the software industry.
Gates maintained Microsoft's success over rivals such as Oracle and IBM was simply the result of smart, strategic decision making. U.S. District Judge Thomas P. Jackson did not agree, and in November 1999, he found Microsoft to be a monopoly (a company with exclusive control) that used its market power to harm competing companies. Because of the ruling, Gates faced the prospect of breaking up Microsoft.
On January 13, 2000, Gates handed over day-to-day management of Microsoft to his friend and right-hand man Steve Ballmer, adding chief executive officer to his existing title of president. Gates held on to his position as chairman in the reshuffle, and added the title of chief software architect.

Aside from being the most famous businessman of the late 1990s, Gates also has distinguished himself as a philanthropist (someone working for charity). He and his wife Melinda established the Bill & Melinda Gates Foundation, which focuses on helping to improve health care and education for children around the world. The foundation has donated $4 billion since its start in 1996.

Although many describe Gates as cold and distant, his friends find him friendlier since his marriage and since the birth of his daughter, Jennifer, in April 1996. Further, he is recognized for his overall contribution to both the world of technology and his efforts in philanthropy. In *Forbes* magazine's 2002 list of the two hundred richest people in the world, Gates was number one for the eighth year running, coming in with a net worth of $52.8 billion.

adapted from
http://www.notablebiographies.com/Fi-Gi/Gates-Bill.html

Ex. 3

Work as a group. Invent 5 untrue facts about your character. Write them down.

Ex. 4

Make pairs - student A and student B. Describe your biography to your partner, including the 5 untrue facts. Your partner will listen and guess which facts are not true.
Change roles.

Ex. 5

When you have finished, read the other biography. Discuss the answers to Exercise 1 with your partner.

Ex. 6

Divide into two groups.
Prepare arguments to discuss the following question:
Who contributed more to IT development?
Group A – support Steve Jobs;
Group B – support Bill Gates.
When you are ready – have a debate.

Tapescripts

UNIT 1

Ex. 9 and 10 ***T2***

Another distinguishing characteristic of I/O modules consists of the supported operating modes. Four different operating modes can be identified:

Programmed I/O. The I/O module is directly controlled by the microprocessor, which handles its entire set of functions by means of a single program. The microprocessor is thus forced to wait while an I/O module is performing an operation. Therefore, this operating mode is used exclusively with microprocessors handling only input/output tasks (e.g. controlling intelligent sensors and actuators).

Polled I/O. The I/O module is capable of performing independent operations, during which the input/output data are committed to intermediate storage in special buffers. The microprocessor periodically checks the status of the I/O module and transfers new data if required. This operating mode is suitable mainly for those microprocessors that feature only a software based interrupt system, a so-called *software interrupt system*.

Interrupt-driven I/O. The I/O module independently processes all input/output operations. Using a so-called *interrupt line*, it informs the microprocessor of the presence of new data or of a required microprocessor operation. As an essential advantage of this operating mode, the microprocessor and I/O modules are able to operate in parallel. The microprocessor program should be interrupted only in situations where the I/O module requires microprocessor assistance.

Direct-memory I/O access (*DMA*). In this operating mode, the I/O modules are capable of a direct data exchange with the memory area without the need for microprocessor participation. This operating mode is supported mainly by microprocessors belonging to the top-end performance category. As is the case with the interrupt-driven I/O, this operating mode requires hardware that prioritises all waiting requests and even blocks these if required. More often than not, the software components covering the described hardware-oriented aspects of the I/O modules of a microcontroller are grouped together in a layer of the platform software.

UNIT 2

Ex. 7 ***T4***

Examples of LBS Applications

Applications of LBS include several areas of our everyday life.

A – Emergency Services

One of the most recent and popular applications of LBS is the ability to locate people who are either unaware of their exact location or are not able to reveal it due to an emergency situation such as injury, criminal attack, car accident and so on. In this context, systems that can transfer the exact location of a mobile terminal to the emergency service enable assistance to be provided quickly and efficiently.

B – Navigation Services

These are navigation services for mobile users who need to find the route to a particular location based on their current geographical position. Positioning a mobile phone informs the user exactly where he is and provides him with detailed directions about how to get to a desired destination. Apart from routing functionalities, the mobile phone user can be notified about interesting places (so called Points of Interest – POI) in the nearby area such as restaurants, banks, gas stations etc. Above all, the user gets the pre–calculated route via the mobile network connection with a detailed description of particular parts of the desired route.

C – Augmented Reality
A new approach for mobile applications, which is an area of interest for researchers and many R&D (Research and Development) companies, is pulling graphics out of the phone display and integrating them into the real-world environment. This new technology, called augmented reality, is supposed to blur the line between the real world and computer generated reality. In contrast to other virtual environments, in augmented reality the user can see the real world around him, with computer graphics superimposed or composed with the real world.

UNIT 3

Ex. 4 ***T6***

Message and Information

The goal of any Information System is to **transmit certain information** from the sender (source) to the recipient (destination) and Information Theory is the science which specifies **conditions that must be fulfilled** by basic blocks of the information/communication system. The landmark event that established the discipline of information theory, and brought it to immediate worldwide attention, was the publication of Claude E. Shannon's classic paper "Mathematical Theory of Communication" published in 1948. In his paper Shannon, for the first time, introduced the qualitative and quantitative model of communication **as a statistical process.**

The most basic concept in Information Theory is information, but what do we mean by information? Everybody would agree that information is transferred **from a source to a recipient** only if the recipient of the information did not already have the information to begin with. Messages that convey information that is certain to happen and already known by the recipient contain **no real information**. This leads to the following definition: information is knowledge obtained by the recipient of the message which reduces his uncertainty and hence allows him to realize or improve his actions.

Having defined information, the engineer has to answer another question: how do we measure the information? A message which conveys no information has an information measure of zero. Infrequently occurring messages contain more information than more frequently occurring messages, with an improbable message conveying an infinite amount of information. On that basis Shannon derived a measure of information contained in message *m*:

$$I(X=m)=\log_2 \frac{1}{P(X=m)}=-\log_2 P(X=m)\ \ [bit]$$

where $P(X = m)$ is the probability that message m is generated by the source X. Note that information measure has nothing to do with **the importance of messages**, as the messenger should not look into the message he carries, analysing if it is important or not.

Ex. 10 and 11 ***T7***

Cryptography

Channel coding provides error protection but many users also need another kind of protection. They want to be certain that their message will be received only by the intended receiver. On the other hand, there is always someone who wants to know the content of the message. Cryptography is the science of writing messages that no one, except for the intended receiver, can read; while cryptanalysis is the science of reading them anyway.

Cryptography is a subject that has been intensively studied and applied since ancient Roman times, and efforts made to come up with better encryption methods, resistant to attempts to break them, continue to this day. Cryptography is the art of encoding and decoding messages so that they can be securely transmitted from a sender to a receiver without the fear that an outside party (the enemy) can intercept and read or alter the message's contents. A very important time for the development and proliferation of cryptography was WWII, when rapid intelligence needed to be sent by secret methods on all sides of the fronts. The Germans had their famous Enigma machine, a cipher machine consisting of rotors, a plug board and a keyboard. The machine worked with any combination of rotors, where a specific rotor revolved slightly with each keystroke, thus changing the current flow through the machine, and therefore the encryption of the letter. The Enigma cryptosystem was first broken by Polish cryptanalysts even before WWII began. Later their work was continued by British cryptanalysts. WWII also brought the one time pad (OTP) algorithm into common use. OTP is an encryption algorithm where the plaintext is combined with a random key that is as long as the plaintext, so each character is used only once. If the key is truly random, never reused and kept secret, the OTP, as proven by Claude Shannon, is unbreakable. Major advances in cryptography arrived in the mid-1970s with the advent of the Data Encryption Standard. Since then we have observed accelerating growth of civil cryptography connected with the rapidly evolving Internet, network banking services and Internet shopping.

UNIT 4

Ex. 3 and 4 ***T9***

There is an increasing need for the development of new human-computer interfaces (HCI). They are especially useful in situations when it is not possible, difficult or ineffective to use traditional input devices, like a keyboard and a mouse. The main goal of each HCI application is to make these tools more intuitive and easier to use and more effective for their desired function. One of the main areas of applications of new human-computer interfaces is making it possible for people with permanent or temporary disabilities to use computers in an efficient way. There are two main types of such solutions. The first group utilizes devices mounted directly on the user's body. Applications in the second group are contactless and they use remote sensors only. As a result, they are much more comfortable for the user. Amongst contactless solutions, vision-based human-computer interfaces are the most promising ones. They utilize cameras and image processing algorithms to detect signs and gestures made by a user and execute configured actions. The most common vision-based applications employ eye and hand tracking.

Ex. 6 ***T10***

Fig. 4.2 presents the diagram of the algorithm used in LipMouse. First, the user's face is detected in every image frame captured by a web camera. Then, the mouth region is localized and its shift from the reference mouth position is calculated. This shift is directly used to move a screen cursor; the greater the shift, the faster the cursor moves towards a given direction. Simultaneously, a small region (blob) placed on the user's lips is found in the mouth region. This blob is used as an initial condition for an iterative method for the approximation of lip shape. Lip gestures are recognized by an artificial neural network (ANN). ANNs are powerful tools for solving various classification tasks. ANNs require training. A training set contains feature vectors describing items that are to be divided into different categories. Each training sample is labelled with a proper class name. Using a training algorithm, ANN is able to determine the weight of each feature in the feature vector and learn how to properly classify feature vectors into categories. ANNs have generalization capabilities, which means that during normal operation they are able to correctly classify samples that are not present in a training set.

UNIT 5

Ex. 4 ***T12***

Geographic information describes places on the Earth. It can refer to the location and characteristics of objects at every scale, such as continents, countries, cities, districts and buildings, down to single trees and lamp posts. The location of such objects on the surface of the planet is expressed in geographic coordinates, which consist of latitude, longitude and height above sea level. In this context, a Geographic Information System (GIS) is a software which allows for creation, manipulation, storage, analysis and visualization of spatial data. The investigation of geographic data in a GIS (e.g. by means of spatial interaction, autocorrelation, regression or interpolation) is referred to as spatial analysis.
Geographic Information Systems have been created for the purpose of administering and monitoring the environment. They operate on geographic data such as outlines and placement of buildings, parks or transport routes, as well as attributes such as addresses, post and zip codes. Because this information is placed in a geographical context, it makes it very easy to find the geographical coordinates of a particular address – this process is called geocoding. This spatial data is kept in geodatabases. A Geodatabase, in a broad sense, is a regular relational database which supports spatial extensions.
For proper storage and handling, digital geographic data requires appropriate file formats

UNIT 6

Ex. 5 T14

The ultimate success of any IT project is the satisfied customer actually using and paying for the product, an information system. Information systems engineering, together with project management, answers the question how to develop the system and run the project to achieve this success. The former focuses on delivering a "good" system, while the latter focuses on delivering it effectively.

A "good" information system must provide adequate functionality to satisfy the needs of the users. The set of functional features of a system is called *the system scope*. Matching the scope of a delivered system with the actual needs of the users is the key challenge in IT business and a prerequisite for successful system deployment.

However, addressing appropriate user needs alone is not enough. The product must exhibit sufficient quality to be actually used (or even to be usable at all). The quality of a software-based system is understood as a set of attributes such as reliability, performance, scalability or usability. A taxonomy of software quality is called *the quality tree*. Based on earlier models of McCall and Boehm, ISO issued the ISO 9126 standard quality model.

UNIT 8

Ex. 3 and 4 T17

The term "Robot" was first used in a play called "Rossum's Universal Robotts" by the Czech writer Karel Capek in 1921. In 1941, science fiction writer Isaac Asimov first used the word "robotics" to describe the technology of robots and predicted the rise of a powerful robot industry. In 1948, an influential artificial intelligence research paper, "Cybernetics", was published by Norbert Wiener.

Today's robots are neither mechanically nor structurally close to handling the same tasks as their fictitious counterparts. But they have graduated from remote-controlled contrivances to true autonomous mobile robots that can navigate in space on their own. The most desirable robots can do more than merely follow human commands. They come in different shapes and sizes. Some of them can even carry on a conversation, recognize their environment, identify people and do a variety of useful tasks.

UNIT 9

Ex. 4 T19

One of the most important commodities in people's lives is information. Its importance is constantly growing. There is a huge difference between the situation twenty years ago and the situation now for people who are looking for a specific piece of information. Twenty years ago, they would have had to go to a bookshop or to a library, and not only to one. Even when the data they were looking for was there, it was not easy to find. There were catalogues printed on small cards, but their quality was disputable. With the advent of the Internet, things changed. People are faced with a deluge of information. However, this does not mean that the task of finding a specific piece of data is very easy. The problem is that there is so much information that rejecting irrelevant items and ranking the relevant ones is more difficult than finding anything that might be related to a *query*. Additional problems may arise from situations where texts are written in a foreign language, or when long documents match a query, so they have to be summarized.

UNIT 10

Ex. 3 and 4 T21

Video Games

Video games are electronic games that involve interaction with a user interface to generate visual feedback on a video device. They usually serve for *entertainment* but can also be used in other fields, such as *education*, *training*, *practice*, *communication*, *social therapy*, *advertising*, *popularization*, and others. Video games are in fact a serious industry creating a huge market of products and services worth billions of dollars each year.

Game design is a process of game creation that can be treated as some kind of craft combining both artistic and functional elements. It is not pure art, though it demands artistic skills to create aesthetic appeal and harmony. It is also not pure engineering, though it requires great programming skills to create an efficient and fluent game. The whole process of game design and development is usually complex and requires the cooperation of many staff members of the development team. However, it is still possible to create great games in small teams or even alone. It is possible because the most important thing about games is a good **concept**.

UNIT 11

Ex. 4 and 5 T23

According to Codd's theory, an essential element of a database is a database relation. The database relation is a special case of a mathematical relation. The restriction imposed by Codd on the mathematical relation is that domains of the components of the database relation must contain only simple (i.e. atomic) values. This limitation is called the *first normal form* (1NF). Let us explain this by a simple example. Look at the table below. It presents a simple database relation in tabular form. This relation, called STUDENTS, represents a collection of some college students. Each student has a name, a surname, a date of birth, and is assigned a unique album number. For each student information about the year of admission to the college is also stored. The rows of this table represent *tuples* of the relation, and the columns represent *attributes*. The attributes have their names shown in the table header. Note that in the table cells there are single (simple) values rather than composite values. This means that they are elements of simple domains. So, the relation STUDENTS satisfies the first normal form, and thus it is a proper database relation.

STUDENTS

Album_No	**Surname**	**Name**	**Date_of_Birth**	**Date_of_Adm**
143879	Kowalski	Jan	23.02.1992	1.10.2010
654889	Nowak	Marek	12.12.1989	1.10.2009
324672	Dworak	Krzysztof	5.06.1990	1.10.2008
321445	Ziółkowski	Stefan	1.06.1989	1.03.2010

UNIT 12

Ex. 6 and 7 *T25*

All combinatorial problems can be categorized with respect to the length of the output string and with respect to the inherent complexity. According to the first categorization we have:

Decision problems. In these problems the output is a simple "yes" or "no" answer for each input. In other words, the algorithm has to decide whether the input satisfies a given property. An example of such a problem is the halting problem mentioned earlier.

Optimization problems. In these problems the task is to find a mathematical object satisfying a given property. An example of such a problem is finding the best move in a particular arrangement of chess pieces on the board.

Glossary

A

acceleration – przyspieszenie
accelerometer – miernik przyspieszenia
access – dostęp
accessible – dostępny
actuator – pozycjoner, rozrusznik, pobudzacz
address bus – szyna adresowa
adhere – przylegać, trzymać się, stosować się do
advance – posuwać się naprzód, wspomagać, poprawiać
advent – nadejście, przybycie, nastanie
agile – zwinny, zręczny, sprawny
algorithm – algorytm
alternative – alternatywa, wybór
approach – zbliżanie się, podejście, metoda, sposób rozwiązania
appropriate – stosowny, odpowiedni, trafny
artificial neural network (ann) – sztuczna sieć neuronowa
assembly – zgromadzenie, montaż
attenuate – osłabiać, łagodzić, tłumić
attribute – przypisywać, właściwość, atrybut
augmented – powiększony, rozszerzony
authentication – legalizacja, uwiarygodnienie, poświadczenie

B

bandwidth – przepustowość, szerokość pasma
bit – bit, cyfra dwójkowa
bitmap – bitmapa, zatomizowany obraz komputerowy

C

cache – skrytka, pamięć podręczna
calibration – skalowanie
cantilever – podpora, wspornik, dźwignia
catalyse – katalizować
chassis – podwozie, płyta montażowa
checkmate – pobić, szachować, udaremniać
circuit – obwód
closed-loop – pętla zamknięta
coaxial – współosiowy
coefficient – współczynnik
coherent – logiczny, spójny, zwarty, zrozumiały
combinatorial – kombinatoryczny
commodity – towar, artykuł
compensate – kompensować
composite – zbór, połączenie, złożony
composite key – klucz złożony
concatenate – połączyć, związać, powiązać
concomitant – współistniejący, towarzyszący
concurrency – zgodność, współbieżność
constituent – element, składnik
constraint – przymus, ograniczenie
contaminant – zanieczyszczenie, czynnik skażający
contrivance – przyrząd, urządzenie, pomysł, sprytny plan
convenient – dogodny, wygodny, poręczny
convex – wypukły
coordinate – współrzędna
cosine – kosinus
countermeasure – przeciwdziałanie, środek zaradczy
counterpart – odpowiednik
cross-reference – odsyłacz
crucial – istotny
cryptography – kryptografia, szyfrowanie
cumulative – kumulacyjny, łączny
cylindrical – cylindryczny, walcowaty

D

data bus – szyna danych
database management – system zarządzania bazami danych
data integrity – intagralność danych
debugged – odpluskwiony, odwirusowany
deluge – potop, zalew
density – gęstość
deployment – rozmieszczenie, rozstawienie
depth – głębokość
digitised – cyfrowy
disability – niemożność, niezdolność, upośledzenie
discrete – dyskretny (nie ciągły)
disoblige – zaniedbać, pominąć
disperse – rozpraszać
distributed – rozproszony
divisor – dzielnik
domain – domena, dziedzina
doubtful – wątpliwy, niepewny

E

efficient – skuteczny, zdolny, sprawny
elicit – wywoływać, uzyskiwać
embedded – osadzony, wyryty, zagnieżdżony, wbudowany
enable – umożliwić
encapsulate – wyrażać, obejmować, zawierać, hermetyzować
encounter – napotkać, natknąć się
encryption – szyfrowanie, utajnianie
entity – jednostka, istota, byt
entrepreneur – przedsiębiorca
entropy – entropia
explicitly – wyraźnie, kategorycznie, otwarcie, jawnie
exponentially – wykładniczo

F

facilitate – ułatwiać, umożliwiać
failure – nipowodzenie, porażka
fair – uczciwy, sprawiedliwy, przychylny
feasible – wykonalny, możliwy
fictitious – fikcyjny, zmyślony
fluctuation – wahania, zmiany
foreign key – klucz obcy
former – ten pierwszy
framework – szablon, zręb, biblioteka, wzorzec
fully-fledged – w pełni rozwinięty
functionality – funkcjonalność, zespół,

G

gantry – brama, portal
gear – mechanism, bieg, przełożenie
genre – gatunek, rodzaj
gesture – gest
grayscale – skala szarości, odcienie szarości
grid – krata, siatka, sieć
gripper – chwytak, zacisk

H

halting – wahający się, niepewny, zatrzymujący się
handicap – upośledzenie, utrudnienie, wada
header – nagłówek
heterogeneous – różnorodny
host – gospodarz, maszyna uruchamiająca program

I

identifier – identyfikator
immerse – zanurzać, pogrążać
impact – wpływ
imperceptible – niedostrzegalny, niezauważalny, nieuchwytny
implement – wykonywać, wdrażać, realizować
implementation – wdrażanie, implementacja
implicitly – bezwarunkowo, domyślnie, bezwzględnie, niejawnie
inbound data – dane wejściowe
inconvenience – niedogodność
incorporate – wcielać, włączać, zawierać
innovative – innowacyjny
insertion – wstawienie, wrzucenie
integer – liczba całkowita
integrity – integralność
interaction – interakcja
intercept – przecinać
interface – interfejs
interoperability – zdolność do współpracy
interpolation – interpolacja
intersect – przeciąć, krzyżować się
intuitive – intuicyjny
involve – włączać, angażować
iterative – iteratywny, wielokrotny

J

jaw – szczęka

L

label – etykieta
latitude – szerokość geograficzna
latter – ten drugi
length – długość
leverage – dźwignia, wywrzeć nacisk
logarithm – logarytm
longitude – długość geograficzna

M

matrix – matryca, macierz
metadata – metadane
microcontroller – mikrokontroler
misconception – błędne wyobrażenie

N

neural – nerwowy
node – węzeł
normalisation – normalizacja
notional – hipotetyczny

O

object-oriented database – obiektowa baza danych
object-relational database – obiektowo-relacyjna baza danych
open-loop – pętla otwarta
orthogonal – ortogonalny
oscillator – oscylator
overlap – nakładać się, pokrywać się

P

pace – tempo
paradigm – paradygmat, model
parity – parzystość
participant – uczestnik
patent-pending – patent zgłoszony/ zgłoszenie patantowe
peer – równouprawniny, rówieśnik
pivotal – kulminacyjny, decydujący, osiowy, obrotowy
pneumatic – pneumatyczny
polynomial – wielomian, wielomianowy
predefined – uprzednio zdefiniowany
prerequisite – warunek wstępny
primality – pierwszość
primary key – klucz główny, klucz podstawowy
procedure – procedura
proliferation – rozprzestrzenianie się
proof – dowód
proportional – proporcjonalny
protocol – protokół
puckered – skrzywiony, zmarszczony

Q

qualitative – jakościowy
quantifiable – wymierny
quantitative – ilościowy
query – zapytanie
quotient – współczynnik

R

raster – raster
raster graphics – grafika rastrowa – prezentacja obrazu za pomocą pionowo – poziomej siatki odpowiednio kolorowanych pikseli
recipient – odbiorca
recompile – rekompilować
rectilinear – prostoliniowy
recur – powtarzać się
recursive – rekurencyjny
recurrent – powracający
redundancy – redundancja, nadmiar
relational database – relacyjna baza danych
reliability – niezawodność
repository – repozytorium, składnica
resource – zasób
retain – zachować
retrieval – odzyskanie, wyszukiwanie
rigid – sztywny
rigorous – rygorystyczny, surowy
robust – solidny, zdeterminowany
root – pierwiastek, korzeń (drzewa)
route – trasa

S

scalability – skalowalność
scope – zakres
sequence – sekwencja, ciąg
spatial – przestrzenny
staffing – zatrudnianie
stakeholder – bukmacher, udziałowiec
steganography – steganografia
stem – trzon, rdzeń
storage – pamięć, zapamiętywanie
string – łańcuch
submit – dostarczyć, przedłożyć
subset – podzbiór
sufficient – wystarczający
suffix – przyrostek
suite – pakiet

T

tabular – tabelaryczny
termination – zakończenie
therapy – terapia
threshold – próg
tier – poziom, sekwencja
traceability – wykrywalność
transducer – przetwornik
transparency – przezroczystość, jawność, foliogram
trapdoor – luka w systemie
trigger – uruchamiać, odpalić
tuning – dopasowywanie
tuple – krotka

U

ubiquitous – wszechobecny
unambiguous – niedwuznaczny, jednoznaczny
unauthorized – bezprawny, nieuprawniony
unsolvability – nierozwiązalność
utilization – wykorzystanie

V

validate – potwierdzić, uprawomocnić
vendor – dostawca
versatile – wielofunkcyjny
violation – naruszenie
virtual reality – rzeczywistość wirtualna
visualization – wizualizacja
volatile – ulotny

W

wrist – nadgarstek, przegub

Dictionaries Consulted

[1] Fisiak J., Adamska-Sałaciak A., Idzikowski M., Jagła E., Jankowski M., Lew R.: Słownik współczesny, angielsko-polski, polsko-angielski. Pearson Education Ltd 2004.
[2] Jezierska H.: Angielsko-polski słownik matematyczny. Warszawa, WNT 2003.
[3] Linde-Usiekniewicz J.: Wielki słownik angielsko-polski, polsko-angielski. Warszawa, Wyd. Naukowe PWN 2004
[4] Skrzyńska M., Jaworska T., Romkowska E.: Mały słownik techniczny, angielsko-polski, polsko-angielski. Warszawa, WNT 2003.
[5] Skrzyńska M., Czerni S., Jaworska T., Romkowska E.: Słownik naukowo-techniczny, angielsko-polski, polsko-angielski. Warszawa, WNT 1997.
[6] The Free Online Dictionary.
[7] PWN Oxford, Wielki słownik angielsko-polski, polsko-angielski. Warszawa, 2007.
[8] Polish-English/English-Polish Dictionary of Science and Technology. Warszawa, WNT 2008.
[9] Dictionary of Contemporary English for Advanced Learners, Pearson Education Ltd 2009.

Key to exercises

Unit 1 Key

Ex. 1

1g, 2e, 3h, 4d, 5f, 6a, 7b, 8c, 9k, 10j, 11l, 12i

Ex. 3

1. It is a digital electronic circuit with a specialized microprocessor and with necessary devices for its independent work placed in one chip. Hence, it is able to work autonomically, i.e. it doesn't need external devices (e.g. bus controllers, system clock generators, reset circuits). It was designed to work in control-measurement systems and communication systems. Therefore, it owns a complex communication system to connect with its surroundings. It works mostly in real time.
2. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, medical devices, remote controls, office machines, appliances, power tools, and toys. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes.
3. The microprocessor contains its own control unit and arithmetic and logic unit (ALU). The latter unit handles arithmetic and logical operations, whereas the former unit ensures the execution of instructions received from the program memory. This division of labour facilitates adaptation to a variety of practical applications through appropriate programming.
4. Semiconductor memory of the microcontroller is used to store the following:
   - Data, such as I/O data, states, and intermediate results that often require rapid read and write access.
   - The executable program, which, in most cases, requires permanent storage.
   - Constant parameter sets, which also require permanent storage in many cases.
5. This stores the data that are changed as a consequence of program execution.

Unit 2 Key

Ex. 1

1h, 2g, 3a, 4b, 5f, 6c, 7d, 8e, 9i, 10l, 11k, 12j

Ex. 3

1. personal digital assistant
2. The future will start to see the mobile phone as an alternative to a PC.

3. Currently released modern phones include some games, tools for social communication (i.e. Facebook, twitter) and make it possible to download additional applications for other purposes.
4. It provides tools for playing multimedia formats such as .m4a, .m4b, .mp3, .wma (Windows Media Audio 9).
5. Devices meant for business users: physical qwerty keyboard, large screen, push e-mail applications.
6. Such additional gadgets like digital video camera, accelerometer, media player, mobile internet browser, light sensor, GPS and others.

Ex. 8

Smartphones, hand-held devices and PDA are becoming part of our everyday life, making digital information accessible at every time and everywhere. One of the most popular applications of mobile technologies are so called location-based services. They are different from more conventional paper and internet based media (guides, directories, maps etc.) because they are aware of the context in which they are invoked. That particular advantage makes LBS capable of delivering specific information according to what is specifically desired by the user. Many of the solutions presented in the paper are already provided for users worldwide including business activities, weather information, traffic services and so on. The future of the sector is focused on the development of a fast wireless communication infrastructure and the steady growth of technologies providing tools and services for compact devices.

Ex. 10

1. are going to be
2. will start
3. will need,
4. will be,
5. will show.

Unit 3 key

Ex. 1

1d, 2a, 3h, 4b, 5e, 6c, 7g, 8f, 9j, 10k, 11l, 12i

Ex. 6

Diagram for text A

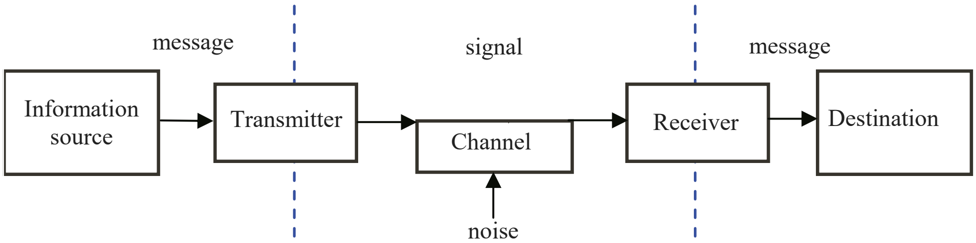
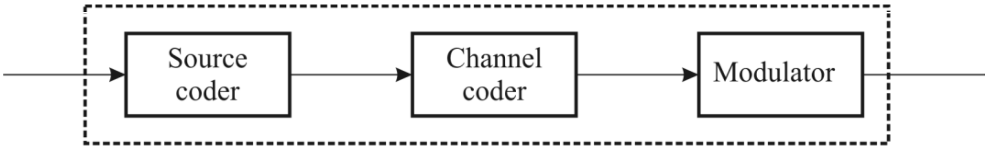


Diagram for text B



Ex. 8

1. is represented,
2. are reserved,
3. is sometimes called,
4. is used,
5. are needed,
6. is presented,
7. is used,
8. is coded.

Unit 4 key

Ex. 1

1b, 2h, 3a, 4c, 5g, 6d, 7e, 8f, 9k, 10i, 11l, 12j

Ex. 3

HCI – human-computer interface

Ex. 4

The first group utilises devices mounted directly on the user's body; in the second group they are contactless and they use remote sensors only.

Ex. 5

1. 90%
2. on average three errors
3. twice

4. 102
5. during the calibration stage
6. after the training process is completed
7. lip shape and lip region image features
8. people who cannot or do not want to use traditional input devices
9. to tune LipMouse to detect gestures made by the user in the current lighting conditions
10. to help disabled people / to detect and analyse images of the user's mouth region in a video stream acquired from a web-camera

Ex. 6

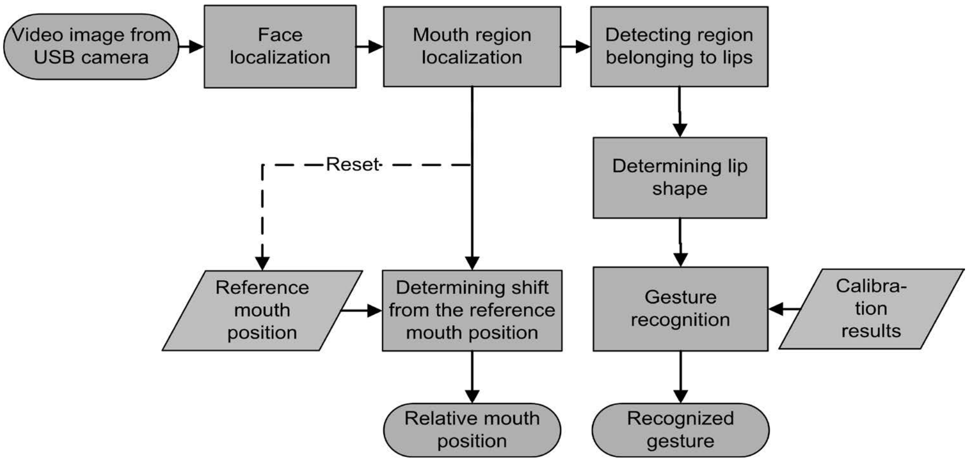


Fig. 6.2. Diagram of the LipMouse algorithm

Ex. 7

1. vision-based
2. patent-pending
3. display-mounted
4. web camera
5. screen cursor
6. mouse buttons
7. neural network
8. training set
9. lighting conditions
10. target user
11. input devices
12. disabled people

Unit 5 key

Ex. 1

1d, 2e, 3a, 4b, 5h, 6c, 7f, 8g, 9l, 10i, 11j, 12k

Ex. 2

Geographic:	**Computing:**
latitude	matrix
longitude	bitmap
density	raster
depth	grayscale
grid	retrieval
coordinate	multipatch
curve	node
elevation	cache
surface	pixel
floodplain	grid

Ex. 4

a. It refers to the location and characteristics of objects at every scale, such as continents, countries, cities, districts and buildings, down to single trees and lamp posts.
b. Geographic coordinates consist of latitude, longitude and height above sea level.
c. It is a software which allows for creation, manipulation, storage, analysis and visualization of spatial data.
d. The investigation of geographic data in a GIS (e.g. by means of spatial interaction, autocorrelation, regression or interpolation) is referred to as spatial analysis.
e. Geographic Information Systems have been created for the purpose of administering and monitoring the environment.
f. They operate on geographic data such as outlines and placement of buildings, parks or transport routes, as well as attributes such as addresses, post and zip codes.
g. A Geodatabase is a regular relational database which supports spatial extensions.

Ex. 8

D, C, E, B, A

Unit 6 key

Ex. 1

1h, 2g, 3a, 4c, 5b, 6f, 7d, 8e, 9i, 10k, 11l, 12j

Ex. 4

a. satisfied customer actually using and paying for the product
b. delivering a “good” system

c. Information systems engineering focuses on delivering a “good” system, while project management allows to deliver it effectively.
d. the set of functional features of a system
e. matching the scope of a delivered system with the actual needs of the users
f. a set of attributes such as reliability, performance, scalability or usability

Ex. 7

The steps are:
business modelling (eliciting the users’ needs
document analysis (discussion with stakeholders)
design of systems architecture (defining the overall structure of the system)
designing the constituent subsystems
software testing (detecting defects in the code)
software implementation (including user training)

Ex. 9

1. defects
2. users
3. quality
4. development
5. product
6. techniques
7. software
8. performance
9. problems
10. management

Ex. 10

1. It means keeping the project within assumed time and budget constraints.
2. It creates an advantageous environment for systems engineering to provide customers with effective IT solutions.
3. Selection of appropriate development methodology is one of the most essential success factors of IT projects.
4. Methodology is a set of principles and procedures driving the process of software development.
5. Plan-driven methodologies focus on well-defined processes with detailed schedules to follow, while risk-driven ones organize the project phases around risk reduction.
6. Project management covers a set of specific areas such as time management, budgeting, human resource management, configuration management, and risk management.
7. It deals with staffing, training, task delegation, motivation, and monitoring of the project team.
8. Effective team management requires many social (“soft”) competencies like communication skills, fast decision-making, empathy, intuition, as well as presentation and negotiation skills.
9. It provides for versioning resources with repositories such as Subversion or CVS, controlled resource sharing, document formatting and naming conventions and, last

but not least, the management of resource configurations into correct product packages.
10. It aims at reducing the risk of project failure with an organized process comprising risk identification, analysis and mitigation.
11. First, risks with their contributing factors should be identified using a check-list, questionnaire or brainstorming. Then, the risks are estimated in terms of likelihood and impact, and a number of the biggest risks are qualified to mitigation, while the rest are only monitored.
12. Risk mitigation employs various strategies such as reduction, avoidance, transfer or acceptance.

Unit 7 key

Ex. 1

1a, 2h, 3d, 4b, 5f, 6g, 7e, 8c, 9l, 10k, 11j, 12i

Ex. 4

1. standards
2. academic
3. environment
4. decision
5. computers
6. intelligence
7. sensors
8. information
9. actuators
10. system
11. context
12. user

Ex. 5

1. It specifies how sensors should integrate with ubiquitous environments and how their data should be processed.
2. The primary focus of the suite is to enable sensor-to-network plug-and-play integration.
3. Transducer Electronic Data Sheet describes transducer giving sensor type, measurement range, calibration data and user information.
4. They cover Sensor Web Enablement and IEEE 1451 Web Services Interface.
5. Typically, devices are integrated by a wireless communication with numerous standards available.

Ex. 7

1. enables
2. may easily be configured
3. manages
4. is equipped

5. be controlled
6. is typically used
7. is widely used
8. unlock
9. is detected
10. were used
11. appears
12. is connected
13. enable
14. is given
15. is realised

Ex. 8

The door lock service
Radio-frequency identification
Automatic file upload and download

Unit 8 key

Ex. 1

1f, 2g, 3b, 4c, 5h, 6d, 7e, 8a, 9k, 10l, 11i, 12j

Ex. 3

Karel Čapek

Ex. 4

1. false
2. false
3. false
4. true
5. false
6. true
7. false
8. true

Ex. 7

1. kinematic
2. environment
3. gripper
4. task
5. items
6. screws
7. object
8. knowledge
9. jaws

10. type
11. base
12. actuator

Ex. 9

An industrial robot is easy, but putting it into an intelligent work cell requires much more than the robot. Important accessories such as process tools, safety devices, programmable logic controllers, computers and simulation programs are needed to make robots easier to use. The control system is the set of logic and power functions that allows the automatic monitoring and control of mechanical structure and permits it to communicate with the other equipment and user in the environment. Open architecture control refers to software designs that can use or be used with products from a variety of manufacturers. The move towards open architecture controls is relatively recent but follows the trend in computers.

Ex. 10

1. bought/didn't work
2. will tell
3. would have let
4. would invest
5. would have finished
6. would be
7. used
8. had started
9. wouldn't have asked
10. wouldn't have opened

Unit 9 key

Ex. 1

1d, 2a, 3g, 4e, 5h, 6f, 7b, 8c, 9j, 10k, 11i, 12l

Ex. 5

1. information retrieval
2. picture
3. sequence
4. universe
5. languages

6./7. text retrieval/ document retrieval

8. query
9. document
10. documents
11. knowledge
12. sentence

Ex. 9

1. There is a large difference between the situation twenty years ago and now for people who are looking for a specific piece of information. Twenty years ago, they would have had to go to a bookstore or to a library, and not only to one. Now they have access to information from the Internet.
2. The problem is that there is so much information that rejecting irrelevant items and ranking the relevant ones is more difficult than finding anything that might be related to a *query*.
3. Information retrieval may contain pictures, sound, films and text; text retrieval deals with pure text.
4. A *bag of words* is a set of words without their context.
5. Certain words occur in all documents. They include determiners, prepositions, pronouns. The idf measure for them would therefore be zero. Such words are called *stopwords*.
6. A *recall* is the number of relevant documents returned by the text retrieval system divided by the number of relevant documents in the collection.
7. This not only improves the search, it also reduces the size of the indexes containing our vectors.
8. Extending the query with a list of synonyms may help finding relevant documents.
9. C*overage* is the number of documents for which the system can find the similarity measure divided by the total number of documents in the collection.

Unit 10 key

Ex.1

1d, 2e, 3g, 4c, 5h, 6b, 7f, 8a, 9k, 10j, 11l, 12i

Ex. 3

1. Video games are electronic games that involve interaction with a user interface to generate visual feedback on a video device.
2. Video games can be used for *entertainment* but can also be found in other fields, such as *education*, *training*, *practice*, *communication*, *social therapy*, *advertisement*, *popularization*, and others.
3. Game design is part of the process of game creation.
4. It requires artistic and programming skills.
5. It requires the cooperation of many staff members of the developing team.
6. Yes
7. The most important thing about games is a good **concept**.

Ex. 5

1. false
2. true
3. false
4. false

Ex. 10

1. Humans can entertain themselves by playing on three different levels, depending on their growing maturity.
2. Giving a toy a distinct goal to achieve brings it to the next level on which the toy becomes a **puzzle**.
3. Playing **multi-player games** requires social cooperation of the players.
4. Video games are a subset of a universe of all games that are mediated by a computer.
5. As a game world is usually fictional, creation of the pretended reality takes place in the imagination of the players.
6. Each game should have at least one goal that can be arbitrarily defined by a game designer.
7. **Rules** are definitions and instructions that the players agree to accept for the duration of the game.
8. **Actions** are any activities that can be performed by a player within a game
9. There are many factors that influence the design and production of an enjoyable and commercially successful game.
10. There are two fundamental misconceptions that must not occur during the design process.

Unit 11 key

Ex. 1

1c, 2g, 3h, 4a, 5b, 6d, 7f, 8e, 9i, 10l, 11j, 12k

Ex. 3

1. applications
2. processing
3. databases
4. related
5. software
6. system
7. foundations
8. base

Ex. 4

The recording describes the way of presenting information in a tabular form.

Ex. 5

1. According to Codd's theory, an essential element of a database is a *database relation*.
2. The restriction imposed by Codd on the mathematical relation is that domains of the components of the database relation must contain only simple (i.e. atomic) values.
3. This limitation is called the *first normal form* (1NF).
4. It presents a simple database relation in tabular form.

5. Each student has a name, a surname, a date of birth, and is assigned a unique album number. For each student information about the year of admission to the college is also stored.
6. The rows of this table represent *tuples*.
7. The columns represent attributes.

STUDENTS

Album_No	Surname	Name	Date_of_Birth	Date_of_Adm
143879	Kowalski	Jan	23.02.1992	01.10.2010
654889	Nowak	Marek	12.12.1989	01.10.2009
324672	Dworak	Krzysztof	05.06.1990	01.10.2008
321445	Ziółkowski	Stefan	01.06.1989	01.03.2010

Ex. 9

1. *Database Management System*
2. Relational Database
3. Object-oriented Database
4. Object-relational Database
5. Structured Query Language
6. Second Normal Form

Ex. 10

1. failed
2. have shown
3. showed
4. have not been completed
5. demonstrated
6. have been
7. have completed/haven't started
8. have developed
9. have investigated

Unit 12 key

Ex. 1

1g, 2a, 3h, 4b, 5d, 6e, 7f, 8c, 9l, 10k, 11i, 12j

Ex. 3

1. An *algorithm* is a finite sequence of well-defined operations, which transforms any proper string of input data into a proper string of output data and halts in a finite time.
2. Recipe, procedure, program, method, strategy, technique or computation.
3. It means that a computer program can be written which will produce the correct answer for any input if we let it run long enough and allow it as much storage space as it needs.

4. It concentrates on describing or characterizing problems that could be solved algorithmically and exhibiting some problems that could not be.
5. The *halting problem* involves deciding whether an arbitrarily given algorithm (or computer program) will eventually halt (rather than getting into an infinite loop) while working on a given input. A computer program that solves this problem cannot exist.
6. *A semidecidable problem* is a problem for which "yes" can be obtained in a finite time, while there is no such guarantee in the case of negative answer.
7. Yes, it is.
8. There is only a finite number of ways of arranging the chess pieces on the board, and a game must terminate after a finite number of moves.
9. The program could consider each of the computer's possible moves, each of its opponent's possible responses, each of its possible responses to those moves, and so on, until each sequence of possible moves reaches an end.
10. There are numerous problems with practical applications that have algorithmic solutions, but for which the time and storage requirements are much too great for the corresponding programs to be of practical use. Clearly, the time and space requirements of a computer program are of practical importance.

Ex. 12

1. What do we consider in this unit?
2. What is it possible to write?
3. What is it possible to do?
4. How many ways of arranging the chess pieces on the board are there?
5. What could consider each of the computer's possible moves?
6. What did much emphasis in the early work in this field concentrate on?
7. On what condition will it produce a correct answer for any input?
8. What does computability theory have obvious and fundamental implications for?
9. What is this computer program connected with?
10. What kind of computer programs do not exist?

KEY TO BIOGRAPHY QUIZ

Ex. 1

1. Charles Babbage
2. John Backus
3. Ada Lovelace
4. John Carmack
5. Seymour Cray
6. Tim Berners-Lee
7. Edsger W. Dijkstra
8. Marvin Minsky
9. Donald Erwin Knuth
10. Donald Erwin Knuth
11. John von Neumann
12. John von Neumann

13. Bjarne Stroustrup
14. Linus Torvalds
15. Linus Torvalds
16. Dennis Ritchie
17. Konrad Zuse
18. Niklaus Wirth
19. Alan Turing
20. Alan Turing

Ex. 3

NAME	NATIONALITY	CONTRIBUTION
David Huffmann	American	He made contributions to the study of finite state machines, switching circuits, synthesis procedures, and signal designs. David Huffman is best known for the invention of the Huffman code, a highly important compression scheme for lossless variable length encoding.
Charles Babbage	English	He originated the concept of a programmable computer.
John Backus	American	He assembled a team to define and develop Fortran for the IBM 704 computer. Fortran was the first high-level programming language to be put to broad use.
Ada Lovelace	English	Her notes contain what is considered the first computer program, that is, an algorithm encoded for processing by a machine.
John Carmack	American	He was the lead programmer of the id computer games *Wolfenstein 3D*, *Doom*, *Quake*, their sequels and the Commander Keen series of games. Though Carmack is best known for his innovations in 3D graphics, he is also a rocketry enthusiast and the founder and lead engineer of Armadillo Aerospace.
Seymour Cray	American	He designed a series of computers that were the fastest in the world for decades. Cray has been credited with creating the supercomputer industry.
Tim Berners-Lee	British	He implemented the first successful communication between a Hypertext Transfer Protocol (HTTP) client and server via the Internet. Berners-Lee is the director of the World Wide Web Consortium.
Edsger Dijkstra	Dutch	His contributions are: the *shortest path-algorithm*, also known as *Dijkstra's algorithm*; *Reverse Polish Notation* and the related *Shunting yard algorithm*;

		the THE multiprogramming system, the *Banker's algorithm*; and the semaphore construct for coordinating multiple processors and programs, the self-stabilization concept, Shortest Path First.
Marvin Minsky	American	Minsky's inventions include the first head-mounted graphical display and the confocal microscope He developed the first Logo "turtle". Minsky also built the first randomly wired neural network learning machine, SNARC.
Donald Knuth	American	He contributed to the development of the rigorous analysis of the computational complexity of algorithms and systematized formal mathematical techniques for it. Knuth is the creator of the TeX computer typesetting system, the related METAFONT font definition language and rendering system, and the Computer Modern family of typefaces.
John von Neumann	Hungarian	He developed set theory, functional analysis, quantum mechanics, ergodic theory, continuous geometry, economics and game theory, computer science, numerical analysis, hydrodynamics, and statistics.
Bjarne Stroustrup	Danish	He created and developed the widely used C++ programming language.
Linus Torvalds	Swedish	He developed the Linux kernel
Dennis Ritchie	American	He developed C and had an influence on other programming languages, as well as operating systems such as Multics and Unix.
Konrad Zuse	German	He invented the first high-level programming language, Plankalkül.
Niklaus Wirth	Swiss	He designed several programming languages, including Pascal.
Alan Turing	British	He provided a formalization of the concepts of "algorithm" and "computation" with the Turing machine, which played a significant role in the creation of the modern computer.

KEY TO BILL GATES AND STEVE JOBS EXERCISES

Steve Jobs: 1, 2, 5, 7, 10, 11, 13, 14
Bill Gates: 3, 4, 6, 8, 9, 12.

WYDAWNICTWO POLITECHNIKI GDAŃSKIEJ

Wydanie I. Ark. wyd. 6,7, ark. druku 9,0, 1001/718

Druk i oprawa: *EXPOL* P. Rybiński, J. Dąbek, Sp. Jawna
ul. Brzeska 4, 87-800 Włocławek, tel. 54 232 37 23