

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»  
КАФЕДРА ІНОЗЕМНИХ МОВ

**Навчально-методична розробка до курсу  
«Англійська мова» для спеціальності  
«Комп'ютерна інженерія»  
ч.2**

Ужгород – 2023

УДК 811.111(076):004  
А64

Англійська мова для студентів спеціальності «Комп'ютерна інженерія» (English for .....): Навчально-методична розробка до курсу «Англійська мова» для спеціальності «Комп'ютерна інженерія» / Уклад. Н.І.Годованець, В.П.Леган, І.О. Бура. Ужгород: Видавництво УжНУ «Говерла», - Ч-2, 2023. 76 с.

Рецензенти:

кандидат філологічних наук, доцент Герцовська Н. О.  
кандидат педагогічних наук, доцент Канюк О.Л.

Мета навчально-методичного посібника – забезпечити практичне оволодіння студентами лексичними та мовленнєвими моделями, необхідними для спілкування англійською мовою за фахом.

*Рекомендовано до друку  
Вченою радою факультету іноземної філології ДВНЗ «УжНУ»  
від 16 лютого 2023 року, протокол № 1.*

## ПЕРЕДМОВА

З прискоренням глобалізації і поширенням міжнародних ділових зв'язків України з іншими державами на політичному та економічному рівнях зростає потреба у висококваліфікованих фахівцях, які здатні вільно володіти основами ділового іншомовного спілкування. У таких умовах важливим є усвідомлення майбутніми фахівцями різних галузей необхідності їх майбутніх зв'язків із міжнародним середовищем, а одним з першочергових завдань освіти стає якісна підготовка фахівців, здатних до успішної професійної діяльності в межах світової спільноти. В цьому контексті істотно змінюються вимоги до володіння іноземною мовою фахівцями всіх рівнів, першорядного значення набувають практичні навички, що передбачають знання ділової іноземної мови в усному та писемному мовленні, вміння використовувати іноземну мову у своїй професійній діяльності.

Пропонований посібник має на меті ознайомити з основами іншомовного ділового спілкування, розширити словниковий запас за рахунок спеціальної лексики, виробити комунікативні навички ділового мовлення та навички ділового листування, організації ділових зустрічей та переговорів, оволодіння мовним матеріалом, необхідним під час здійснення ділових подорожей.

Посібник складається з тематичних розділів, кожен з яких містить 1) тексти інформативного характеру, які допоможуть орієнтуватися в певних ситуаціях професійної сфери, 2) лексичний матеріал з найчастіше вживаними мовленнєвими конструкціями відповідної тематики, та 3) систему вправ для успішного засвоєння та вдосконалення комунікативних навичок професійного спілкування.

## C O N T E N T S

|   |    |
|---|----|
| TOPIC 1 Programming Languages .....         | 5  |
| TOPIC 2. Imperative Mood.....               | 18 |
| TOPIC 3. Computer Networks.....             | 19 |
| TOPIC 4. Gerund.....                        | 24 |
| TOPIC 5. Computer software.....             | 28 |
| TOPIC 6. The Infinitive.....                | 32 |
| TOPIC 7. Computer architecture.....         | 35 |
| TOPIC 8. Conditionals.....                  | 39 |
| TOPIC 9. Storage and retrieval of Data..... | 41 |
| Topic 10.Computer Organisation.....         | 46 |
| Vocabulary .....                            | 49 |
| LITERATURE USED.....                        | 75 |

## **TOPIC 1.**

### **PROGRAMMING LANGUAGES**

Computer languages are generally classed as being “high-level” (like Pascal, Fortran, Ada, Modula-2, Oberon, C or C++) or “lowlevel” (like ASSEMBLER). High-level languages may further be classified as “imperative” (like all of those just mentioned), or “functional” (like Lisp, Scheme, ML, or Haskell), or “logic” (like Prolog). High-level languages are claimed to possess several advantages over low-level ones:

- **Readability:** A good high-level language will allow programs to be written that in some ways resemble a quasi-English description of the underlying algorithms. If care is taken, the coding may be done in a way that is essentially self-documenting, a highly desirable property when one considers that many programs are written once, but possibly studied by humans many times thereafter.
- **Portability:** High-level languages, being essentially machine independent, hold out the promise of being used to develop portable software. This is software that can, in principle (and even occasionally in practice), run unchanged on a variety of different machines — provided only that the source code is recompiled as it moves from machine to machine. To achieve machine independence, high-level languages may deny access to low-level features, and are sometimes spurned by programmers who have to develop low-level machine dependent systems. However, some languages, like C and Modula-2, were specifically designed to allow access to these features from within the context of high-level constructs.
- **Structure and object orientation:** There is general agreement that the structured programming movement of the 1960’s and the object-oriented movement of the 1990’s have resulted in a great improvement in the quality and reliability of code. High-level languages can be designed so as to encourage or even subtly enforce these programming paradigms.
- **Generality:** Most high-level languages allow the writing of a wide variety of programs, thus relieving the programmer of the need to become expert in many diverse languages.

• Brevity: Programs expressed in high-level languages are often considerably shorter (in terms of their number of source lines) than their low-level equivalents. • Error checking: Being human, a programmer is likely to make many mistakes in the development of a computer program. Many highlevel languages — or at least their implementations — can, and often do, enforce a great deal of error checking both at compile-time and at run-time. For this they are, of course, often criticized by programmers who have to develop time-critical code, or who want their programs to abort as quickly as possible. These advantages sometimes appear to be overrated, or at any rate, hard to reconcile with reality. For example, readability is usually within the confines of a rather stilted style, and some beginners are disillusioned when they find just how unnatural a high-level language is. Similarly, the generality of many languages is confined to relatively narrow areas, and programmers are often dismayed when they find areas (like string handling in standard Pascal) which seem to be very poorly handled. The explanation is often to be found in the close coupling between the development of high-level languages and of their translators. When one examines successful languages, one finds numerous examples of compromise, dictated largely by the need to accommodate language ideas to rather uncompromising, if not unsuitable, machine architectures. To a lesser extent, compromise is also dictated by the quirks of the interface to established operating systems on machines. Finally, some appealing language features turn out to be either impossibly difficult to implement, or too expensive to justify in terms of the machine resources needed. It may not immediately be apparent that the design of Pascal (and of several of its successors such as Modula-2 and Oberon) was governed partly by a desire to make it easy to compile. It is a tribute to its designer that, in spite of the limitations which this desire naturally introduced, Pascal became so popular, the model for so many other languages and extensions, and encouraged the development of superfast compilers such as those found in Borland's Turbo Pascal and Delphi systems.

### Exercise 1.

Find in text the English for: більша частина; мати перспективу; з точки зору; в усякому разі; якщо потурбуватися; розчаровувати; привабливий; за умови, що; треба віддати належне; узгоджувати(ся) з реальністю.

Exercise 2. Learn the words below, then read and translate text 2.

judgement — розсудливість, здоровий глузд controversy — суперечка, дискусія, полеміка exacerbate — посилювати, поглиблювати typify — бути типовим представником, уособлювати tempting — спокусливий, привабливий cripple — завдавати шкоди, калічити 154 conceivable — мислимий orthogonality — ортогональність host — велика кількість, безліч blemish — недолік, вада vulnerable — вразливий modularity — модульність self-consistency — самоузгоджуваність, несуперечність

### Text 2

#### **SOME ASPECTS OF LANGUAGE DESIGN**

The design of a programming language requires a high degree of skill and judgement. There is evidence to show that one's language is not only useful for expressing one's ideas. Because language is also used to formulate and develop ideas, one's knowledge of language largely determines how and, indeed, what one can think. In the case of programming languages, there has been much controversy over this. For example, in languages like Fortran — for long the lingua franca of the scientific computing community — recursive algorithms were “difficult” to use (not impossible, just difficult!), with the result that many programmers brought up on Fortran found recursion strange and difficult, even something to be avoided at all costs. It is true that recursive algorithms are sometimes “inefficient”, and that compilers for languages which allow recursion may exacerbate this; on the other hand it is also true that some algorithms are more simply explained in a recursive way than in one which depends on explicit repetition (the best examples probably being those associated with tree manipulation). There are two divergent schools of thought as to how programming languages should be designed. The one, typified by the Wirth school, stresses that languages should be small and understandable,

and that much time should be spent in consideration of what tempting features might be omitted without crippling the language as a vehicle for system development. The other, beloved of languages designed by committees with the desire to please everyone, packs a language full of every conceivable potentially useful feature. Both schools claim success. The Wirth school has given us Pascal, Modula-2 and Oberon, all of which have had an enormous effect on the thinking of computer scientists. The other approach has given us Ada, C and C++, which are far more difficult to master well and extremely complicated to implement correctly, but which claim spectacular successes in the marketplace. Other aspects of language design that contribute to success include the following:

- Orthogonality: Good languages tend to have a small number of well thought out features that can be combined in a logical way to supply more powerful building blocks. Ideally these features should not interfere with one another, and should not be hedged about by a host of inconsistencies, exceptional cases and arbitrary restrictions. Most languages have blemishes — for example, in Wirth’s original Pascal a function could only return a scalar value, not one of any structured type. Many potentially attractive extensions to well-established languages prove to be extremely vulnerable to unfortunate oversights in this regard.

- Familiar notation: Most computers are “binary” in nature. Blessed with ten toes on which to check out their number-crunching programs, humans may be somewhat relieved that high-level languages usually make decimal arithmetic the rule, rather than the exception, and provide for mathematical operations in a notation consistent with standard mathematics. When new languages are proposed, these often take the form of derivatives or dialects of well-established ones, so that programmers can be tempted to migrate to the new language and still feel largely at home — this was the route taken in developing C++ from C, Java from C++, and Oberon from Modula-2, for example. Besides meeting the ones mentioned above, a successful modern high-level language will have been designed to meet the following additional criteria:



- Clearly defined: It must be clearly described, for the benefit of both the user and the compiler writer.
- Quickly translated: It should admit quick translation, so that program development time when using the language is not excessive.
- Modularity: It is desirable that programs can be developed in the language as a collection of separately compiled modules, with appropriate mechanisms for ensuring self-consistency between these modules.
- Efficient: It should permit the generation of efficient object code.
- Widely available: It should be possible to provide translators for all the major machines and for all the major operating systems. The importance of a clear language description or specification cannot be over-emphasized. This must apply, firstly, to the so-called syntax of the language — that is, it must specify accurately what form a source program may assume. It must apply, secondly, to the so called static semantics of the language — for example, it must be clear what constraints must be placed on the use of entities of differing types, or the scope that various identifiers have across the program text. Finally, the specification must also apply to the dynamic semantics of programs that satisfy the syntactic and static semantic rules — that is, it must be capable of predicting the effect any program expressed in that language will have when it is executed.

**Exercise 3. Find in text 2 the English for:** вразливий; філософський напрям; суперечка; безліч протиріч; мислимий; мати успіх; прикрій недогляд; за будь-яку ціну; щодо цього.

**Exercise 4. Translate into English paying special attention to the italicized words** 1. Цей студент має перспективу стати гарним програмістом за умови, що наполегливо працюватиме. 2. Ми були розчаровані, дізнавшись, що більша частина отриманої інформації не відповідала дійсності (не узгоджувалася з реальністю). 3. Треба віддати належне розробникам цього літака у тому, що він має успіх. 4. Через прикрій недогляд на початковій стадії (at the outset of) розробки цієї системи вона виявилася вразливою до

багатьох зовнішніх факторів. 5. Зараз не існує суперечок щодо того, що бортові обчислювальні системи повинні розроблятися передусім в інтересах забезпечення надійності авіаційної техніки та безпеки польоту. 6. У будь-якому разі важливість цього проекту не треба переоцінювати, яким би привабливим він не видавався. 7. Поставлене завдання треба виконати за будь-якою ціною. Жоден із нас не має ніяких сумнівів щодо цього. 8. Якщо докласти певних зусиль, то більшу частину недоліків даного проекту можна усунути вже найближчим часом. 9. У межах даного філософського напрямку існує безліч протиріч щодо концепцій подальшого розвитку інженерії програмного забезпечення. 10. Запропоноване технічне рішення видається досить привабливим передусім із точки зору можливості протидіяти будь-яким мислимим втручанням ззовні.

Additonal information to the topic:

### **OPERATING SYSTEMS**

An operating system is software that manages the overall operation of the computer system. Its primary purpose is to support application programs. The parts of an operating system can be grouped into four broad categories. One set of parts forms the shell or user interface; another set of parts is responsible for coordinating multiple computers in a network; a third set coordinates multiple tasks or basic units within a single computer; and, finally, the kernel of the operating system is software that ties the hardware to the software and performs such tasks as keeping track of everything in memory and managing the flow of information to and from disks, the keyboard, and the display screen. An analogy can be drawn between a computer user and someone who attends a live performance, such as a play or concert. When you attend a live performance, you are seeing the end results of a carefully coordinated production. What you do not see in a good performance are the technical details, the behind-the-scene activities that are making the performance run smoothly. Technicians manage light and sound, stagehands manage props and change scenes, and actors change costumes and wait for their cues. Similarly, an operating system manages the details needed to support a

smoothly running application program. All application programs share some tasks in common. They include accepting characters typed at the keyboard, displaying information on the screen, managing information on a disk, and managing information in memory. The operating system takes care of the details of these tasks. A most important example of how operating systems support application programs is the task of managing files. A file is a named collection of information. Whether your application is general or special purpose, your program needs to store information in files. By itself, a disk is an empty platter waiting to be filled. The operating system takes care of: formatting the disk, which involves electronically preparing the disk to be able to store files; managing the location of information on the disk; checking to make sure that errors do not occur when reading to and writing from the disk; performing the input and output necessary to retrieve and store information on the disk. Operating systems also manage the other components of a computer system. They support programs called device drivers that control the various hardware devices, such as the keyboard, display screen, and printer. The device driver translates instructions from the application into commands the hardware understands. For example, if an application program wants to print something, it simply sends the information and the appropriate instructions to the operating system, which, in turn, calls upon the printer device driver to manipulate the printer to perform the desired task. A single-tasking operating system runs one application program at a time. With multitasking operating systems, you can have more than one application program or task active at a time. The operating system takes care of the details that are required to keep track of each task, know where the task's data are in memory, and allocate a fair share of computing services to each task. It is common to connect groups of computers into networks so people can share information and communicate with one another via electronic mail. Some operating systems provide network services such as file-sharing, print-sharing, and electronic mail. Some operating systems allow integration of dynamic data types such as sound, animation, and video. To accomplish this, the operating system uses multimedia extensions — time-based

synchronizing software for managing the coordination of video, sound, and animation. For example, a film might contain a video track and a sound track that need to be coordinated when the film is played on a computer's display screen.

***Learn the words:***

pull-down menu — низхідне меню

scroll bar — лінійка прокрутки

volume — великий, великомасштабний

power user — досвідчений, кваліфікований користувач; користувач із підвищеними вимогами до продуктивності комп'ютера

DLL (Dynamic-Link Library) — бібліотека, що динамічно підключається; бібліотека динамічного компонування (зв'язків)

plague — надокучати, набридати

not to sit well with sb — не подобатися кому-небудь

security vulnerabilities — вразливості захисту

default user interface — стандартний інтерфейс користувача concern — проблема (яка потребує вирішення)

peer-to-peer technology — технологія однорангового або прямого зв'язку

garner — накопичувати, збирати

***COMPARING OPERATING SYSTEMS*** Although in theory all operating systems perform very similar functions, in practice there are many different operating systems. One primary reason is the highly competitive nature of the computer industry. Operating systems, like other major computer products, evolve and undergo improvements and updates over time. In addition, applications are developed to take advantage of specific features in operating systems, so applications developed for one operating system may not be available for other operating systems. MS-DOS. The ancestor of MS-DOS was developed in the early 1970s by Tim Patterson of Seattle Computer Products Inc. Later, Microsoft Corporation acquired the rights and began licensing it as MS-DOS (Microsoft Disk-Operating System). MS-DOS is a single-tasking operating system. Newer versions have a command-driven user interface. Their commands can be used to

format disks; copy, rename, delete, and back up files; and organize and manage files on the disk. Newer versions also include a user interface called the shell with pull-down menus to access commands. MS-DOS has several advantages. An extensive number of applications have been written for MS-DOS, so you can generally find an application to meet your needs. MS-DOS does not require a powerful computer or a large amount of memory to run. It also has several disadvantages. MS-DOS was specifically developed for the Intel family of microprocessors. It has no network services or multimedia extensions. And it has no limitations on how it uses memory. Also, MS-DOS does not provide a user interface for application programs. Thus, applications from, say, Lotus Development Corporation and Borland International present the user with completely different interfaces. A graphical user interface allows a user to learn one interface that works with all applications. MS-DOS users who want a graphical user interface can obtain Microsoft Windows, an extension to MS-DOS that provides standard methods for switching among applications, file and program management, windowing, icons, pull-down menus, scroll bars and dialog boxes — temporary windows that contain choices when the program needs additional information from the user. To take advantage of these features, most software vendors have released Windows versions of their MS-DOS application programs.

Windows NT. Windows NT (New Technology), introduced in 1992, is a descendant of DOS and the Windows extension to DOS developed at Microsoft. It is a multitasking operating system. It has several advantages. It can run DOS and Windows programs without modification. It has network services and multimedia extensions, and it is not limited to the Intel family of processors. It has several disadvantages. It is designed for high-performance computers and requires large amounts of memory and disk space. It does not have as many applications developed for it as do other operating systems.

Windows XP. Windows XP is a line of operating systems developed by Microsoft for use on personal computers, including home and business desktops, notebook computers, and media centers. The name “XP” stands for eXPerience. It was codenamed “Whistler”, after

Whistler, British Columbia, as many Microsoft employees skied at the Whistler-Blackcomb ski resort during its development. Windows XP is the successor to both Windows 2000 Professional and Windows Me, and is the first consumer-oriented operating system produced by Microsoft to be built on the Windows NT kernel (version 5.1) and architecture. Windows XP was first released on October 25, 2001, and over 400 million copies were in use in January 2006. It is succeeded by Windows Vista, which was released to volume license customers on November 8, 2006 and worldwide to the general public on January 30, 2007. The most common editions of the operating system are Windows XP Home Edition, which is targeted at home users, and Windows XP Professional, which has additional features such as support for Windows Server domains and two physical processors, and is targeted at power users and business clients. Windows XP Media Center Edition has additional multimedia features enhancing the ability to record and watch TV shows, view DVD movies, and listen to music. Windows XP Tablet PC Edition is designed to run the ink-aware Tablet PC platform. Two separate 64-bit versions of Windows XP were also released, Windows XP 64-bit Edition for IA-64 (Itanium) processors and Windows XP Professional x64 Edition for x86-64. Windows XP is known for its improved stability and efficiency over the 9x versions of Microsoft Windows. It presents a significantly redesigned graphical user interface, a change Microsoft promoted as more user-friendly than previous versions of Windows. New software management capabilities were introduced to avoid the “DLL hell” that plagued older consumer-oriented 9x versions of Windows. It is also the first version of Windows to use product activation to combat software piracy, a restriction that did not sit well with some users and privacy advocates. Windows XP has also been criticized by some users for security vulnerabilities, tight integration of applications such as Internet Explorer 6 and Windows Media Player, and for aspects of its default user interface. Later versions with Service Pack 2, and Internet Explorer 7 addressed some of these concerns. Windows Vista. Windows Vista is a line of operating systems developed by Microsoft for use on personal computers, including home and business desktops, laptops, Tablet PCs, and media

centers. Prior to its announcement on July 22, 2005, Windows Vista was known by its codename “Longhorn. The release of Windows Vista comes more than five years after the introduction of its predecessor, Windows XP, the longest time span between successive releases of Microsoft Windows. Windows Vista contains many changes and new features, including an updated graphical user interface and visual style dubbed Windows Aero, improved searching features, new multimedia creation tools such as Windows DVD Maker, and redesigned networking, audio, print, and display sub-systems. Vista also aims to increase the level of communication between machines on a home network, using peer-to-peer technology to simplify sharing files and digital media between computers and devices. Windows Vista includes version 3.0 of the .NET Framework, which aims to make it significantly easier for software developers to write applications than with the traditional Windows API. Microsoft’s primary stated objective with Windows Vista, however, has been to improve the state of security in the Windows operating system. One common criticism of Windows XP and its predecessors has been their commonly exploited security vulnerabilities and overall susceptibility to malware, viruses and buffer overflows. In light of this, Microsoft chairman Bill Gates announced in early 2002 a company-wide “Trustworthy Computing initiative” which aims to incorporate security work into every aspect of software development at the company. Microsoft stated that it prioritized improving the security of Windows XP and Windows Server 2003 above finishing Windows Vista, thus delaying its completion. While these new features and security improvements have garnered positive reviews, Vista has also been the target of much criticism and negative press. Criticism of Windows Vista has targeted high system requirements, its more restrictive licensing terms, the inclusion of a number of new digital rights management technologies aimed at restricting the copying of protected digital media, lack of compatibility with certain pre-Vista hardware and software, and the number of authorization prompts for User Account Control. As a result of these and other issues, Vista has seen adoption and satisfaction rates lower than Windows XP. Operating System/2 (OS/2). When IBM decided to introduce a

second generation of personal computers in 1987, it named them Personal Systems/2. The operating system, called Operating System/2 (OS/2) is an operating system for this generation of personal computers. It was developed under a joint agreement between IBM and Microsoft. OS/2 is a multitasking operating system. The latest version has a graphical user interface called the Workplace that provides a window based user interface for the operating system's file management functions and allows applications to share data among themselves. OS/2 has several advantages. It has network services and multimedia extensions. However, it also has disadvantages. It was specifically designed for the Intel family of microprocessors. It requires a high-performance computer with large amounts of memory and disk space. It does not have as many applications developed for it as do other operating systems. Macintosh Operating System. Apple's Macintosh Operating System, simply called System, was introduced in 1984. It is a descendant of research done at Xerox PARC (Palo Alto Research Center) and Apple's Lisa computer, the user interface of which was designed by a team headed by Larry Tesler. The latest version, System 7, is a multitasking operating system. Its graphical user interface, called the Finder, has extensive graphics capabilities and is known for being very easy to use. System 7 has several advantages. It has network services and multimedia extensions. It also has disadvantages. It was developed specifically for the Motorola family of processors, although it can also run on the PowerPC microprocessors — a family of microprocessors jointly developed by Apple, IBM, and Motorola. UNIX. UNIX was developed by Ken Thompson and Dennis Ritchie in 1969 at Bell Laboratories, the research and development arm of AT&T. Now available commercially from over 30 vendors, UNIX has been popular with programmers and scientific and engineering customers because of its flexibility. UNIX is a multitasking multiuser operating system licensed by Novell Corporation. There are two standard versions of UNIX: System V.4 is offered by UNIX International, an organization led by AT&T, Novell, and Sun Microsystems OSF/1 is offered by the Open Software Foundation, an organization led by IBM, Digital, and Hewlett Packard. Both versions offer



similar graphical user interfaces. Unix's major advantage is that it is a very modular operating system that can be assembled like building blocks to perform various functions. It has network services and is processor independent. UNIX suffers from a lack of standardization: programs that run under one version often do not run under another version. Computer programmers with years of experience find UNIX commands hard to learn, and text editors (vi, emacs) are equally hard to use.

**Exercise 5 Find in the text the English for:** зазнавати поліпшень і вдосконалень із часом; використовувати характерні особливості операційних систем; призначений для домашніх користувачів; призначений для кваліфікованих користувачів і бізнес-клієнтів; боротися з комп'ютерним піратством; обмеження, яке не дуже сподобалося деяким користувачам і прихильникам конфіденційності; головна проголошена мета компанії Майкрософт щодо Windows Vista; чутливість до шкідливого ПЗ, вірусів та переповнення буфера; включити роботу щодо захисту в усі види розробки програмного забезпечення; за спільною угодою між компаніями Ай-Бі-Ем та Майкрософт. **Exercise 6.** Discuss with your group-mates the peculiarities of different operating systems using the words: aware of sth — 1) обізнаний, поінформований;  
2) свідомий чого-небудь fortunate — щасливий  
program entry — введення програми  
that is not to claim that — йдеться не про те, що; ми не стверджуємо, що  
hexadecimal — шістнадцятковий  
clumsy — незграбний  
source language — вхідна мова  
target language — вихідна мова  
fail to do sth — не змогти що-небудь зробити  
trap — пастка  
blur — робити нечітким, затуманювати  
underlying — прихований; той, що мається на увазі

## TOPIC 2

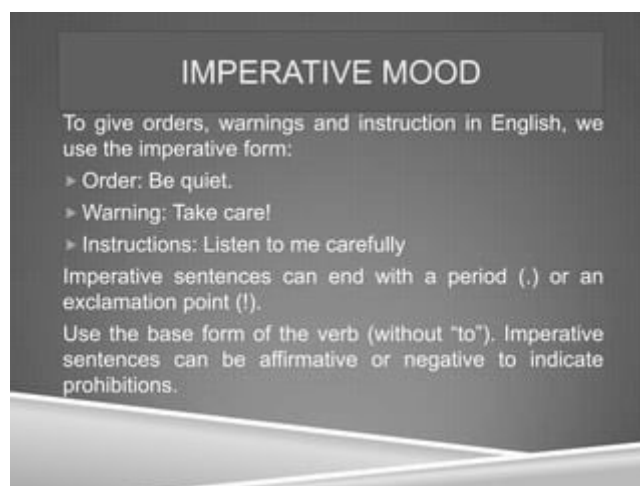


Рис.1

| Infinitive | Base  | Affirmative Imperative | Negative Imperatives |
|------------|-------|------------------------|----------------------|
| to be      | be    | be                     | do not be            |
| to do      | do    | do                     | do not do            |
| to eat     | eat   | eat                    | do not eat           |
| to drink   | drink | drink                  | do not drink         |
| to go      | go    | go                     | do not go            |
| to sleep   | sleep | sleep                  | do not sleep         |
| to stay    | stay  | stay                   | do not stay          |
| to stop    | stop  | stop                   | do not stop          |
| to study   | study | study                  | do not study         |
| to wait    | wait  | wait                   | do not wait          |

### 1. Put the verbs into imperative mood

1.  upstairs. (*to go*)
2.  in this lake. (*not/to swim*)
3.  your homework. (*to do*)
4.  football in the yard. (*not/to play*)
5.  your teeth. (*to brush*)
6.  during the lesson. (*not/to talk*)
7.  the animals in the zoo. (*not/to feed*)
8.  the instructions. (*to read*)
9.  late for school. (*not/to be*)
10.  your mobiles. (*to switch off*)

## TOPIC 3

### COMPUTER NETWORKS

#### *Active vocabulary*

token — маркер

susceptible — чутливий, вразливий

occur — траплятися

release — звільняти

hierarchy network — ієрархічна мережа

trunk node — магістральний вузол

branch node — вузол розгалуження

split into — розділятися на

mesh topology — змішана топологія (є дві топології ділянкових мереж: 1) full

mesh — кожен вузол має канал, який з'єднує його з кожним іншим вузлом

мережі; 2) partial mesh — деякі вузли можуть мати зв'язок лише з одним або двома іншими вузлами)

redundancy — надлишок, дублювання (методи захисту від збоїв шляхом дублювання основних пристроїв системи, додаванням надлишкових даних у повідомлення, яке пересилається та ін.)

broadcast — передавати; розповсюджувати

exhibit — 1) показувати; 2) виставляти

**NETWORK CONFIGURATIONS** Configuration or topology is the way in which computers are connected into a network. In other words, network topology refers to the shape of a network, or the network's layout. How different nodes in a network are connected to each other and how they communicate is determined by the network's topology. Topologies are either physical or logical. Physical topology means the physical design of a network including the devices, location and cable installation. Logical topology refers to how data is actually transferred in a network as opposed to its physical design. In general, physical topology relates to a core network whereas logical topology relates to basic network. Here is a description of the most common network topologies. Star. In the star configuration the central computer or the special device, which is called a hub, performs all processing and control functions. All access devices are linked directly to the central computer or hub and this central device sends a token to all computers. Nodes communicate across the network by passing data, address and token through the hub. The star configuration has two major limitations. First of all, the remote devices are unable to communicate directly. Instead, they must communicate via the central device only. Secondly, the star network is very susceptible to failure, either in the central device or the transmission links. Bus/Ethernet. A bus network

consists of one piece of cable terminated at each end to which all devices (computers) are connected. This piece of cable is called the bus or backbone. The Star Configuration detected silence for a fixed period of time. And each computer is listening, if data, which are now transmitted in the bus, are addressed to it. If it is so it accepts it. When a computer wants to send some data it first sends address of the computer, to which data will be sent and then it sends data. The only problem occurs when two devices try to send at the same time. When a sending device detects another's transmission, it aborts its own and should wait until it finishes. Each device is attached to a network shaped as a continuous loop. To put it simply, the first computer is connected to the second, second to the third, and so on, and the last one is connected to the first so they create a ring. Data proceeds in only one direction and at a constant speed round the loop. Devices may send information only when they are in control of the 'token'. The token is a package of data that indicates which device has control. If the computer wants to send some information it waits until the token comes to it, and only then it sends address of a receiver and data. Then these data and address begin to travel from one computer to another with the token and, when they achieve the receiver, the token is released and travels across the ring again. If there is no receiving computer in the network, the sending computer releases the token when it comes to it. Only one device may send data at any given moment, and each device must be working for the network to function.

Tree. It is also known as a hierarchy network. This is a hybrid topology. Groups of star-configured networks are connected to a linear bus backbone. A hierarchical topology is created similar to an extended-star topology. A Ring Network use a central node. Instead, it uses a trunk node from which it branches to other nodes. Two types of tree topologies exist: the binary tree (each node splits into two links) and the backbone tree (a backbone trunk has branch nodes with links hanging from it). The tree topology is the most wide-spread nowadays.

Mesh. In this topology all devices are connected with many redundant interconnections between network nodes. In a true mesh topology each node has a connection to all the other nodes in the network .There is a full-mesh topology and a partial-mesh topology. The Mesh Topology The full-mesh topology connects all devices (nodes) to each other for redundancy and fault tolerance. In the partial-mesh topology, at least one device maintains multiple connections to others without being fully meshed. Two common examples for hybrid network are: star-ring network and star-bus network. A Snowflake topology is really a "Star of Stars" network, so it exhibits characteristics of a hybrid network topology but is not composed of two different basic network topologies being connected together.

**Exercise 1 Find in text the English for:** інакше кажучи; топологія мережі; фізична / логічна топологія; фізична схема мережі; бути вразливим до неполадок; сталий період часу; безперервний цикл; простіше кажучи; стала швидкість; контролювати маркер; пакет даних; ієрархічна топологія; розширена топологія «зірка»; багато надлишкових взаємозв'язків; повністю / частково змішана топологія.

**Exercise 2. Translate into English paying special attention to the italicized words.** 1. В ієрархічній топології замість центрального вузла використовується магістральний вузол, від якого мережа відгалужується до інших вузлів. 2. Простіше кажучи, всі комп'ютери в конфігурації «кільце» з'єднані у безперервний цикл, і дані в такій мережі передаються зі сталою швидкістю. 3. Повністю змішана топологія має багато надлишкових взаємозв'язків між усіма вузлами мережі. 4. Інакше кажучи, частково змішана топологія мережі більше вразлива до неполадок через відсутність взаємозв'язків деяких її вузлів. 5. У розширеній топології «зірка» пристрої контролюють маркер. 6. Фізична топологія мережі зумовлює вибір типу комутаційного обладнання. 7. Канальний рівень протоколів описує, як передаються пакети даних через фізичний рівень, включаючи кодування. 8. Нове прикладне програмне забезпечення дає можливість фінансовим управлінням робити виписки будь-яких операцій за сталий період часу.

**Additional info:**

**CLASSIFICATION OF NETWORKS** Like snowflakes, no two networks are alike. The best way to classify them is by some general characteristics. A given network can be characterized by its: – size: the geographic size of the network; – security and access: who can access the network and how access is controlled; – protocol: the rules of communication in use on it (for example, TCP/IP, NetBEUI, or AppleTalk); – hardware: the types of physical links and hardware that connect the network (adapters, cross-bar or coaxial cable, hubs, routers). Regarding the size, there are two main types of modern computer networks Local Area Network (LAN) and Wide Area Network (WAN). The first one is a network of some enterprise, university, and office. A LAN is typically housed in one building or campus. And the second type network is a network, which connects other networks. Computers of these networks may be situated in different parts of the world. An example of a Wide Area Network is the Internet. The simplest network is two computers connected by a wire to each other. They can exchange information. For example such well-known shell as Norton Commander allows you to do that. But there are more than two computers that are needed to be connected, you have to think how to do that. In the case of a LAN there are two

types of networks concept: – peer-to-peer network concept; – client-server network concept or network with dedicated server. This type of networks is sometimes named as dedicated file-server or hierarchical network. In the network of the first type all computers of the network work as clients and servers simultaneously. All computers are equal; they handle security and administration for themselves. Each computer has a unique name (computer name) and usually the password for an input (entrance) in it in load time of operating system. The owner of the PC resources of operating system assigns the name and the password of an input (entrance). The users must make the decisions about who gets access to what. Peer-to-peer networks may be organized with the help of such operating systems, as LANtastic, Windows 3.11, Novell NetWare Lite. The indicated programs work both with a DOS, and with a Windows. Peer-to-peer networks may be organized also on the basis of all modern 32-bit operating systems — Windows 7, Windows NT Workstation version, OS/2 and some other. Such networks usually are not big and include up to 10 computers. Often they are called workgroups. On the contrary, in networks with dedicated server most of computers work as clients and one or several computers work as servers. The latter ones usually execute one certain task (controlling access to the printer, storing files for sharing, general network management). That is why there are file servers, communication servers, application servers, etc. Computers working as servers are usually very powerful. Servers usually represent high-efficiency computers, probably, with several parallel-working processors, with hard disks of a high capacity, with a high-speed network card (100 Mbit/s and more). Networks of the first type are cheaper. But they need more expensive computers. And networks of the second type need cheaper computers, which work as clients, and need rather expensive servers. But if you know that your network will grow, you don't want computers to depend on each other or you want to connect your network to WAN later then you need network of this type. WANs are all over the world networks. Well-known Internet is a representative of a global network. Initially Internet was a military idea to be developed. But now it can be used by anyone for information exchange, search of information, on-line conferences, and real-time data transmission from one point of the world to another one. It's hard to find a student who never used Internet to find some reports or just to have some fun.

***Learn the words below***

hit — 1) відповідь (при пошуку);

2) збіг (при пошуку)

mine data — отримувати інформацію, отримувати дані

web data mining — отримання даних із Інтернету

Web crawling — навігація, повільний рух по Всесвітній павутині

Web crawler — навігатор Всесвітньої павутини

indexing — індексна адресація

Web browser — web-браузер, система перегляду та спрямування інформації в Інтернеті

meta tag — мета-тег (тег верхнього рівня, який містить ідентифікаційні ознаки, але який не впливає на екранні відображення web-сторінок) і

index database — індексна база даних

query — запит (критерій пошуку об'єктів у базі даних)

source page — початкова сторінка

boolean operator — знак логічної операції, логічний оператор

proximity search — пошук із відстанню (вид пошуку, при якому користувач вказує, на якій відстані один від одного повинні розташовуватися ключові слова у документах)

venture — справа (особливо пов'язана зі значним ризиком)

revenue — дохід, доходи

ad(vertisement) — реклама; оголошення

advertise — рекламувати; давати оголошення

advertiser — рекламодавець

## TOPIC 4 GERUND

**Герундій (Gerund)** – це безособова форма дієслова з закінченням **-ing**, що поєднує у собі властивості **іменника** та **дієслова**. Герундій не виражає число, особу, способи та не вживається з **артиклями**. В українській мові форми герундія відповідають іменники, що утворилися від дієслів (reading - читання), форма інфінітиву, а також в деяких випадках дієприкметники та дієприслівники (being seen - помічений, walking - гуляючи). Герундій не слід плутати з **дієприкметником теперішнього часу**.

- **Герундій**

- *Eating hamburgers is not good for you.* – Вживання (в їжу) гамбургерів не добре для вас.

Частка **not**, а також **no** ставиться **перед герундієм**, до якого вона відноситься.

- *No playing and no talking after 10 p.m.!* – Ніяких ігор та розмов після 10-ї вечора!
- *I am thinking of not going to the party.* – Я думаю про те, щоб не піти на вечірку.

- **Форми герундія**

**Герундій** має дві форми в активному стані (Indefinite та Perfect Gerund Active) та дві форми пасивного стану (Indefinite та Perfect Gerund Passive).

| <b>Tense</b>                                       | <b>Active Voice</b>                                   | <b>Passive Voice</b>                                      |
|--|---|---|
| Indefinite (Simple) Gerund<br>неозначений герундій | <b>studying</b><br>дієслово з<br>закінченням -ing     | <b>being studied</b><br>being + 3 форма дієсл             |
| Perfect Gerund<br>доконаний герундій               | <b>having studied</b><br>having + 3 форма<br>дієслова | <b>having been studied</b><br>having been + 3 ф.<br>дієсл |

- **Активний стан**

- *Excuse me for disturbing you.* – Вибачте, що турбую вас. (дослівно: вибачте за моє втручання)
- *He won't mind your coming.* – Він не буде проти, якщо ти прийдеш. (дослівно: він не буде проти твого приходу)

- **Пасивний стан**

- *He entered the house without being seen.* – Він увійшов до будинку непоміченим.
- *Being eaten by aboriginals was not something Cook was dreaming about.* – Бути з'їденим аборигенами не було тим, про що мріяв Кук.

**Герундій активного стану** здебільшого вживається після дієслів **to want** (хотіти), **to need** (потребувати), **to deserve** (заслужувати), **to require** (вимагати) та прикметника **worth** (вартий), навіть якщо в реченні мається на увазі дія в пасивному стані.



- *His house wants repairing.* – Його дім слід відремонтувати. (дослівно: його дім потребує ремонтування.)
- *Our children deserve praising.* – Наші діти заслуговують на похвалу.
- *This play was worth seeing.* – Цей спектакль вартий того, щоб його подивитись.

### Функції герундія

**Герундій** в реченні може вживатися у якості підмета, додатку або ж іменної частини складного присудка, так як ця форма може виконувати ті ж самі функції, що й іменник.

- **Підмет** (головний член речення)
- *Smoking can destroy your health.* – Куріння може знищити твоє здоров'я.
- **Додаток** (другорядний член речення)
- *Jane likes taking people happy.* – Джейн любить дарувати людям щастя.
- **Іменна частина складного присудка** (головний член речення)
- *Our duty is taking care of those, who need help.* – Наш обов'язок – це турбота про тих, кому необхідна допомога.

### Використання герундія

**Герундій** завжди використовується після наступних простих та фразових дієслів.

*to admit* – визнавати, допускати

*to appreciate* – цінувати, бути вдячним

*to avoid* – уникати

*to burst out* – починати, спалахувати

*to consider* – розглядати, обговорювати, вважати

*to continue* – продовжувати

*to go on* – продовжувати, тривати

*to imagine* – уявляти

*to keep (on)* – продовжувати

*to leave off* – переставати, покидати

*to mind* – заперечувати, бути проти (в заперечних та питальних реченнях)

*to miss* – сумувати, скучити

*to deny* – заперечувати

*to excuse* – вибачати

*to fancy* – уявляти, думати, радіти

*to finish* – закінчувати, завершувати

*to forgive* – простити, вибачати

*to give up* – здаватися, полишати

*to go* – займатися (спортом)

*to postpone* – відкладати

*to practice* – практикувати

*to prevent* – запобігати, попереджати

*to put off* – відкладати

*to quit* – припиняти, полишити

*to save* – зберігати

*to suggest* – пропонувати

- *Fancy meeting you here!* – Не міг навіть уявити, що зустріну тебе тут!
- *Relax. Imagine lying on the beach and drinking cocktails.* – Розслабся. Уяви собі, що ти лежиш на пляжі та п'єш коктейлі.

**Герундій** вживається після дієслів, коли ці дієслова вказують на чийсь уподобання, часто після дієслів **to love** (любити), **to like** (подобатися), **to enjoy** (насолоджуватися), **to prefer** (віддавати перевагу), **to dislike** (не любити), **to hate** (ненавидіти)

- *She hates cooking.* – Вона ненавидить готувати.
- *I don't like writing letters.* – Я не люблю писати листи.

- *Kate enjoys watching football with her boyfriend.* – Кейт подобається дивитись футбол з її хлопцем.

**Герундій** використовується після дієслів **to spend** (витрачати), **to waste** (марнувати, марно витрачати), **to lose** (загубити), коли вони вживаються у значенні марної трати грошей, часу, здоров'я тощо.

- *She lost her health taking care of children.* – Вона згубила своє здоров'я, доглядаючи дітей.
  - *Matt spent a lot of money buying a computer he doesn't even use.* – Метт витратив багато грошей на комп'ютер, яким він навіть не користується.
- Форма **герундія** вживається з дієсловами **to hear** (чути), **to listen to** (слухати), **to notice** (помічати), **to see** (бачити), **to watch** (дивитись), **to feel** (відчувати), коли вони вказують на дію, що була не до кінця завершена.
- *I watched the actors rehearsing the play.* – Я дивився, як актори репетирували спектакль. (я бачив тільки частину репетиції)
  - *Ann heard some guys talking about her.* – Енн чула, як якісь хлопці розмовляли про неї. (вона чула тільки частину розмови)

**Герундій** використовується після дієслів та виразів з **прийменниками**, що зазначені нижче.

*cannot help* – бути не в змозі зупинитися

*cannot stand* – не переносити

*not to like the idea of* – не поділяти намір

*to accuse of* – звинувачувати

*to agree to* – погоджуватися

*to approve of* – схвалювати

*to be astonished at* – бути враженим, здивованим

*to be aware of* – знати, бути у курсі

*to be busy in* – бути зайнятим

*to be capable of* – бути спроможним

*to be displeased at* –

бути незадоволеним

*to be fond of* – любити

*to be guilty of* – бути винуватим

*to be indignant at* – обурюватися

*to be pleased at* – бути задоволеним

*to be proud of* – пишатися, гордитися

*to be sure of* – бути певним у

*to be surprised at* – бути здивованим

*to be used to* – бути звичним до

*to be worth* – бути гідним, достойним

*to complain of* – скаржитися, жалітися

*to depend on* – залежати

*to feel like* – хотіти

*to give up the idea of* – відмовитися від ідеї

*to have difficulty in* – мати труднощі з

*to have trouble* – мати труднощі з

*to insist on* – наполягати

*to look forward to* – чекати з

нетерпінням

*to look like* – схоже, що, збиратися

*to miss an opportunity of* – упустити можл

*to object to* – заперечувати, бути проти

*to persist in* – наполягати, наполегливо продовжувати

*to prevent from* – вберегти, зашкодити

*to rely on* – покластися, довіряти

*to speak of* – висловлюватися

*to succeed in* – мати успіх, досягти успіху

*to suspect of* – підозрювати

*to thank for* – бути вдячним, дякувати

*to think of* – думати, збиратися

- *It looks like raining.* – Схоже, (скоро) задощить.
- *I was thinking of inviting Paul to my party.* – Я думав про те, щоб запросити Пола на свою вечірку.

**Герундій** також вживається в певних сталих виразах.

- *it is no use ...* – немає сенсу, необхідності в ...
- *it is (no) good ...* – недобре (добре) ...
- *what is the use of ..?* – яка необхідність в ..?
- *there is no point in ...* – немає сенсу в ...
- *in addition to ...* – на додаток (до) ...

### Exercise

Give the correct gerund form of these verbs (do not use 'to' and do not capitalize the verbs)

| Verbs  | Gerunds                |
|--|------------------------|
| 1. She likes (paint)                           | <input type="text"/> . |
| 2. I can't bear (listen) to loud music         | <input type="text"/> . |
| 3. He enjoys (play) tennis                     | <input type="text"/> . |
| 4. Leila dreams of (set) up her own business   | <input type="text"/> . |
| 5. He is interested in (emigrate) to Canada    | <input type="text"/> . |
| 6. Are you good at (dance)                     | <input type="text"/> . |
| 7. She is crazy about (read) romantic poems    | <input type="text"/> . |
| 8. I can't help (laugh) when I watch Mr Bean   | <input type="text"/> . |
| 9. I can't imagine (be) anywhere else but here | <input type="text"/> . |
| 10. You should give up (smoke)                 | <input type="text"/> . |

## TOPIC 5 COMPUTER SOFTWARE

### Active vocabulary

control — керувати

specify — задавати, визначати

meaningful — значущий

ever-larger — постійно зростаючий

arise (arose, arisen) — з'явитися, постати

perceive — 1) розуміти, усвідомлювати; 2) сприймати

care — ретельність, уважність, обережність

maintainability — експлуатаційна надійність;

зручність експлуатації

application software — прикладне програмне забезпечення

system software — системне програмне забезпечення

one-off — одноразовий, (тут спеціальний)

alternatively — або

software house — фірма з розробки та реалізації програмних засобів

payroll — платіжна відомість (тут назва відповідної прикладної програми)

pay slip — бланк платіжної відомості

word processing — оброблення (редагування) тексту

ultimately — зрештою, в кінцевому підсумку

machine language — машинна мова

assembly language — мова асемблера

closeness — близькість, наближеність

translator program — програма-транслятор

run — виконувати (програму)

debug — налагоджувати (програму)

opcode — код операції

language processor — транслятор або інтерпретатор (програма, яка трансліює або інтерпретує програму, написану мовою програмування)

handle — оперувати, працювати з чимось

therefore — тому, внаслідок чого

one-to-one — однозначний

trend — тенденція

remote — 1) віддалений; 2) несхожий

**APPLICATION SOFTWARE.** A digital computer cannot work without software, i.e. a program which controls different hardware components of the computer in a sequence specified by the user. By simply changing the program, the computer can be made to work in a different manner. In other words, software provides a digital computer with general-purpose computing capability. One should keep in mind that software and hardware are complementary to each other. Both have to work together to produce a meaningful result. The software, that is becoming an ever-larger part of the computer system, is growing more and more complicated, requiring teams of programmers and years of effort to develop. As a consequence, a new subdiscipline, software engineering, has arisen. The development of a large piece of software is perceived as an engineering task, to be approached with the same care as the construction of a skyscraper, for example, and with the same attention to cost, reliability, and maintainability of the final product. Computer software is usually classified into two broad categories: application software and system software. Application software is a set of programs to carry out operations for a specific application. Application software may consist of one-off programs written by the user or alternatively of packages written by commercial software houses and bought by the user to accomplish specific tasks. For example, payroll is an application software for an organization to produce pay slips as an output. Application software is useful for word processing, billing system, accounting, producing statistical reports, analysis of numerous data in research, weather forecasting, etc. Such systems as MS Word, Lotus 1-2-3, dBASE III Plus are all application softwares. Another example of application software is programming languages. Programming languages are the languages in which a programmer writes the instructions that the computer will ultimately execute. There are two major types of programming languages. These are low-level languages and high-level languages. Low-level languages are further divided into Machine language and Assembly language. The term low-level means closeness to the way in which the machine has been built. Low-level languages are machine oriented and require extensive knowledge of computer hardware and its configuration. Machine language is the only language that is directly understood by the computer. It does not need any translator program. Machine language instructions, which are also called machine code, are written in binary form — i.e., strings of 1s and 0s (e.g. 01011010) — because electronic circuits inside computers, including memories, work only with the signals 0 and 1. When this sequence of codes is fed to the computer, it recognizes the codes and converts it into electrical signals needed to run it. For example, a program instruction may look like this: 10110010110100. It is not an easy language for the user to learn. It is also difficult to debug programs written in this language. Thus, programmers use

languages that are readable for them but not for computers. For example, the mnemonic opcode LOAD in a human readable language is far easier to understand than 0101 in machine code. Such a human readable language must be translated into machine language by special purpose programs called language processors. Machine language is considered to be the first generation language. Assembly language was the first step to improve the programming structure. You know that computer can handle numbers and letters. Therefore some combinations of letters can be used to substitute for numbers of machine codes. The set of symbols and letters forms the assembly language and a translator program is required to translate this language to machine language. This program is called Assembler. The symbolic programming of this language is easier to understand and saves a lot of time and effort of the programmer. It is also easier to correct errors and modify program instructions. Assembly language has the same efficiency of execution as the machine level language because there is a one-to-one translator between an assembly language program and its corresponding machine language program. Assembly language is considered to be a second-generation language. So, you now understand that programming in either assembly language or machine language is not an easy business requiring deep knowledge of computer hardware. To make the programmer's work easier more convenient, high-level languages were soon (beginning in the mid-1950s) invented. High-level languages are problem oriented because the instructions are suitable for solving a particular problem. For example the language FORTRAN (FORmula TRANslation) was invented for solving mathematical problems. It was originally much like assembly language; however it allowed programmers to write algebraic expressions instead of coded instructions for arithmetic operations. COBOL (COMmon Business Oriented Language) was developed to handle records and files and the operations necessary for simple business application. The trend since then has been toward developing increasingly abstract languages, allowing the programmer to think and communicate with the machine at a level ever more remote from machine code.

***Exercise 1 Find in text the English for:*** працювати з числами і буквами; обробка текстів; стає дедалі складнішим; по-іншому; як наслідок; конкретна задача; мова 117 низького/високого рівня; програмне та апаратне забезпечення доповнюють одне одне; дедалі більше несхожий на машинний код; у визначеній користувачем послідовності; схожість з тим, як.

***Exercise 2 Translate into English paying special attention to the italicized words.*** 1. Персональні комп'ютери та ноутбуки, що зараз вже стали для нас звичними, дуже несхожі на обчислювальні машини 1960–1970-х років. 2.

Цифрові комп'ютери працюють із числами і буквами, які представлені двійковим кодом. 3. Гармонійний розвиток особистості досягається поєднанням розумової і фізичної праці, які доповнюють одна одну. 4. Мініатюризація елементної бази електроніки призвела до створення інтегрованих мікросхем, і, як наслідок, наприкінці минулого століття стали з'являтися комп'ютери, які за своїми розмірами ставали дедалі меншими. 5. Для полегшення праці програміста із середини 1950-х років розробляються мови, які дедалі більше схожі на природну мову. 6. Схожість із тим, як було скоєно інший злочин, допомогла поліції швидко встановити особу злочинця. 7. Мови програмування, які схожі на людську мову і несхожі на машинний код, і, як наслідок, є зручнішими для людини, називаються мовами високого рівня. 8. У сучасний футбол, який дуже не схожий на той, що був сто років тому, грають зовсім по-іншому. 9. Розвиток інформаційних технологій та вдосконалення апаратних і програмних засобів привело до того, що сучасні комп'ютери, з одного боку, стають дедалі складнішими, а з іншого — дедалі зручнішими в користуванні. 10. Прикладна програма спрямована на розв'язання конкретної задачі шляхом виконання операцій у визначеній алгоритмом послідовності

## ТОПІС 6

### INFINITIVE

**Інфінітив** – це безособова або неозначена форма дієслова, що відповідає на питання «**що робити?**», «**що зробити?**». Інфінітив називає тільки дію і не виражає категорії особи, числа або способу. Відмінною особливістю форми інфінітиву є частка **to**, однак в деяких випадках вона може не вживатися (так званий **голий інфінітив**).

- *I would love to come but I'm busy now.* – Я б хотів прийти, та я зараз зайнятий.

В реченнях заперечна частка **not** ставиться перед **інфінітивом**, до якого вона належить.

- *I will try not to be late.* – Я постараюся не запізнитися.
- *I asked you not to scream.* – Я попросив тебе не кричати.

#### Форми інфінітиву

**Інфінітив** в англійській мові може вживатися в чотирьох особливих формах, що відповідають чотирьом видам дієслів. Інфінітиви **перехідних дієслів** можуть мати форми активного та пасивного стану.

| <b>Tense</b>   | <b>Active Voice</b>  | <b>Passive Voice</b>   |
|--|--|--|
| Indefinite (Simple) Infinitive<br>простий інфінітив                        | <b>to eat (їсти)</b><br>to + перша форма дієслова                        | <b>to be eaten (з'їстися)</b><br>to be + третя форма дієслова    |
| Continuous (Progressive) Infinitive<br>тривалий інфінітив                  | <b>to be eating</b><br>to be + дієслово з закінченням -ing               | ---<br>не вживається   |
| Perfect Infinitive<br>доконаний інфінітив                                  | <b>to have eaten</b><br>to have + третя форма дієслова                   | <b>to have been eaten</b><br>to have been + третя форма дієслова |
| Perfect Continuous (Progressive) Infinitive<br>доконано-тривалий інфінітив | <b>to have been eating</b><br>to have been + дієслово з закінченням -ing | ---<br>не вживається   |



*to agree* – погоджуватися  
*to aim* – мати наміром,  
намагатися, цілитися  
*to appear* – виявлятися,  
виявитися  
*to arrange* – домовлятися,  
урегулювати  
*to ask* – просити  
*to attempt* – намагатися  
*to be able* – бути в змозі,  
могти  
*to beg* – просити, благати  
*to begin* – починати  
*to care* – хотіти, мати  
бажання  
*to choose* – вибирати  
*to claim* – заявляти,  
стверджувати  
*to dare* – наважуватися,  
сміти, мати сміливість  
*to decide* – вирішити,  
прийняти рішення  
*to demand* – вимагати,  
просити

*to deserve* – заслуговувати  
*to dread* – боятися, жахатися  
*to expect* – очікувати  
*to fail* – зазнавати невдачі,  
провалюватися, не зробити  
*to get* – отримати дозвіл,  
мати потребу  
*to happen* – виявитися,  
статися  
*to hesitate* – вагатися, не  
наважуватися  
*to hope* – сподіватися  
*to hurry* – поспішати  
*to intend* – мати намір,  
збиратися  
*to learn* – вчитися  
*to manage* – впоратися,  
зуміти  
*to need* – потребувати, мати  
потребу  
*to neglect* – забувати, не  
зробити, нехтувати  
*to plan* – планувати

*to prepare* – приготуватися  
*to pretend* – робити вигляд,  
прикидатися  
*to proceed* – перейти до,  
продовжувати  
*to promise* – обіцяти  
*to refuse* – відмовлятися  
*to seem* – здаватися, мати  
вигляд  
*to strive* – намагатися,  
докладати зусилля  
*to swear* – клястися  
*to tend* – мати тенденцію,  
схилитись, прагнути  
*to threaten* – погрожувати,  
залякувати  
*to vow* – давати обітницю,  
клястися  
*to wait* – чекати  
*to want* – хотіти  
*to wish* – хотіти, бажати

- *My children refuse to drink milk.* – Мої діти відмовляються пити молоко.
- *Kate wanted to enter her flat but she forgot her keys.* – Кейт хотіла увійти до своєї кімнати, однак забула ключі.
- *I agree to come with you if you promise to take me back home after.* – Я згодна піти з тобою, якщо ти обіцяєш після відвезти мене додому.

**Інфінітив з to** використовується після дієслів **to ask** (спросити), **to learn** (навчитися, дізнатися), **to explain** (пояснювати), **to decide** (вирішити), **to find out** (дізнатися), **to want to know** (хотіти дізнатися), коли після них стоїть питальне слово.

- *I learned how to swim when I was 5.* – Я навчилася плавати, коли мені було 5 років.
- *We should decide now what to do next.* – Ми повинні вирішити, що нам робити далі.

## Gerunds/Infinitives

### A. Circle the correct words to complete these ideas from a book about memory.

1. Get into the habit of **repeating/ to repeat** things aloud.
2. Never rely on someone else's memory. Learn **trusting / to trust** your own.
3. It's easy **forgetting / to forget** what you don't want **remembering / to remember**.
4. Study immediately before **going / to go** to sleep. You'll remember a lot more.
5. Our memories are filled with things we never meant **remembering / to remember**.
6. Make it a habit to look back at your car every time you get out and you'll never forget **turning off / to turn off** your headlights.
7. **Playing / To play** games is a fun way of **improving / to improve** your memory skills.

### B. Complete the sentences with prepositions followed by gerunds.

- 1 After a long time we eventually succeeded.....a flat. (find)
- 2 I've been thinking.....for a new job. (look)
- 3 His parents didn't approve.....him .....out so late. (stay)
- 4 I wonder what prevented him .....to the party. (come)
- 5 I'm getting hungry. I'm looking forward .....dinner. (have)
- 6 I don't feel.....today. (study)
- 7 Forgive me .....you but I must ask you a question. (interrupt)
- 8 The arrested man was suspected .....into a house. (break)
- 9 Have you ever thought.....married? (get)
- 10 I've always dreamed.....on a small island in the Pacific. (live)
- 11 The cold water didn't stop her.....a swim. (have)
- 12 Have you ever been accused .....a crime? (commit)
- 13 She apologised .....so rude to me. (be)
- 14 We have decided .....a new car. (buy)
- 15 Jack insisted .....by himself. (go)

### C. Transform the sentences. Begin each of your sentences in the way shown.

**Example:** 'It was nice of you to help me. Thanks very much.' (George said to you)

George thanked..... **me.... for...helping.... him**.....

- 1 'I'll drive you to the station. I insist.' (Tom said to Ann)  
Tom insisted.....
- 2 'I hear you passed your examinations. Congratulations!' (Jim said to you)  
Jim congratulated .....
- 3 'It was nice of you to visit me. Thank you.' (Mrs Dent said to Sue)  
Mrs Dent thanked.....
- 4 'Don't stay at the hotel near the airport.' (I said to Jack)  
I warned .....
- 5 'I'm sorry I didn't phone you earlier.' (Margaret said to you)  
Margaret apologised.....not .....
- 6 'You didn't pay attention to what I said.' (The teacher said to the boy)  
The teacher accused .....

### D. Put the verb into the gerund or infinitive. (Sometimes either form is possible.)

- 1 A: You lent me some money a few months ago.  
B: Did I? That's strange. I don't remember.....(lend) you any money.
- 2 We tried.....(put) the fire out but we were unsuccessful. We had to call the fire-brigade.
- 3 When you see Tom, remember.....(give) him my regards, won't you?
- 4 What do you intend.....(do) about this problem?
- 5 Someone must have taken my bag. I clearly remember.....(leave) it by the window and now it has gone.
- 6 When she saw what had happened, she began.....(laugh) loudly.
- 7 Sue needed some money. She tried.....(ask) Gerry but he couldn't help her.
- 8 He tried.....(reach) the shelf but he wasn't tall enough.
- 9 'Did you remember.....(phone) Ann?' 'Oh no, I completely forgot.'
- 10 I asked them to be quiet but they continued.....(make) a lot of noise.
- 11 Have you finished.....(fill in) that form?
- 12 They threatened.....(deport) her on the next flight.
- 13 He avoided.....(cross) the bridge by swimming the river.
- 14 Would you mind.....(show) me your passport, sir?

## TOPIC 7

### COMPUTER ARCHITECTURE

#### Active Vocabulary

blueprint — проект

design implementation — конструкційна реалізація

constituent part — складова частина

interoperate — взаємодіяти

interconnects — схема з'єднань,

розводка off-load — ненавантажений, непрацюючий

register-transfer level — рівень регістрових передач

latch — защіпка (цифрова логічна мікросхема, що використовується для проміжного збереження одного або більше бітів, напр. address latch — защіпка адреси, level latch — защіпка рівня сигналу)

chip floorplan — базовий план кристала

proprietary — власний, розроблений всередині фірми для власних потреб

communication — тут повідомлення

luxuriously — пишно, розкішно

embellish — прикрашати (тж. перен.)

gain currency — набувати поширення

#### NOTION OF COMPUTER ARCHITECTURE

In computer science and computer engineering, computer architecture or digital computer organization is the conceptual design and fundamental operational structure of a computer system. It forms a blueprint and functional description of requirements and design implementations for the various parts of a computer, focusing largely on the way by which the central processing unit (CPU) performs internally and accesses addresses in memory. It may also be defined as the science and art of selecting and interconnecting hardware components to create computers that meet functional, performance and cost goals. Computer architecture comprises at least three main subcategories: Instruction set architecture, or ISA. is the abstract image of a computing system that is seen by a machine language (or assembly language) programmer, including the instruction set, word size, memory address modes, processor registers, and address and data formats. Microarchitecture, also known as Computer organization is a lower level, more concrete and detailed, description of the system that involves how the constituent parts of the system are interconnected and how they interoperate in order to implement the ISA. The size of a computer's cache for instance, is an organizational issue that generally has nothing to do with the ISA. System Design which includes all of the other hardware components within a computing system such as: – system interconnects

such as computer buses and switches; – memory controllers and hierarchies; – CPU off-load mechanisms such as direct memory access (DMA). Issues like multiprocessing. Once both ISA and microarchitecture have been specified, the actual device needs to be designed into hardware. This design process is called the implementation. Implementation is usually not considered architectural definition, but rather hardware design engineering. Implementation can be further broken down into three (not fully distinct) pieces: Logic Implementation — design of blocks defined in the microarchitecture at (primarily) the register-transfer and gate levels. Circuit Implementation — transistor-level design of basic elements (gates, multiplexers, latches etc.) as well as of some larger blocks (ALUs, caches etc.) that may be implemented at this level, or even (partly) at the physical level, for performance reasons. Physical Implementation — physical circuits are drawn out, the different circuit components are placed in a chip floorplan or on a board and the wires connecting them are routed. For CPUs, the entire implementation process is often called CPU design. More specific usages of the term include more general wider-scale hardware architectures, such as cluster computing and Non-Uniform Memory Access (NUMA) architectures. History. The term “architecture” in computer literature can be traced to the work of Lyle R. Johnson, Muhammad Usman Khan and Frederick P. Brooks. Jr., members in 1959 of the Machine Organization department in IBM’s main research center. Johnson had the opportunity to write a proprietary research communication about Stretch, an IBM-developed supercomputer for Los Alamos Scientific Laboratory. In attempting to characterize his chosen level of detail for discussing the luxuriously embellished computer, he noted that his description of formats, instruction types, hardware parameters, and speed enhancements was at the level of “system architecture” a term that seemed more useful than “machine organization”. Subsequently, Brooks, one of the Stretch designers, started Chapter 2 of a book (Planning a Computer System: Project Stretch, ed. W. Buchholz, 1962) by writing, “Computer architecture, like other architecture, is the art of determining the needs of the user of a structure and then designing to meet those needs as effectively as possible within economic and technological constraints”. Brooks went on to play a major role in the development of the IBM System/360 line of computers, where “architecture” gained currency as a noun with the definition “what the user needs to know”. Later the computer world would employ the term in many less-explicit ways.

***Exercise 1 Translate into English paying attention to the italicized words.*** 1.

Бажано виконати це завдання якомога швидше. 2. Для роботи з відеофайлами комп’ютер повинен мати потужну відеокарту з міркувань продуктивності. 3.

Сучасні комп'ютери здатні задовольнити найрізноманітніші потреби. 4. Для того, щоб вирішувати обчислювальні завдання якомога ефективніше, комп'ютер повинен мати потужний процесор. 5. З міркувань безпеки всі ремонтні роботи слід проводити на знеструмленому комп'ютері. 6. Це тебе не обходить. 7. Перед нами стояло завдання, яке треба було виконати якомога ефективніше за існуючих фінансових і технічних обмежень. 8. Завдяки наполегливій праці цей молодий програміст став відігравати важливу роль у дослідницькій групі професора Янга. 9. Він далі сказав, що мусив так вчинити, аби виконати поставлене завдання за існуючих часових і фінансових обмежень. 10. Вираз “computer architecture” з'явився на межі 1950–1960-х років і згодом набув поширення як термін, що позначає організаційну структуру комп'ютера або обчислювальної системи.

**TYPES OF COMPUTER ARCHITECTURES** There are many types of computer architectures: – Quantum computer vs Chemical computer; – Scalar processor vs Vector processor; – Non-Uniform Memory Access (NUMA) computers; – Register machine vs Stack machine; – Harvard architecture vs von Neumann architecture; – Cellular architecture. The quantum computer architecture holds the most promise to revolutionize computing. Sub-definitions. Some practitioners of computer architecture at companies such as Intel and AMD use more fine distinctions: Macroarchitecture — architectural layers that are more abstract than microarchitecture, e.g. ISA. Instruction Set Architecture (ISA) — as defined in the previous texts. Assembly ISA — a smart assembler may convert an abstract assembly language common to a group of machines into slightly different machine language for different implementations. Programmer Visible Macroarchitecture — higher level language tools such as compilers may define a consistent interface or contract to programmers using them, abstracting differences between underlying ISA, UISA and microarchitectures. E.g. the C, C++ or .java standards define different Programmer Visible Macroarchitecture — although in practice the C microarchitecture for a particular computer includes: UISA (Microcode Instruction Set Architecture) — a family of machines with different hardware level microarchitectures may share a common microcode architecture, and hence a UISA. Pin Architecture — the set of functions that a microprocessor is expected to provide, from the point of view of a hardware platform. Pin architecture functions are more flexible than ISA functions — external hardware can adapt to changing encodings, or changing from a pin to a message — but the functions are expected to be provided in successive implementations even if the manner of encoding them changes. Computer architecture: the definition. The coordination of abstract levels of a processor under changing forces, involving

design, measurement and evaluation. It also includes the overall fundamental working principle of the internal logical structure of a computer system. It can also be defined as the design of the task-performing part of computers, i.e. how various gates and transistors are interconnected and are caused to function per the instructions given by an assembly language programmer. Instruction set architecture. The ISA is the interface between the software and hardware. It is the set of instructions that bridges the gap between high level languages and the hardware. For a processor to understand a command, it should be in binary and not in High Level Language. The ISA encodes these values. The ISA also defines the items in the computer that are available to a programmer. For example, it defines data types, registers, addressing modes, memory organization etc.

### ***Learn the words below***

supervisory software — комплект керівних програм

clock speed = clock rate — тактова частота

metric — показник

misleading — оманливий, дезорієнтувальний

bus speed — частота шини, швидкість шини latency — 1) час (період)

чекання; 2) затримка, запізнення interrupt latency — затримка (обробка)]

переривання throughput — 1) продуктивність (обчислювальної системи); 2)

пропускна здатність (каналу зв'язку)

design choice — конструкційне рішення

pipelining a processor — конвеєризація (реалізація конвеєрної структури)

процесора

anti-lock brake — гальмо, що не заклинює CPU (I/O, memory)

bound — обмежений швидкістю процесора (можливостями пристроїв

введення-виведення, об'ємом пам'яті)

benchmarking — 1) проведення оцінювальних випробувань; 2) тип виконання

програми, що дозволяє простежити кількість циклів ЦП, що витрачається на задану область коду

show strengths — виявляти переваги

power efficiency — енергоефективність

trade sth for sth — обмінювати що-небудь на щось інше

power density — щільність енергії

## TOPIC 8 CONDITIONALS

English Grammar

# Conditionals

Woodward ENGLISH ENGLISH

**CONDITION** + **RESULT**

**ZERO conditional** If you stand in the rain, you get wet.  
If you heat ice, it melts.  
**PRESENT SIMPLE** + **PRESENT SIMPLE**  
*USES: Facts which are generally true or scientific facts  
The condition always has the same result*

**FIRST conditional** If it rains, we will cancel the trip.  
If you study, you will pass the exam.  
**PRESENT SIMPLE** + **WILL / WON'T + VERB**  
*USES: A possible situation in the future  
Predicting a likely result in the future (if the condition happens)*

**SECOND conditional** If I won the lottery, I would travel a lot.  
If they sold their house, they would be rich.  
**PAST SIMPLE** + **WOULD + VERB**  
*USES: Hypothetical or unlikely situations  
Unreal or improbable situation now or in the future*

**THIRD conditional** If you had studied, you would have passed the exam.  
If I hadn't been sick, I would have gone to your party.  
**PAST PERFECT** + **WOULD HAVE + PAST PARTICIPLE**  
*USES: The person is imagining a different past  
Imaginary situation that did not happen*

[www.grammar.cl](http://www.grammar.cl) [www.woodwardenglish.com](http://www.woodwardenglish.com) [www.vocabulary.cl](http://www.vocabulary.cl)

Рис.2

### Exercises on Conditional Sentences Type 1

Complete the Conditional Sentences (Type I) by putting the verbs into the correct form.

1. If you (send) \_\_\_\_\_ this letter now, she (receive) \_\_\_\_\_ it tomorrow.
2. If I (do) \_\_\_\_\_ this test, I (improve) \_\_\_\_\_ my English.
3. If I (find) \_\_\_\_\_ your ring, I (give) \_\_\_\_\_ it back to you.
4. Peggy (go) \_\_\_\_\_ shopping if she (have) \_\_\_\_\_ time in the afternoon.
5. Simon (go) \_\_\_\_\_ to London next week if he (get) \_\_\_\_\_ a cheap flight.
6. If her boyfriend (phone / not) \_\_\_\_\_ today, she (leave) \_\_\_\_\_ him.
7. If they (study / not) \_\_\_\_\_ harder, they (pass / not) \_\_\_\_\_ the exam.
8. If it (rain) \_\_\_\_\_ tomorrow, I (have to / not) \_\_\_\_\_ water the plants.
9. You (be able / not) \_\_\_\_\_ to sleep if you (watch) \_\_\_\_\_ this scary film.
10. Susan (can / move / not) \_\_\_\_\_ into the new house if it (be / not) \_\_\_\_\_ ready on time.

### Exercise on Conditional Sentences Type 2

Complete the Conditional Sentences (Type II) by putting the verbs into the correct form. Use conditional I with **would** in the main clause.

1. If we (have) \_\_\_\_\_ a yacht, we (sail) \_\_\_\_\_ the seven seas.
2. If he (have) \_\_\_\_\_ more time, he (learn) \_\_\_\_\_ karate.
3. If they (tell) \_\_\_\_\_ their father, he (be) \_\_\_\_\_ very angry.
4. She (spend) \_\_\_\_\_ a year in the USA if it (be) \_\_\_\_\_ easier to get a green card.
5. If I (live) \_\_\_\_\_ on a lonely island, I (run) \_\_\_\_\_ around naked all day.
6. We (help) \_\_\_\_\_ you if we (know) \_\_\_\_\_ how.
7. My brother (buy) \_\_\_\_\_ a sports car if he (have) \_\_\_\_\_ the money.
8. If I (feel) \_\_\_\_\_ better, I (go) \_\_\_\_\_ to the cinema with you.
9. If you (go) \_\_\_\_\_ by bike more often, you (be / not) \_\_\_\_\_ so flabby.
10. She (not / talk) \_\_\_\_\_ to you if she (be) \_\_\_\_\_ mad at you.

### Exercise on Conditional Sentences Type 3

Complete the Conditional Sentences (Type III) by putting the verbs into the correct form.

1. If you (study) \_\_\_\_\_ for the test, you (pass) \_\_\_\_\_ it.
2. If you (ask) \_\_\_\_\_ me, I (help) \_\_\_\_\_ you.
3. If we (go) to the cinema, we (see) my friend Jacob.
4. If you (speak) English, she (understand).
5. If they (listen) \_\_\_\_\_ to me, we (be) \_\_\_\_\_ home earlier.
6. I (write) \_\_\_\_\_ you a postcard if I (have) \_\_\_\_\_ your address.
7. If I (not / break) \_\_\_\_\_ my leg, I (take part) \_\_\_\_\_ in the contest.
8. If it (not / start) \_\_\_\_\_ to rain, we (walk) \_\_\_\_\_ to the museum.
9. We (swim) \_\_\_\_\_ in the sea if there (not / be) \_\_\_\_\_ so many sharks there.
10. If she (take) \_\_\_\_\_ the bus, she (not / arrive) \_\_\_\_\_ on time.



## TOPIC 9

### STORAGE AND RETRIEVAL OF DATA

Associated with the central processing unit is the storage unit, or memory, where results or other data are stored for periods of time ranging from a small fraction of a second to days or weeks before being retrieved for further processing. Once made up of vacuum tubes and later of small doughnut-shaped ferromagnetic cores strung on a wire matrix, main storage now consists of integrated circuits, each of which contains thousands of semiconductor devices. Where each vacuum tube or core represented one bit and the total memory of the computer was measured in thousands of bytes (or kilobytes, KB), each semiconductor device now represents millions of bytes (or megabytes, MB) and the total memory of mainframe computers is measured in billions of bytes (or gigabytes, GB). Random-access memory (RAM), which both can be read from and written to, is lost each time the computer is turned off. Read-only memory (ROM), which cannot be written to, maintains its content at all times and is used to store the computer's control information. Programs and data that are not currently being used in main storage can be saved on auxiliary storage, or external storage. Although punched paper tape and punched cards once served this purpose, the major materials used today are magnetic tape and magnetic disks, which can be read from and written to, and two types of optical disks, the compact disc (CD) and its successor the digital versatile disc (DVD). DVD is an improved optical storage technology capable of storing vastly greater amounts of data than the CD technology. CD-Read-Only Memory (CD-ROM) and DVD-Read-Only Memory (DVD-ROM) disks can only be read — the disks are impressed with data at the factory but once written cannot be erased and rewritten with new data. The latter part of the 1990s saw the introduction of new optical storage technologies: CD-Recordable (CD-R) and DVD-Recordable (DVD-R), optical disks that can be written to by the computer to create a CD-ROM or DVD-ROM, but can be written to only once; and CD-ReWritable (CD-RW), DVD-ReWritable (DVD-RW and DVD+RW), and DVD-Random Access Memory (DVD-RAM), disks that can be written to multiple times. Magnetic and Optical Storage. When compared to semiconductor memory, magnetic and optical storage is less expensive, is not volatile (i.e., data is not lost when the power to the computer is shut off), and provides a convenient way to transfer data from one computer to another. Thus operating instructions or data output from one computer can be stored away from the computer and then retrieved either by the same computer or another. In a system using magnetic tape the information is stored by a specially designed tape recorder somewhat similar to

one used for recording sound. In magnetic and optical disk systems the principle is the same except that the magnetic or optical medium lies in a path, or track, on the surface of a disk. The disk drive also contains a motor to spin the disk 190 and a magnetic or optical head or heads to read and write the data to the disk. Drives take several forms, the most significant difference being whether the disk can be removed from the drive assembly. Removable magnetic disks are most commonly made of mylar enclosed in a paper or plastic holder. These floppy disks have varying capacities, with very high-density disks holding 250 MB — more than enough to contain a dozen books the size of Tolstoy's *Anna Karenina*. Compact discs can hold many hundreds of megabytes, and are used, for example, to store the information contained in an entire multivolume encyclopedia or a set of reference works, and DVD disks can hold ten times as much as that. Non-removable disks are made of metal and arranged in spaced layers. They can hold more data and can read and write data much faster than floppies. Data entered into the computer and the processed data are made available via input/output devices. All auxiliary storage devices are used as input/output devices. For many years, the most popular input/output medium was the punched card. Although this is still used, the most popular input device is now the computer terminal and the most popular output device is the high-speed printer. Human beings can directly communicate with the computer through computer terminals, entering instructions and data by means of keyboards much like the ones on typewriters, by using a pointing device such as a mouse, trackball, or touchpad, or by speaking into a microphone that is connected to computer running voice-recognition software. Responses may be displayed on a cathode-ray tube, liquidcrystal display, or printer. The CPU, main storage, auxiliary storage, and input/output devices collectively make up a system. Sharing the Computer's Resources. Generally, the slowest operations that a computer must perform are those of transferring data, particularly when data is received from or delivered to a human being. The computer's central processor is idle for much of this period, and so two similar techniques are used to use its power more fully. Time sharing, used on large computers, allows several users at different terminals to use a single computer at the same time. The computer performs part of a task for one user, then suspends that task to do part of another for another user, and so on. Each user only has the computer's use for a fraction of the time, but the task switching is so rapid that most users are not aware of it. Most of the tens of millions of computers in the world are standalone, single-user devices known variously as personal computers or workstations. For them, multitasking involves the same type of switching, but for a single user. This permits a user, for example, to have one file printed and another sorted while editing a third in a wordprocessing session. Such personal computers can also be linked together in a

network, where each computer is connected to others, usually by wires or coaxial cables, permitting all to share resources such as printers, modems, and hard-disk storage devices.

**Exercise 1 Find in text the English for:** феромагнітне ядро; індивідуальний пристрій; перфокарта; цифровий універсальний диск; вузол дисководу; електронна лампа; високошвидкісний принтер; запровадження нових технологій; переносити інформацію з комп'ютера на комп'ютер; інтегрована схема; диски, на яких записано нову інформацію; подальша обробка; магнітна стрічка; швидке перемикання задач; енергозалежна (енергонезалежна) пам'ять; пластмасова касета; розподіляти ресурси; зовнішня пам'ять; спеціально призначений; диски з нанесеною на заводі інформацією; знімний (незнімний) диск; диск високої щільності; програмні засоби розпізнавання голосу; розподіл часу; живлення комп'ютера; тимчасово призупиняти задачу; зберігати інформацію поза комп'ютером; автономний пристрій; пристрій керування курсором.

**Exercise 2 Say whether the following sentences are true or false. Correct the false ones.** 1. In the past main storage was built primarily on the basis of optical and magnetic-optical media. 2. Random-access memory, which both can be read from and written to, is non-volatile. 3. Read-only memory is used to store the computer's control information. 4. We may regard the digital versatile disk as a predecessor of the compact disk. 5. CD-RW, DVD-RW, and DVD-RAM disks are very much like CD-R and DVD-R disks, but the latter can be written to multiple times. 6. Magnetic and optical storage provides a convenient way to transfer data from one computer to another. 7. The disk drive also contains a motor to spin the disk and a magnetic or optical head (or heads) to read and write data to the disk. 8. DVD disks can hold ten thousand times as much as compact disks. 9. Non-removable disks are made of metal and arranged in paths, or tracks. 10. All auxiliary storage devices are used as input/output devices.

**Exercise 3 Complete the sentences translating their Ukrainian parts into English.** 1. (Із центральним процесором пов'язаний пристрій пам'яті), where results or other data are stored for periods of time ranging from a small fraction of a second to days or weeks before being retrieved for further processing. 2. (Постійний запам'ятовувальний пристрій, у який не можна записувати, зберігає свій зміст) and is used to store the computer's control information. 3. The major materials used for external storage today are magnetic tape and magnetic disks, (з яких можна зчитувати і на які можна записувати), and two types of optical disks. 4. (Інформація наноситься на CD-ROM та DVD-ROM диски на заводі-виробнику) but once written, the disks cannot be erased and rewritten with

new data. 5. Operating instructions or data output from one computer (можуть зберігатися поза цим комп'ютером і згодом бути вибрані як цим самим комп'ютером, так і іншим). 6. In magnetic and optical disk systems the principle is the same (за винятком того, що магнітний або оптичний носій знаходиться на доріжці на поверхні диска). 7. (Дисководи бувають різними за формою), the most significant difference being whether the disk can be removed (з вузла дисководу). 8. Data entered into the computer and the processed data are made available (за допомогою пристроїв введення-виведення). 9. The CPU, main storage, auxiliary storage, and input/output devices (разом утворюють систему). 10. Each user has the computer's use for only a fraction of the time, (але комутація завдань є настільки швидкою, що більшість користувачів цього не усвідомлюють).

**Exercise 4 Translate the sentences.** 1. Нині оперативна пам'ять складається з інтегрованих схем, кожна з яких містить тисячі напівпровідникових приладів. 2. Кожний напівпровідниковий прилад містить мегабайти інформації, тоді як одна електронна лампа містила один біт. 3. Програми і дані, які в даний час в оперативній пам'яті не використовуються, можуть зберігатися у зовнішній пам'яті. 4. DVD становить собою вдосконалену оптичну технологію пам'яті, здатну зберігати незрівнянно більшу кількість інформації, ніж CD-технологія. 5. У другій половині 1990-х років були запроваджені нові оптичні технології пам'яті, які уможливили запис і перезапис інформації на оптичні диски за допомогою комп'ютера. 6. Порівняно з напівпровідниковою пам'яттю магнітооптична пам'ять дешевша та енергонезалежна (тобто інформація не зникає, коли живлення комп'ютера вимикають). 7. У системі, де використовується магнітна стрічка, інформація запам'ятовується спеціально призначеним пристроєм магнітного запису, дещо схожим на пристрій для записування звуку. 8. Компакт-диски вміщують сотні мегабайт і можуть використовуватися, наприклад, для збереження інформації, яка міститься в багатотомній енциклопедії. 9. Протягом багатьох років найпоширенішим засобом введення-виведення була перфокарта. 10. Людина може безпосередньо спілкуватися з комп'ютером за допомогою терміналу, використовуючи для введення даних та команд клавіатуру, дуже схожу на клавіатуру друкарської машинки. 11. Найповільнішою операцією, що її виконує комп'ютер, є передача даних, особливо коли дані надходять від людини або надсилаються їй. 12. Розподіл часу, який застосовується на великих комп'ютерах, дозволяє декільком користувачам за різними терміналами одночасно користуватись одним комп'ютером.

**Exercise 5. Answer the questions to the text.** 1. What did computer's main storage consist of in the past and what is it made up of now? 2. What is the difference in storage capacity between modern semiconductor memory devices and their predecessors based on vacuum tubes or ferromagnetic cores? 3. What is the difference between RAM and ROM? 4. What kind of storage can save programs and data that are not currently being used? What materials were used for this kind of storage in the past and what materials are used now? 5. What types of CD and DVD disks do you know? Dwell on them. 6. What advantages does magnetic and optical storage have when compared with semiconductor memory? 7. What is the principle of storing data in magnetic and optical disk systems? 8. What are removable magnetic disks most commonly made of? What is their capacity? And that of compact and DVD disks? 9. What is the typical arrangement of non-removable disks? 10. What input/output medium was the most popular for many years and what input/output devices are the most common now? How can a user communicate with the computer through computer terminals? 11. What is the main reason for sharing the computer's resources? 12. What are the peculiarities of time sharing for multiple users and for a single user?

## TOPIC 10

### COMPUTER ORGANIZATION

Computer organization helps optimize performance-based products. For example, software engineers need to know the processing ability of processors. They may need to optimize software in order to gain the most performance at the least expense. This can require quite detailed analysis of the computer organization. For example, in a multimedia decoder, the designers might need to arrange for most data to be processed in the fastest data path. Computer organization also helps plan the selection of a processor for a particular project. Multimedia projects may need very rapid data access, while supervisory software may need fast interrupts. Sometimes certain tasks need additional components as well. For example, a computer capable of virtualization needs virtual memory hardware so that the memory of different simulated computers can be kept separated. The computer organization and features also affect the power consumption and the cost of the processor. Performance. Computer performance is often described in terms of clock speed (usually in MHz or GHz). This refers to the cycles per second of the main clock of the CPU. However, this metric is somewhat misleading, as a machine with a higher clock rate may not necessarily have higher performance. As a result manufacturers have moved away from clock speed as a measure of performance. Computer performance can also be measured with the amount of cache a processor has. If the speed, MHz or GHz, were to be a car then the cache is like the gas tank. No matter how fast the car goes, it will still need to get gas. The higher the speed, and the greater the cache, the faster a processor runs. Modern CPUs can execute multiple instructions per clock cycle, which dramatically speeds up a program. Other factors influence speed, such as the mix of functional units, bus speeds, available memory, and the type and order of instructions in the programs being run. There are two main types of speed: latency and throughput. Latency is the time between the start of a process and its completion. Throughput is the amount of work done per unit time. Interrupt latency is the guaranteed maximum response time of the system to an electronic event (e.g. when the disk drive finishes moving some data). Performance is affected by a very wide range of design choices — for example, pipelining a processor usually makes latency worse (slower) but makes throughput better. Computers that control machinery usually need low interrupt latencies. These computers operate in a realtime environment and fail if an operation is not completed in a specified amount of time. For example, computer-controlled anti-lock brakes must begin braking almost immediately after they have been instructed to brake. The performance of a computer can be measured using other metrics, depending upon its application

domain. A system may be CPU bound (as in numerical calculation), I/O bound (as in a webserving application) or memory bound (as in video editing). Power consumption has become important in servers and portable devices like laptops. Benchmarking tries to take all these factors into account by measuring the time a computer takes to run through a series of test programs. Although benchmarking shows strengths, it may not help one to choose a computer. Often the measured machines split on different measures. For example, one system might handle scientific applications quickly, while another might play popular video games more smoothly. Furthermore, designers have been known to add special features to their products, whether in hardware or software, which permit a specific benchmark to execute quickly but which do not offer similar advantages to other, more general tasks. Power consumption. Power consumption is another design criterion that factors in the design of modern computers. Power efficiency can often be traded for performance or cost benefits. With the increasing power density of modern circuits as the number of transistors per chip scales (Moore's law), power efficiency has increased in importance. Recent processor designs such as the Intel Core 2 put more emphasis on increasing power efficiency. Also, in the world of embedded computing, power efficiency has long been and remains the primary design goal next to performance.

**Exercise 1 Find in text the English for:** для того, щоб отримати найбільшу продуктивність; за найменших витрат; зробити так, щоб більша частина даних оброблялася в найшвидшому інформаційному каналі; так, щоб пам'ять різних комп'ютерів, що моделюються, могла залишатися окремою; продуктивність комп'ютера часто характеризується такто- 275 вою частотою; не обов'язково має більшу продуктивність; виробники відійшли від тактової частоти як міри продуктивності; енергоефективністю часто можуть жертвувати заради виграшу в продуктивності або у вартості; актуальність енергоефективності зростає; надавати більшого значення зростанню енергоефективності; головна мета проектування після продуктивності.

**Exercise 2. Translate into English** 1. Мейнфрейми мають багато процесорів для того, щоб можна було забезпечити високу потужність обчислень. 2. Падіння напруги на елементі електричного кола характеризується різницею потенціалів на його виводах (leads). 3. Будь-який підприємець прагне отримати максимальний прибуток за найменших витрат. 4. Через обмежену пропускну здатність інформаційного каналу нам довелося зробити так, щоб більша частина інформації оброблялася послідовно. 5. Із міркувань економії розробники мусили пожертвувати функціональністю цього пристрою заради

виграшу у вартості. 6. Операційна система, що є оптимальною для настільного комп'ютера, необов'язково буде такою для ноутбука або нетбука. 7. Для збільшення швидкодії нової системи розробникам довелося зробити так, щоб більша частина даних опрацьовувалася в найшвидшому інформаційному каналі. 8. Програмування — мій найулюбленіший предмет в університеті, після архітектури комп'ютера. 9. У зв'язку з появою зовнішніх носіїв великої ємності, виробники відійшли від використання гнучких дисків. 10. Для того, щоб зробити свій продукт привабливим для покупця, виробники приділяють значну увагу оригінальності конструкції і інколи навіть жертвують заради цього зручністю в експлуатації.



## Vocabulary

### A

- abacus** — рахівниця
- abstract differences** — усувати відмінності
- access** — доступ
- accommodate** — тут задовольняти
- accomplish** — досягати
- accounting records** — бухгалтерська документація
- acoustic** — акустичний
- acquire** — 1) отримувати; 2) збирати (дані)
- adapter** — адаптер; сполучний прилад (пристрій)
- addition** — додавання
- add-on** — додатковий; допоміжний
- address** — тут розв'язати (проблему)
- administer assignments** — призначати завдання
- advent** — прихід; поява
- affair** — річ, справа
- aggregate** — сукупність
- akin to** — подібно до
- algorithmic trading** — професійна або комерційна діяльність, що базується на використанні алгоритмічних методів
- align** — розташовувати, орієнтувати, вирівнювати
- all rolled into one** — все разом, все вкупі
- alphabetical character** — буква
- alternating current (AC)** — змінний струм
- alternatively** — або
- amber** — 1) янтар, бурштин; 2) бурштиновий
- amplification** — підсилення
- analog** — аналоговий
- analogous** — аналогічний
- analog-to-digital converter** — аналого-цифровий перетворювач (АЦП)
- analogy** — аналогія
- anti-lock brake** — гальмо, що не заклинює
- apparent** — 1) явний, очевидний; 2) позірний
- appliance** — пристрій
- application** — прикладна програма
- application layer** — прикладний рівень
- application server** — сервер прикладних програм

**application software** — прикладне програмне забезпечення  
**appreciate** — усвідомлювати, розуміти  
**approach** — підхід  
**arbitrary** — довільний  
**arise** — поставати, виникати  
**arise (arose, arisen)** — з'явитися, постати  
**arithmetic-logic unit** — арифметико-логічний пристрій  
**arrangement** — розташування, організація  
**array** — матриця, сітка  
**array processor** — матричний (векторний) процесор  
**arsenide** — хім. арсенід  
**artificial intelligence** — штучний інтелект  
**ASCII (American Standard Code for Information Interchange)** —  
Американський стандартний код для обміну інформацією  
**assembly language** — мова асемблера  
**assign** — призначати; приписувати  
**associated** — пов'язаний  
**at all times** — завжди  
**attain** — досягати  
**attempt** — спроба  
**availability** — експлуатаційна готовність, працездатність  
**aware of sth** — 1) обізнаний, поінформований; 2) свідомий чого-небудь

## **В**

**backward(s) compatibility** — зворотня сумісність (сумісність із попередніми версіями)  
**basics** — основи  
**battery** — акумулятор  
**battery gauge** — індикатор акумулятора  
**be at a premium** — високо цінуватися, мати великий попит  
**be at the bottom** — бути в основі  
**be aware of smth** — усвідомлювати що-небудь, бути свідомим чогонебудь  
**be idle** — простоювати  
**beam** — промінь  
**benchmarking** — 1) проведення оцінювальних випробувань; 2) тип виконання програми, що дозволяє простежити кількість циклів ЦП, що витрачається на задану область коду  
**binary digit** — двійкова цифра

**bit-map display** — дисплей із поелементним відображенням, растровий дисплей  
**bit-map graphics** — растрова графіка  
**blemish** — недолік, вада  
**block addressing scheme** — схема адресації блоків  
**blueprint** — проект  
**blur** — робити нечітким, затуманювати  
**Boolean operator** — знак логічної операції, логічний оператор  
**boot up** — виконувати початкове завантажування  
**bootstrapping** — розкрутка  
**branch** — відгалуження (схеми)  
**branch node** — вузол розгалуження  
**breach** — прогалина  
**breakthrough** — прорив  
**brevity** — короткість, стислість  
**bridge** — долати перешкоди, наводити мости  
**bridge** — міст (пристрій, що з'єднує локальні мережі, які можуть мати різну топологію, але однаковий протокол)  
**bridge a gap** — ліквідувати розрив  
**bring down** — руйнувати, спричиняти крах  
**broadcast** — передавати; розповсюджувати  
**bubble memory** — пам'ять на циліндричних магнітних доменах (ЦМД-пам'ять)  
**bubblejet** — струменевий принтер  
**build up** — нарощувати  
**bulb** — лампа (електрична)  
**bus speed** — частота шини, швидкість шини  
**business process** — виробничий процес  
**business processing** — оброблення ділової інформації  
**by means of** — за допомогою

## С

**cable** — кабель  
**cache** — кеш-пам'ять  
**calculus** — числення  
**call upon sb/sth** — звертатися до кого-небудь / чого-небудь  
**capacitance** — ємнісний опір  
**capacitor** — конденсатор  
**capacity** — ємність / об'єм (пам'яті)

**captivating** — захоплюючий  
**card reader** — пристрій для читання карт  
**care** — ретельність, уважність, обережність  
**carry out** — виконувати  
**category theory** — теорія категорій  
**cater for sth/sb** — 1) бути призначеним для чого-небудь (кого-небудь);  
**cathode-ray tube (CRT)** — електронно-променева трубка  
**cause** — 1) спричиняти; 2) змушувати  
**caustic** — їдкий, колючий, в'їдливий  
 **caveat** — застереження  
**cell** — елемент живлення, батарейка  
**cellular architecture** — стільникова архітектура  
**cellular phone** — стільниковий телефон  
**census** — перепис (населення)  
**character** — символ  
**charge** — заряд  
**charge** — заряджати (акумулятор)  
**chassis** — системний блок  
**chip floorplan** — базовий план кристала  
**chipset** — мікропроцесорний набір  
**circuit** — ел. коло, схема  
**circuit board** — схемна плата  
**circuitry** — схеми  
**circulatory system** — система кровообігу  
**clarity** — ясність  
**clock rate** — тактова частота  
**clock speed = clock rate** — тактова частота  
**close to** — майже  
**closed circuit** — замкнене коло  
**closeness** — близькість, наближеність  
**clout** — поштовх  
**clumsy** — незграбний  
**coaxial cable** — коаксіальний кабель  
**cohesive architecture** — зв'язана архітектура  
**coin** — створювати, винаходити (нові слова)  
**collaborate** — співпрацювати  
**collection** — сукупність  
**collectively** — спільно, разом, у сукупності  
**combatant** — 1) боєць; 2) поборник

**come to terms with sth** — миритися, дійти згоди з чим-небудь / ким-небудь  
**command-line interface** — інтерфейс «рядок-команда»  
**commence** — починати  
**common** — поширений  
**commonly** — зазвичай, як правило  
**commonplace** — поширений  
**communication** — тут повідомлення  
**communication server** — процесор-шлюз (у мережі передачі даних)  
**communications infrastructure** — інфраструктура зв'язку  
**communications network** — мережа зв'язку  
**communities** — працівники, кадри (у даній сфері)  
**compatible** — сумісний  
**compiler** — компілятор  
**complicated** — складний  
**composite** — суміш; композит  
**computer architecture** — архітектура комп'ютера  
**computer network** — комп'ютерна мережа  
**computer-aided design** — автоматизоване проектування  
**computer-aided, computer-assisted** — автоматизований  
**conceivable** — мислимий  
**conceive** — уявляти собі  
**concern** — проблема (яка потребує вирішення)  
**concerned** — такий, що має стосунок; відповідний  
**conducting** — електропровідний  
**conductor** — провідник  
**confine** — 1) межа; 2) обмежувати  
**conform to smth** — узгоджуватися із чим-небудь  
**congested** — перевантажений  
**consecutive** — послідовний  
**consideration** — міркування  
**consistent interface** — сумісний інтерфейс  
**constituent part** — складова частина  
**constitute** — становити собою  
**construction** — будівництво  
**consumer electronics** — побутова електроніка  
**consumption** — споживання  
**contention** — суперечка  
**contentious** — дискусійний, суперечливий  
**continuous** — неперервний

**contract** — контракт [набір чітко визначених умов, що регулюють відносини між класом-сервером (supplier) і його клієнтами (clients)]  
**contradistinction** — протиставлення, протилежність  
**contribute** — робити внесок, сприяти  
**contribution** — вклад, внесок  
**control** — керувати  
**control parallelism** — паралелізм керування  
**control unit** — пристрій керування  
**controversy** — суперечка, дискусія, полеміка  
**conventional circuit** — звичайна схема  
**convert** — перетворювати  
**core** — ядро  
**count** — лічити  
**courteous** — ввічливий, чемний  
**CPU (I/O, memory) bound** — обмежений швидкістю процесора (можливостями пристроїв введення-виведення, об'ємом пам'яті)  
**cripple** — завдавати шкоди, калічити  
**critical** — важливий  
**cross-fertilization** — взаємне збагачення (ідеями)  
**crucial** — ключовий, важливий  
**crude** — сирий, недосконалий  
**cue** — театр. репліка  
**curiously** — цікаво  
**current** — струм  
**customize a PC** — пристосовувати ПК до потреб  
**cut across** — охоплювати, включати  
**cycle per second** — циклів за секунду

## **D**

**data item** — елемент даних  
**data path** — 1) інформаційний канал; 2) тракт даних  
**date to** — датуватися  
**DBCS (Double-Byte Character Set)** — набір двобайтових символів  
**deal with** — мати справу з  
**debug** — налагоджувати (програму)  
**DEC PDP series** — серія 16-розрядних міні-комп'ютерів фірми DEC (Digital Equipment Corporation)  
**decide on** — вибрати

**decimal** — десятковий  
**declarative language** — декларативна мова  
**decryption** — розшифрування  
**default user interface** — стандартний інтерфейс користувача  
**delete** — видаляти; знищувати  
**denizen** — житель, мешканець  
**densely** — щільно  
**deploy** — розгортати, організувати  
**deployment** — застосування, використання  
**descend into** — перейти в  
**descendant** — нащадок  
**design** — розробка  
**design choice** — конструкційне рішення  
**design implementation** — конструкційна реалізація  
**desktop publishing** — комп'ютерна верстка (друкованих видань)  
**development cycle** — цикл розробки  
**device** — пристрій  
**devise** — розробляти  
**dial-up connection** — з'єднання за телефоном  
**die** — кристал  
**die-cast** — литий  
**difference engine** — різницева машина  
**digital broadcast** — цифрова телепередача  
**digital computer** — цифровий комп'ютер  
**digital-to-analog converter** — цифроаналоговий перетворювач (ЦАП)  
**dimensions** — розміри  
**dimmer** — реостат  
**direct current (DC)** — постійний струм  
**discrete number** — дискретне число  
**disillusion** — розчаровувати  
**disk drive** — 1) дисковод; 2) дисковий накопичувач  
**dismay** — бентежити, засмучувати  
**display** — 1) відображення; 2) дисплей; 3) відобразити  
**disruption** — порушення (в роботі)  
**distributed array processor** — розподілений матричний процесор  
**distributed computing** — розподілене обчислення  
**distributed system** — розподілена система  
**diverse** — різний, різноманітний  
**diversified** — різноманітний

**division** — ділення  
**DLL (Dynamic-Link Library)** — бібліотека, що динамічно підключається;  
бібліотека динамічного компонування (зв'язків)  
**domain theory** — теорія доменів  
**doubtful** — сумнівний  
**doughnut** — тороїд  
**drive** — 1) повід; 2) накопичувач  
**drive assembly** — вузол дисководу  
**DRM (Digital Right Management)** — управління правами на цифрові матеріали  
**dual-core processor** — двоядерний процесор  
**dumb terminal** — «німий» (неінтелектуальний) термінал  
**durable** — довгочасний; такий, що має тривалий строк служби  
**dwarf** — гном, карлик

## Е

**effect** — тут забезпечувати виконання  
**effectively** — успішно  
**efficient** — ефективний  
**electrical engineering** — електротехніка  
**electromotive force** — електрорушійна сила  
**electron tube** — електронна лампа  
**embed** — вмонтовувати  
**embellish** — прикрашати (тж. перен.)  
**employ** — застосовувати, використовувати  
**enable** — давати можливість, дозволяти  
**enclose** — вкладати  
**encode** — кодувати  
**encompass** — охоплювати  
**encryption** — кодування  
**encryption hardware** — шифрувальна апаратура  
**engineering activities** — конструкторська робота  
**enhance** — вдосконалювати, покращувати; посилювати  
**enormous** — величезний  
**ensure** — гарантувати, забезпечувати  
**entanglement** — заплутаність  
**entry** — елемент (таблиці, списку, каталогу)  
**entry level** — перший рівень  
**envision** — уявляти собі



**ergonomic** — ергономічний  
**estimation** — оцінка  
**Ethernet** — стандарт організації локальних мереж [найпоширенішою організацією Ethernet є локальна мережа 10BaseT; розвитком технології Ethernet є Fast Ethernet (100 Мбіт/с)]  
**ever-larger** — постійно зростаючий  
**exacerbate** — посилювати, поглиблювати  
**exchange** — обмінюватися  
**execution** — виконання  
**exhibit** — 1) показувати; 2) виставляти  
**expand** — розширювати  
**expandability** — розширюваність  
**experience** — враження  
**experimentation** — експериментування, проведення експериментів  
**expertise** — спеціальні знання  
**explicit** — явний  
**exploit** — використовувати (у своїх інтересах)  
**explosive** — вибуховий, стрімкий  
**express purpose** — спеціальна мета  
**expression** — вираз  
**extendibility** — здатність розширюватися  
**extension** — розширення, доповнення  
**extensively** — широко  
**extract** — уривок

## F

**fade away** — поступово зникати  
**fail to do sth** — не змогти що-небудь зробити  
**fall short of sth** — не досягати чого-небудь  
**famously** — чудово, відмінно  
**fault** — несправність  
**fault tolerant** — відмовостійкий  
**feature** — тут містити в собі  
**feed (into)** — подавати, вводити (інформацію, сигнал)  
**feedback** — зворотний зв'язок  
**fetch** — вибирати (інформацію з пам'яті)  
**fiber-optic line** — оптоволоконна лінія  
**file backup** — створення резервної копії файла  
**file dump** — роздрукування файла  
**file server** — файловий сервер (вузол збереження файлів у мережі, які

доступні всім її користувачам)  
**File Transfer Protocol (FTP)** — протокол передачі файлів  
**file updating** — внесення змін у файлі  
**fine print** — дрібний шрифт  
**finite** — скінченний, обмежений  
**fit** — оснащувати  
**flame** — груба, неприємна, принизлива або сувора відповідь, скандал  
**flash drive** — флеш-накопичувач  
**floating-point number** — число з рухомою крапкою  
**fluorescent lights** — люмінесцентне випромінювання  
**font** — шрифт  
**forge** — підробляти, фальсифікувати  
**form factor** — форм-фактор  
**form letter** — лист на бланку  
**fortunate** — щасливий  
**fraction** — частка  
**free of charge** — безкоштовно  
**from scratch** — від самого початку  
**front-side bus** — шина управління  
**frustrating** — такий, що засмучує, розчаровує  
**full-blown** — справжній  
**functional language** — функціональна мова  
**functional parallelism** — функціональний паралелізм  
**fuse** — запобіжник (плавкий)

## G

**gain currency** — набувати поширення  
**garner** — накопичувати, збирати  
**gate** — вентиль, логічний елемент  
**gateway** — шлюз (мережевий пристрій або комп'ютер, що здійснює зв'язок між комп'ютерними мережами, які використовують різні комунікаційні протоколи)  
**general purpose computer** — комп'ютер загального призначення  
**generality** — універсальність  
**give much thought to smth** — звертати значну увагу  
**glibly** — правдоподібно  
**go on to smth** — переходити до чого-небудь  
**grade school = elementary school** — початкова школа (US)  
**gradually** — поступово

**grammar gaffe — граматична помилка**

## **Н**

**handle — керувати, оперувати**

**handle needs — задовольняти потреби**

**harbinger — провісник**

**hard coating — тверде покриття**

**hard disk — жорсткий диск**

**hardware — апаратне забезпечення**

**hardware design engineering — проектування апаратного забезпечення**

**have something in mind — збиратися (мати намір) що-небудь зробити**

**head unit — блок головок**

**help desk — служба комп'ютерної допомоги**

**hexadecimal — шістнадцятковий**

**hierarchy network — ієрархічна мережа**

**high resolution — висока роздільна здатність**

**highly — дуже; надзвичайно**

**highway — магістраль**

**hit — 1) відповідь (при пошуку); 2) збіг (при пошуку)**

**host — велика кількість, безліч**

**host computer — хост-комп'ютер, головний комп'ютер (у мережі)**

**host language — базова мова**

**hub — хаб, концентратор**

**human activities — людська діяльність**

**hybrid — гібридний**

**hyperlink — гіперпосилання**

**hyper-threading technology — гіперпотоктова технологія**

## **І**

**identify — визначити**

**image — зображення**

**imperative language — імперативна мова**

**impervious — стійкий, непроникний**

**impetus — поштовх, стимул, імпульс**

**implementation — реалізація**

**implementation language — мова виконання**

**implication — підтекст, прихований зміст**

**imply — мати на увазі, мати значення, натякати**

**imprint — друкувати, тут наносити**

**in one's own right — сам по собі**

**inconsequential** — несуттєвий; такий, що не має значення  
**incorporate** — включати (до свого складу)  
**increasingly** — дедалі більше  
**increment** — приріст, інкремент  
**index** — 1) індекс (у базах даних); 2) індексувати  
**index database** — індексна база даних  
**indexing** — індексна адресація  
**individual** — окремих  
**inductance** — індуктивний опір  
**inductor** — котушка індуктивності  
**industrial process control** — керування технологічним процесом  
**inevitable** — 1) неминухий; невідворотний; 2) незмінний  
**infinite** — нескінченний  
**inherent** — притаманний, властивий  
**inhibit** — заважати, перешкоджати; стримувати  
**input** — 1) n вхід; 2) n введення; 3) v вводити  
**input data** — вхідні дані  
**input/output units** — пристрої введення-виведення  
**inquiry** — питання, запит  
**insert** — вставляти  
**installation** — установка, інсталяція  
**instruction** — команда  
**Instruction Set Architecture (ISA)** — структура системи команд  
**insulator** — ізолятор  
**integer** — ціле (число)  
**integrated circuit** — інтегральна схема  
**intensity** — інтенсивність, яскравість  
**interconnected** — взаємопов'язаний  
**interconnects** — схема з'єднань, розводка  
**interference** — перешкода  
**intermediary** — посередник  
**Internet Engineering Task Force (IETF)** — інженерна група з розвитку Інтернету  
**Internet Protocol Suite** — стек протоколів IP  
**interoperate** — взаємодіяти  
**interpretable symbol** — інтерпретовний символ  
**interpreter** — інтерпретатор  
**interrupt latency** — затримка (оброблення) переривання  
**invalidation** — визнання недійсним, анулювання

**invariably** — незмінно, постійно  
**inventory records** — інвентарна документація  
**investigation** — дослідження  
**involve** — містити, мати в собі

## **К**

**keep track** — відстежувати  
**keep track of grades** — вести облік оцінок  
**kernel** — ядро  
**key** — клавіша  
**keyboard** — клавіатура  
**knock out** — вивести з гри, вимкнути

## **Л**

**language processor** — транслятор  
**lanyard** — ремінець (напр., бінокля)  
**laptop computer** — портативний комп'ютер  
**large scale of integration (LSI)** — інтеграція високого рівня  
**lastly** — зрештою  
**latch** — зачіпка (цифрова логічна мікросхема, що використовується для проміжного зберігання одного або більше бітів, напр.,  
**address latch** — зачіпка адреси, **level latch** — зачіпка рівня сигналу)  
**latency** — 1) час (період) чекання; 2) затримка, запізнення  
**layered system** — рівнева система; пошарова система  
**layman** — не професіонал, не спеціаліст  
**LCD panel** — рідкокристалічна індикаторна панель  
**leased line** — виділена лінія  
**ledgerlike format** — формат бухгалтерської книги  
**libra (lb)** — фунт, pl. *librae*  
**linking loader** — зв'язуючий завантажувач (завантажувач із редагуванням зв'язків)  
**liquid crystal display (LCD)** — рідкокристалічний дисплей  
**liquidity** — ліквідність  
**listing** — лістинг, роздруківка  
**load** — завантажувати  
**local-area network (LAN)** — локальна мережа  
**locate** — розміщувати  
**loosely-coupled systems** — система зі слабким зв'язком

**lurk** — 1) ховатися (причаїтися) в засідці; 2) залишатися непоміченим; критися

**luxuriously** — пишно, розкішно

## **М**

**machine language** — машинна мова

**machinery** — 1) устаткування; 2) корпус

**magnitude** — значення

**mailbox** — поштова скринька

**mainframe computer** — мейнфрейм, комп'ютер великої потужності

**mainstream** — той, що становить більшість, пересічний

**mainstream circles** — маси

**maintainability** — експлуатаційна надійність; зручність експлуатації

**maintenance** — технічне обслуговування

**make for** — сприяти

**make sure** — пересвідчуватися

**male connector** — рознімач-вилка

**management** — керування

**manipulate** — оперувати, маніпулювати

**manipulating capabilities** — можливості обробки (інформації)

**match with size** — підганяти (узгоджувати) за розміром

**mathematical / engineering emphasis** — математичний / технічний ухил

**meaningful** — значущий

**meet smb's needs** — задовольняти потреби

**megaflops** — мільйони операцій із рухомою комою за секунду

**memory** — пам'ять

**memory unit** — пристрій пам'яті

**mere anarchy** — повна анархія

**mesh topology** — змішана топологія

**meta tag** — мета-тег (тег верхнього рівня, який містить ідентифікаційні ознаки, але який не впливає на екранні відображення web-сторінок)

**metric** — показник

**mighty** — масивний, могутній

**milestone** — віха

**mine data** — оримувати інформацію, дані

**miscellaneous issues** — різні питання

**misleading** — оманливий, дезорієнтувальний

**mission** — місія, завдання; покликання; доручення

**missive** — офіційний лист; послання

**mnemonics** — мнемоніка  
**modularity** — модульність, блочність  
**mortar** — мортира  
**motherboard** — материнська плата  
**motion picture** — фільм, кінозображення  
**MPEG (Motion Pictures Experts Group)** — 1) експертна група з кінематографії (займається алгоритмами ущільнення відеозображень); 2) стандарт на ущільнення та відтворення рухомих зображень, розроблений групою експертів, що зазначена в п. 1  
**muddy** — робити щось неясним, заплутувати  
**multi-lateral** — багатосторонній  
**multiple** — декілька, більше ніж один  
**Multiple Instructions-Multiple Data Stream (MIMD) processing** — архітектура з декількома потоками команд і декількома потоками даних (архітектура, в якій набір процесорів незалежно виконує різні набори команд, які обробляють різні набори даних)  
**multiplication** — множення  
**multitasking** — багатозадачний режим, мультипрограмування  
**multithreading** — багатопотоковість, багатопотоковий режим  
**mylar** — майлар (пластмаса)

## N

**needed** — необхідний, потрібний  
**negate** — зводити нанівець  
**NetBEUI** — скор. від NetBIOS Extended User Interface — транспортний протокол, що використовується всіма мережевими ОС фірми Microsoft  
**netiquette (від network etiquette)** — мережевий етикет, правила поведінки в Інтернеті  
**network interface card (NIC)** — мережева інтерфейсна плата  
**neural network** — нейронна мережа  
**newsgroup** — форум у мережі Інтернет  
**node** — вузол (комп'ютерної мережі)  
**Non-Uniform Memory Access (NUMA) computer** — комп'ютер із технологією доступу до неоднорідної пам'яті  
**not to sit well with sb** — не подобатися кому-небудь  
**notably** — особливо  
**notational system** — система позначень  
**notepad** — блокнот

**notion** — ідея, поняття  
**numeric keypad** — цифрова клавіатура, цифрова клавішна панель  
**numerical** — чисельний, числовий  
**numerical analysis** — числовий аналіз  
**numerical aperture** — числова апертура

## О

**obscene** — непристойний, неподобний; брудний  
**occur** — траплятися  
**offer** — тут забезпечувати  
**off-load** — ненавантажений, непрацюючий  
**oil-field** — нафтове родовище  
**on/off switch** — двопозиційний перемикач  
**one-off** — одноразовий (тут спеціальний)  
**one-to-one** — однозначний  
**opcode** — код операції  
**open (web) directory** — відкрита директорія (баз даних) вибраних веб-сайтів  
**open circuit** — розімкнене коло  
**Open Systems Interconnection (OSI)** — взаємодія відкритих систем  
**operate on smth** — виконувати операцію / операції над чим-небудь  
**operation** — 1) робота, функціонування; 2) операція  
**optical fibre** — оптичне волокно  
**optical pickup** — оптичний зчитувальний пристрій  
**origin** — походження  
**originally** — спочатку  
**orthogonality** — ортогональність  
**oscillator** — генератор  
**output** — 1) n вихід; 2) n виведення; 3) v виводити  
**outset** — початок, початковий етап  
**overheads** — накладні витрати  
**overlap** — перекривати  
**overlook** — 1) недооцінювати; 2) ігнорувати, не звертати уваги  
**overrate** — переоцінювати  
**overwhelming success** — приголомшливий (вражаючий) успіх

## Р

**packet switching** — пакетне підключення  
**padded** — оббитий (м'яким матеріалом)



**padded shoulder strap** — ремінець для носіння на плечі, оздоблений м'яким матеріалом

**page layout** — компонування сторінки

**pamphlet = brochure** — брошура, буклет, проспект

**paragraph indention** — виділення абзацу

**parallel circuit** — паралельне коло

**parallel/serial port** — паралельний / послідовний порт

**particular** — конкретний

**password** — пароль

**pattern matching** — зіставлення зі зразком; ототожнення

**pay slip** — бланк платіжної відомості

**payroll** — платіжна відомість (тут назва відповідної прикладної програми)

**PCI** — **peripheral components interconnect** — між'єднання периферійних компонентів

**peering agreement** — договір із рівноправного інформаційного обміну

**peer-to-peer technology** — технологія однорангового або прямого зв'язку

**penetrate** — проникати

**perceive** — 1) розуміти, усвідомлювати; 2) сприймати

**performance studies** — дослідження експлуатаційних характеристик

**permanent storage** — постійна пам'ять

**persist** — залишатися, зберігатися

**personal digital assistant (PDA)** — електронний секретар

**phenomenon** — явище

**photolithography** — фотолітографія

**pick** — вибирати

**pick sth from sth** — вибирати що-небудь звідки-небудь

**pin architecture** — пін-архітектура

**pipeline technique** — конвеєрний метод

**pipelining a processor** — конвеєризація процесора

**plague** — надокучати, набридати

**platter** — велика пласка тарілка, тареля

**plot** — ділянка

**plug** — вставляти (в контактне гніздо)

**plug in** — під'єднувати, підключати

**plummet** — падати

**pointing device** — пристрій управління курсором (миша, світлове перо, джойстик)

**possess sth** — мати що-небудь, володіти чим-небудь  
**potential difference** — різниця потенціалів  
**power density** — щільність енергії  
**power efficiency** — енергоефективність  
**power supply** — джерело (блок) живлення  
**power user** — досвідчений, кваліфікований користувач; користувач із підвищеними вимогами до продуктивності комп'ютера  
**predate** — статися раніше чого-небудь, випереджати що-небудь у часі  
**predecessor** — попередник  
**preferable** — кращий  
**preprogram** — запрограмувати наперед  
**preschool** — дошкільний заклад  
**primarily** — 1) головним чином; 2) передусім  
**printed circuit board (PCB)** — друкована плата  
**probe** — зонд; датчик  
**process** — обробляти  
**processing** — обробка  
**processing engine** — механізм обробки  
**program entry** — введення програми  
**prominent** — відомий, знаний  
**promise** — перспектива  
**proof** — випробування  
**proprietary** — власний, розроблений всередині фірми для власних цілей  
**props** — реквізит  
**provided that** — за умови, що  
**proximity search** — пошук із відстанню  
**pull-down menu** — низхідне меню  
**pulse of electricity = electric pulse** — електричний імпульс  
**punch card** — перфокарта  
**punched-paper tape** — перфораційна стрічка  
**purchase** — купувати  
**pursuit** — прагнення, домагання; гонитва

## Q

**quantum computer** — квантовий комп'ютер  
**query** — запит  
**queueing theory** — теорія черг  
**quip** — насміхатися, висміювати, глузувати

**quirk** — 1) індивідуальна особливість; 2) примха, випадковість  
**quite a number** — чимало

## R

**radio wave** — радіохвиля  
**random access memory (RAM)** — пам'ять із довільним доступом (оперативна пам'ять)  
**random-access memory** — пам'ять із довільним доступом (оперативний запам'ятовувальний пристрій)  
**range** — діапазон  
**rationale** — розумне пояснення  
**readily** — легко, відразу  
**read-only memory** — пам'ять тільки для читання (постійний запам'ятовувальний пристрій)  
**reception** — прийом  
**rechargeable** — перезаряджуваний  
**rechargeable battery** — акумулятор  
**recognize** — бачити, усвідомлювати  
**records** — 1) реєстрація, облік; 2) документація, дані  
**rectangular** — прямокутний  
**recurring difficulty** — проблема, що періодично повторюється  
**reduce** — зменшувати  
**redundancy** — надлишок, дублювання  
**redundant bit** — надлишковий розряд  
**reference to sth** — згадування чого-небудь  
**reference work** — довідник  
**regarding** — щодо  
**regardless of** — незважаючи на, незалежно від, попри  
**register-transfer level** — рівень регістрових передач  
**related devices** — пов'язані пристрої  
**relay** — реле  
**release** — випускати  
**reliability** — надійність  
**relinquish** — відмовлятися, поступатися  
**rely on** — базуватися на  
**remailer** — переадресатор, поштовий посередник; ретранслятор  
**remote** — 1) віддалений; 2) несхожий  
**remote machine** — віддалена машина  
**removable** — знімний

**report generation** — генерація звітів  
**represent** — представляти  
**representation** — представлення  
**request** — 1) запит; 2) робити запит  
**Request for Comment (RFC)** — запит на коментар  
**requisite** — потрібний  
**research article/research paper** — наукова стаття  
**researcher** — дослідник  
**resemble** — бути схожим, нагадувати  
**reservation** — бронювання, попереднє замовлення  
**resistance** — опір  
**resistor** — резистор  
**resolution** — рішення  
**respectively** — відповідно  
**respond to sth** — відповідати чому-небудь  
**restrict** — обмежувати  
**retention cycle** — термін зберігання  
**retract** — втягувати(ся)  
**retrieval** — пошук  
**retrieve** — здійснювати пошук (інформації)  
**reuse** — багатократне використання  
**revenue** — дохід, доходи  
**rigorous** — ретельний, суворий  
**RISC** — **reduced instruction set computing / computer** —  
обчислення / комп'ютер зі скороченим набором команд  
**ise** — підйом  
**robust** — міцний  
**router** — маршрутизатор  
**routine** — 1) підпрограма; 2) стандартна програма  
**run** — виконувати (програму)

## S

**satellite** — супутник  
**SBCS (Single-Byte Character Set)** — набір однобайтових символів  
**scalability** — здатність до рівноваги, врівноважуваність  
**scalable** — масштабований, нарощуваний  
**scalar processor** — скалярний процесор  
**scheduling system** — планувальник  
**school board** — 1) відділ середньої освіти; 2) шкільна рада

**screen** — 1) екран; 2) зображення на екрані  
**scroll bar** — лінійка прокрутки  
**SCSI** /'skʌzi/— **small computer system interface** — інтерфейс малих комп'ютерних систем, сказі-інтерфейс  
**seamless** — плавний  
**seamless migration** — легкий перехід  
**search** — здійснювати пошук  
**search engine** — пошукова система  
**security vulnerabilities** — вразливості захисту  
**seeing that** — беручи до уваги, що  
**self-consistency** — самоузгоджуваність, несуперечність  
**semiconductor** — напівпровідник  
**semi-solid** — напівтвердий  
**series** — послідовність, ряд  
**series circuit** — послідовне коло  
**set** — множина  
**setting** — оточення  
**shakeout** — витіснення з ринку дрібних фірм; розорення фірм, які не витримують конкуренції  
**share** — 1) розподіляти; 2) спільно використовувати, розділяти  
**share information** — спільно використовувати інформацію  
**share sth in common** — мати що-небудь спільне  
**sharing resources** — спільне користування ресурсами  
**shell** — оболонка ОС; командний процесор в ОС  
**ship out** — відправляти  
**shoot out** — тут видавати  
**short circuit** — коротке замикання  
**show strengths** — виявляти переваги  
**show up** — з'являтися  
**shrink** (past shrank, shrunk; pp shrunk) — стискувати, зменшувати  
**SI unit** — одиниця СІ  
**similar** — схожий  
**simplicity** — простота  
**simulation** — моделювання  
**Single Instruction-Multiple Data (SIMD) stream architecture** — архітектура з одним потоком команд і декількома потоками даних (архітектура паралельної комп'ютерної системи, що припускає виконання однієї поточної команди декількома процесорами)  
**single sided disc** — односторонній диск

**single / dual-layer disc** — одно-/ двошаровий диск  
**single-tasking operating system** — однозадачна ОС  
**smarts** — US, розм. мозок  
**snail mail** — звичайна пошта  
**snappy** — дотепний; енергійний; модний  
**social networking** — взаємодія всередині соціумної мережі  
**software** — програмне забезпечення (ПЗ)  
**software engineering** — програмотехніка, інженерія розробки ПЗ  
**software house** — фірма з розробки та реалізації програмних засобів  
**solely** — тільки, виключно  
**somewhat** — почасти, деякою мірою  
**sorting** — сортування  
**source language** — вхідна мова  
**source page** — початкова сторінка  
**source program** — початкова програма  
**specially arranged** — спеціально, в особливий спосіб організований  
**special-purpose computer** — комп'ютер спеціального призначення  
**specification** — тут деталізація, уточнення  
**specified** — точно визначений, зазначений, заданий  
**specify** — задавати, визначати  
**speed** — швидкодія  
**spelling slip** — орфографічна помилка  
**spin** — крутити(ся), обертатися (навколо своєї осі)  
**split into** — розділятися на  
**spreadsheet** — електронна таблиця  
**spurn** — зневажливо ставитися; гордовито відхиляти  
**stagehand** — робітник сцени  
**stand-alone** — автономний  
**standard/high definition video** — відео стандартної/високої чіткості  
**standard-setting work group** — робоча група, що встановлює стандарти  
**start out** — розпочинати (професійну діяльність)  
**status** — стан  
**stay in touch** — залишатися на зв'язку  
**stilted style** — пишномовний стиль  
**storage** — 1) зберігання; 2) пам'ять  
**stream** — потік  
**subscribe** — надсилати запит на отримання повідомлень за списком розсилання або з конференції  
**substance** — речовина

**subtraction** — віднімання  
**successive** — 1) послідовний; 2) наступний  
**sufficiently** — досить  
**superposition** — накладення, суперпозиція  
**supervisory software** — комплект керівних програм  
**survey** — 1) огляд, дослідження; 2) досліджувати, робити огляд  
**susceptible** — чутливий, вразливий  
**suspend** — тимчасово призупиняти  
**sustain** — витримувати  
**switch** — комутатор  
**system bus** — системна шина  
**system case** — корпус комп'ютера  
**system responsiveness** — швидкість реагування системи  
**system software** — системне програмне забезпечення  
**system-resident software** — резидентне програмне забезпечення

## Т

**tailor (to sth)** — пристосовувати (для кого-небудь / чого-небудь)  
**take in** — приймати  
**take over** — приймати; доставляти; з'єднувати  
**tap into a network** — підключатися до мережі  
**target language** — вихідна мова  
**taxonomy** — систематизація, таксономія  
**TCP/IP** — скор. від Transmission Control Protocol / Internet Protocol —  
протокол керування передачею / Інтернет-протокол  
**TCP/IP model** — модель протоколу керування передачею /  
Інтернет-протоколу  
**technique** — метод  
**techno-jargon** — технічний жаргон  
**Telnet** — протокол емуляції терміналу  
**tempting** — спокусливий, привабливий  
**terminal** — клемма  
**that is not to claim that** — йдеться не про те, що  
**theoretical activities** — теоретична діяльність  
**therefore** — тому, внаслідок чого  
**thread** — потік  
**thread-level parallelism** — паралелізм на рівні потоків (команд)  
**three-tier model** — триланкова модель  
**throughput** — 1) продуктивність (обчислювальної системи);

2) пропускна здатність (каналу зв'язку)  
tied in — пов'язаний  
tightly-coupled system — система з сильним зв'язком  
time-share — розподіл у часі  
to compile — 1) компілювати; 2) укладати, упорядковувати  
to incorporate — включати, об'єднувати (в собі)  
to invoke — викликати, активувати (програму, процедуру)  
to supersede — витіснити  
token — маркер  
Token Ring — маркерне кільце (стандарт організації локальної мережі)  
toll-free — безплатний  
toolbar — панель інструментів  
topology — топологія  
touchpad — сенсорна панель  
trace — простежити, встановити  
track finances — вести облік фінансів  
trackball — трекбол (кульовий маніпулятор)  
trade sth for sth — обмінювати що-небудь на що-небудь  
traffic — потік інформації  
transfer point — точка переходу (передачі трафіка)  
transformer — трансформатор  
translator program — програма-транслятор  
transmission — передача  
transmission medium — передавальний засіб  
transport layer — рівень транспортування  
transport stream — транспортний потік (даних)  
trap — пастка  
trend — тенденція  
tribute — данина, належне  
troubleshooter — спеціаліст, що виявляє несправності  
troubleshooting — пошук (і усунення) несправностей  
trunk node — магістральний вузол  
tune — налагоджувати  
turn in — представляти (роботу)  
twisted pair — вита пара  
typify — бути типовим представником, уособлювати



## U

**ultimately** — зрештою, в кінцевому підсумку  
**unavailability** — непрацездатність, неготовність  
**unbounded** — необмежений  
**undaunted** — безстрашний  
**underlie** — лежати (бути) в основі  
**underlying** — основний, базовий  
**Unicode** — стандарт кодування символів усіх національних мов  
**uniprocessor** — монопроцесор  
**Universal Serial Bus (USB)** — універсальна послідовна шина  
**unscrupulous snooper** — непорядна людина, яка втручається в чужі справи  
**upgrade** — 1) модернізація; 2) модернізувати  
**uppercase letter** — велика літера  
**usher in** — сповіщати  
**utility program** — утиліта

## V

**validation** — випробування  
**value** — значення  
**various** — різноманітний  
**vary** — міняти(ся), змінювати(ся)  
**vastly** — тут набагато, незрівнянно  
**vector processor** — векторний процесор  
**venture** — справа (особливо пов'язана зі значним ризиком)  
**versatile** — універсальний  
**very large scale of integration (VLSI)** — інтеграція надвисокого рівня  
**via** — за допомогою  
**vice versa** — навпаки  
**video display monitor** — відеомонітор  
**virtualization** — віртуалізація, створення віртуального середовища  
**virtually** — фактично  
**virus scanner** — програма пошуку вірусів  
**volatile** — енергозалежний  
**volt** — вольт  
**voltage** — напруга  
**volume** — обсяг  
**vulnerable** — вразливий

## W

**watt** — ват

**Web browser** — web-браузер, система перегляду та спрямування інформації в Інтернет

**Web crawler** — навігатор Всесвітньої павутини

**Web crawling** — навігація, повільний рух Всесвітньою павутиною

**web data mining** — отримання даних з Інтернету

**web page** — web-сторінка

**well** — цілком (після модальних дієслів *may, could*)

**wide-area network (WAN)** — широкомасштабна, глобальна мережа

**wield** — володіти, тримати в руках

**wiring technology** — технологія з'єднання

**word processing** — обробка текстів

**work smth** — обробляти що-небудь

**workstation** — автоматизоване робоче місце (АРМ)

**World Wide Web (WWW)** — Всесвітня павутина

## LITERATURE USED

1. Англійська мова. Комп'ютерні науки : навч. посібник / Вереїтіна І.А. - Навч. посіб. — К.: Вища школа, 2008. — 328 с.
2. Bugaiov O. Ye. English. Types of Modern Computers: Methodological Guide / O. Ye. Bugaiov, H. V. Babiy, Ya. V. Absaliyeva. — К. : National Aviation University, 2003. — 68 p.
3. Fundamentals of Information Technology / Ed. By G. G. Wilkinson, A. R. Winterflood. — Chichester : John Wiley and Sons, 1987. — 363 p.
4. Oxford English for Information Technology 2nd Edition: Student's Book: Eric Glendinning, John McEwan, - Oxford Press , 2003. — 222 p.
5. Professional English: Computer Technologies : навч. посіб. / О. М. Акмалдінова, О. Є. Бугайов, Г. В. Бабій, С. А. Волгіна. — К. : НАУ, 2014. — 352 с.
6. <https://grammarway.com/ua/infinitive>
7. <https://www.onlinemathlearning.com/gerunds-esl.html>

**АНГЛІЙСЬКА МОВА ДЛЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ  
«КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»**

---

**ENGLISH FOR COMPUTER ENGINEERING STUDENTS  
Part 2**

**Навчально-методична розробка  
до курсу «Англійська мова» для спеціальності  
«Комп'ютерна інженерія»  
ч.2**

Підписано до друку 13.02.2023 р. Формат 60x90/16.  
Папір офсет. Друк цифр. Ум. друк. арк. 5,08.  
Наклад 100. Зам. № 0268.

Віддруковано в ТОВ «Поліграфцентр «Ліра»:  
88000, м. Ужгород, вул. Митрака, 25

Свідоцтво про внесення до Державного реєстру видавців,  
виготівників і розповсюджувачів видавничої продукції  
Серія ЗТ №24 від 7 листопада 2005 року.