МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Н. М. Акоп’янц

Переклад у галузі інформаційних технологій

Навчальний посібник
для бакалаврів
спеціальності 035 «Філологія»,
спеціалізація 035.041 «Германські мови та літератури
(переклад включно), перша – англійська»
денної та заочної форм навчання

Рекомендовано Вченою радою НТУ «ХПІ»

Затверджено
редакційно-видавничою
радою НТУ «ХПІ»,
протокол № 3 від 24.10.2024 р.

Харків
НТУ «ХПІ»
2024

УДК 81'255.2:62(075) + 004=030.111(075)
А 40

Рецензенти:
І. М. Шульга, канд. пед. наук, доцент кафедри іноземних мов Національного аерокосмічного університету ім. М. Є. Жуковського «ХАІ»;
О. В. Чубукіна, канд. філ. наук, доцент кафедри іноземних мов Національного аерокосмічного університету ім. М. Є. Жуковського «ХАІ»

Рекомендовано Вченою радою НТУ «ХПІ» як навчальний посібик для студентів спеціальності «Філологія» усіх форм навчання, протокол № 10 від 26.12.2024, 27.12.2024 року

The book includes eighteen units that are related to different kinds of professional situations in the field of information technologies and ensure background knowledge and acquisition of the most used lexical units. The book presents authentic materials and various exercises in each unit. Topic vocabulary and texts for self-study are at the end of the book.

For the bachelor students of the specialty 035 “Philology” for Educational program “The Germanic languages and literature (including translation), the English language major for full-time and correspondence educational courses, it can also be useful for students of IT departments.
Bibl. titles 15.

Акоп’янц Н. М.

А 40 Переклад у галузі інформаційних технологій : навчальний посібник / Н. М. Акоп’янц. – Харків : НТУ «ХПІ», 2024 р. – 182 с.

ISBN 978-617-05-0518-7

Навчальний посібник має вісімнадцять розділів, які за змістом охоплюють різні професійні ситуації і забезпечують засвоєння фонових знань і найбільш вживаної лексики у галузі інформаційних технологій. У кожному розділі подано автентичні матеріали, різноманітні вправи. Тематичні словники та тексти для самостійного опрацювання наведено наприкінці навчального посібника.

Призначено для бакалаврів спеціальності 035 «Філологія», спеціалізація 035.041 «Германські мови та літератури (переклад включно), перша – англійська» денної та заочної форм навчання, а також може бути корисним студентам комп’ютерних спеціальностей.

Бібліогр. 15 назв.

УДК 81'255.2:62(075) + 004=030.111(075)

ISBN 978-617-05-0518-7

Зміст

ВСТУП

Навчальний посібник «Переклад у галузі інформаційних технологій», призначений для бакалаврів спеціальності 035 «Філологія», спеціалізація 035.041 «Германські мови та літератури (переклад включно), перша – англійська» денної та заочної форми навчання, підготовлений згідно з «Положенням про організацію навчального процесу у вищих навчальних закладах» та галузевими стандартами вищої освіти. Ідея написання даного посібника спирається на чинну навчальну програму з предмета «Переклад в галузі інформаційних технологій» для бакалаврів спеціальності 035 «Філологія», спеціалізація 035.041 «Германські мови та літератури (переклад включно), перша – англійська» денної та заочної форми навчанняв НТУ «ХПІ».

Однією з головних переваг посібника є його комплексний підхід до викладання матеріалу. Докладно розглядається не тільки переклад термінів, а й надається контекст їх застосування і тлумачення технічної термінології, що використовується у сфері ІТ. Завдяки цьому студенти мають змогу не лише покращити свої перекладацькі навички, а й заглибитися в особливості сучасних технологій.

Посібник має чітку структуру, яка логічно розділяє матеріал на теми, що відповідають ключовим напрямкам ІТ-галузі. У ньому є 18 розділів, присвячених комп'ютерній термінології, вебтехнологіям, програмуванню, безпеці даних, мовам програмування, комп’єтерній етиці, штучному інтелекту та іншим важливим темам. Такий поділ дозволяє студентам поступово розширювати свої знання: від простіших понять до більш складних.

Також у посібнику представлено практичні завдання, що дає можливість студентам закріпити свої знання на практиці. Це важливий аспект навчання, оскільки дозволяє відпрацьовувати набутий матеріал та розвивати практичні навички перекладу. Завдання включають вправи з перекладу автентичних технічних і художніх текстів, опрацювання комп’ютерного вокабуляру, вправи на перевірку розуміння прочитаного, завдання на розвиток письмових вмінь і навичок говоріння з залученням комп’ютерної термінології, з якою фахівці стикаються у своїй повсякденній роботі.

Серед переваг посібника також можна відзначити наявність деяких новітніх термінів та технологій, що є надважливим, оскільки в ІТ-сфері зміни відбуваються дуже швидко. Крім того, глибина пояснень технічних аспектів у деяких розділах може бути особливо корисною для студентів, які ще не мають досвіду у сфері ІТ.

В посібнику представлено англійську термінологію в галузі інформаційних технологій та її переклад, надано автентичні тексти відповідної сфери для практики перекладу, розроблено завдання для перевірки розуміння прочитаного.

Тексти супроводжуються письмовими завданнями й усними питаннями для стимуляції аналітичного мислення й актуалізації англомовного вокабуляру. Наприкінці посібника подано тексти для самостійного перекладу, запитання для самоконтролю і загальний словник термінів.

Включення практичних завдань із перекладу в галузі інформаційних технологій для студентів спеціальності 035 «Філологія», спеціалізація 035.041 «Германські мови та літератури (переклад включно), перша – англійська», є обов`язковим компонентом освітньо-професійної програми для здобуття освітньо-кваліфікаційного рівня бакалавра і має на меті набуття студентом особистісних і професійних компетентностей, пов'язаних із реалізацією можливостей інтеграції комунікативних та фахових знань і вмінь.

В цілому, посібник «Переклад у галузі інформаційних технологій» є корисним для тих, хто прагне спеціалізуватися у сфері перекладу на ІТ-тематику. Він містить необхідний теоретичний матеріал та практичні завдання, що допомагають формувати навички, потрібні сучасному перекладачеві.

Тематичний виклад навчального матеріалу та система різноманітних вправ для його опрацювання й інтенсивного закріплення дозволяє ефективно використовувати посібник у гуманітарних університетах для студентів перекладацьких відділень.

Даний посібник сприяє удосконаленню всієї системи підготовки перекладача, розвитку індивідуальних творчих здібностей майбутніх фахівців, їх ерудиції, об’єктивній оцінці своїх можливостей, застосуванню різноманітних методів та прийомів перекладу текстів, сучасних комп’ютерних технологій, а також вихованню у них подальшої активності та самостійності у набутті професійних, апперцептивних, проєктувальних, адаптивних, організаційних і пізнавальних умінь.

Unit 1. Introduction to Software

Exercise 1. Think and answer:

What kind of software do you usually use and what for?

Exercise 2. Study the list of terms and examples of their usage below:

Abort (перервати) - to end a program or a process before its completion

When the word processor application crashed, the user had to abort the program and lose all his unsaved changes.

Bug (помилка) - an error in a computer program

An average developer will create one bug for every 10 lines of code written.

closed source(закритий код) - software in which the license stipulates that the user cannot see, edit, or manipulate the source code of a software program

I wanted to develop a new feature for the program, but I couldn't because it was closed source.

Compatible (сумісний) - capable of being used without modification

The IBM 360 was the first commercially successful computer family with a wide range of compatible parts.

Crash (збій) - a computer failure due to faulty hardware or a serious software bug

The user was advised to reboot the computer after a serious crash in which the computer no longer responded.

end user (кінцевий користувач) - a person who uses a product or service on a computer

Developers must maintain a close relationship with end users if they want to have a successful career.

Error (помилка) - an incorrect action attributable to poor judgment, ignorance, or inattention

The computer reported a «division by zero» error and automatically aborted the program.

Execute (виконати) - to start a program on a computer

The program was set to execute every night at midnight.

Feature (функція) - something a computer program is «supposed» to do; these are often reasons to use a particular program or upgrade to a more recent version

The man upgraded his copy of Word because of a new feature that allowed him to spell-check documents in Spanish.

IDE (integrated development environment) (інтегроване середовище розробки) - an application normally consisting of a source code editor, a compiler and/or interpreter, build-automation tools, and a debugger

The new employee asked his boss to buy him a license for his favorite IDE because there was none installed on his new company laptop.

Open source (відкритий код) - a program in which the code is distributed allowing programmers to alter and change the original software as much as they like

The article stated that many programmers prefer open source solutions because they can modify features and fix bugs without waiting for an upgrade or patch from the manufacturer.

Programmer (програміст) - a person who writes or modifies computer programs or applications

The software company needed to hire three new programmers to help debug their flagship application.

Proprietary (пропрієтарний/закритий) - privately developed and owned technology

Because of proprietary code, you may not modify or redistribute the source code of Windows or Macintosh operating systems.

Restriction (обмеження) - a rule or law which limits or controls something

The video website had content restrictions in place for users under the age of 18.

Usability (зручність використання) - a measure of how easy or efficient a program is to use

Back in the day, programmers or salespeople would often be responsible for the user experience, but now we assign that task to a properly trained UX designer.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

INTRODUCTION TO COMPUTER SOFTWARE

For as long as there has been computer hardware, there has also been computer software. But what is software? Software is just instructions written by a **programmer**

which tells the computer what to do. Programmers are also known as 'software developers', or just plain 'developers'.

Nothing much is simple about software. Software programs can have millions of lines of code. If one line doesn't work, the whole program could break! Even the process of starting software goes by many different names in English. Perhaps the most correct technical term is '**execute**', as in «the man executed the computer program.» Be careful, because the term 'execute' also means (in another context) to put someone to death! Some other common verbs used to start a software program you will hear are 'run', 'launch, and even 'boot' (when the software in question is an operating system).

Software normally has both **features** and **bugs.** Hopefully more of the former than the latter! When software has a bug there are a few things that can happen. The program can **crash** and terminate with a confusing message. This is not good. End users do not like confusing **error** messages such as:

Site error: the file /home7/businfc6/public_html/blog/wordpress/wp-content/plugins/seo-blog/core.php requires the ionCube PHP Loader ioncube_loader_lin_5.2.so to be installed by the site administrator.

Sometimes when software stops responding you are forced to manually **abort** the program yourself by pressing some strange combination of keys such as ctrl-alt-delete.

Because of poor **usability**, documentation, and strange error messages, programming still seems very mysterious to most people. That's too bad, because it can be quite fun and rewarding to write software. To succeed, you just have to take everything in small steps, think very hard, and never give up.

I think everyone studying Information Technology should learn at least one programming language and write at least one program. Why? Programming forces you to think like a computer. This can be very rewarding when dealing with a wide range of IT-related issues from tech support to setting up PPC (pay-per-click) advertising campaigns for a client's web site. Also, as an IT professional, you will be dealing with

programmers on a daily basis. Having some understanding of the work they do will help you get along with them better.

Software programs are normally written and compiled for certain hardware platforms. It is very important that the software is **compatible** with all the components of the computer. For instance, you cannot run software written for a Windows computer on a Macintosh computer or a Linux computer. Actually, you can, but you need to have special emulation software or a virtual machine installed. Even with this special software installed, it is still normally best to run a program on the kind of computer for which it was intended.

There are two basic kinds of software you need to learn about as an IT professional. The first is **closed source** or **proprietary** software, which you are not free to modify and improve. An example of this kind of software is Microsoft Windows or Adobe Photoshop. This software model is so popular that some people believe it's the only model there is. But there's a whole other world of software out there.

The other kind of software is called **open source** software, which is normally free to use and modify (with some **restrictions** of course). Examples of this type of software include most popular programming languages, operating systems such as Linux, and thousands of applications such as Mozilla Firefox and Open Office.

But what is the real difference between open source and closed source software? Is open source source software just about saving money? Let's investigate. Let's say for instance you find a bug in the latest version of Mozilla Firefox. The bug is causing a major project to fail and you need to fix it right away. This is not very likely to happen, I realize, but it's just an example. You might take the following steps:

Step 1. Download and unzip (or uncompress) the source code from Mozilla.

Step 2. Use an Integrated Development Environment (**IDE**) and a debugger to find and fix the bug in the source code. Please note that you will need to know a little C++ to debug applications such as this.

Step 3. Test the fix and then use a compiler to turn the source code into a binary file. This can take a long time for big programs. Once the source code is compiled then the program should work!

Step 4. You are almost done. Now send the bug fix back to the Mozilla Firefox team. They may even use your bug fix in the next release!

Now imagine you find a bug in a proprietary code base such as Microsoft Word. What can you do? Not much, just file a bug report and hope someone fixes it at some point.

This is a rather radical example, but I think it illustrates to a large degree why programmers generally prefer open source software to closed source alternatives. Good programmers love code and they want access to it. Hiding the code from a programmer is like hiding the car engine from an auto mechanic. We don't like it!

Now you have learned a little about software. You will learn more about software applications and programming in later units. (Adapted from https://www.english4it.com/module/core/unit/2/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1.What is a key difference between open source and closed source software?

a) Closed source software can only be run on Linux systems.

b) Open source software allows users to modify the source code.

c) Closed source software is always free to use.

d) Open source software cannot be used on a Mac.

2.What does the term 'execute' mean in the context of computer software?

a) To install a new program on a computer.

b) To start a software program.

c) To terminate a program with a crash.

d) To delete an operating system.

3. Why is it important for IT professionals to learn programming?

a) To avoid using closed source software.

b) To understand how hardware components interact with software.

c) To better collaborate with programmers and solve IT issues.

d) To create documentation for open source projects.

Exercise 5. Translate this text into English.

Програмне забезпечення – це набір інструкцій, даних або програм, які використовуються для керування комп’ютерами та виконання конкретних завдань. Воно є невід'ємною частиною сучасних технологій і забезпечує функціонування пристроїв від комп’ютерів до смартфонів.

Існує два основних типи програмного забезпечення:

1. **Системне програмне забезпечення** – це програми, які забезпечують базове функціонування комп'ютера, наприклад, операційні системи (Windows, Linux, macOS) та драйвери. Вони створюють середовище для роботи іншого програмного забезпечення.

2. **Прикладне програмне забезпечення** – це програми, які розробляються для виконання конкретних завдань, таких як текстові редактори, графічні редактори, браузери чи ігри.

Програмне забезпечення може бути платним, безкоштовним або умовно безкоштовним. Також важливу роль відіграють програми з відкритим кодом, які можна модифікувати під власні потреби.

Сучасний світ неможливо уявити без програмного забезпечення. Воно допомагає автоматизувати процеси, полегшувати роботу людей та відкриває нові можливості для розвитку науки, бізнесу й освіти.

Writing Activity

Exercise 6. Pick a question and write about it.

- Have you ever written or modified any software? If so, what were the challenges you faced? If not, why not?
- Name three pieces of software you use frequently. Why do you use them? What would you change about them?
- Pretend you are the world's best programmer and can write computer code as fast as you can think. What kind of software would you write?

Speaking Activity

Exercise 7. Discuss the questions.

- What is the basic difference between hardware and software, as described in the text?
- Why do you think software can be so complex? What challenges do developers face when creating it?
- The text mentions different terms used to start software, like ‘execute’ and ‘run’. Why might it be useful to know multiple terms for starting a program?
- What are bugs, and how can they affect the functionality of a software program? Can you give an example of a bug you’ve encountered?
- Why might learning programming be beneficial for someone working in IT?

Unit 2. Operating Systems

Exercise 1. Think and answer:

What sorts of operating systems have you used? What are their strengths and weaknesses?

Exercise 2. Study the list of terms and examples of their usage below:

CLI (command line interface) (інтерфейс командного рядка) - a text-only link between a computer and its operator.

The technician enjoyed administering the Unix server with the CLI tools.

compression (стиснення) - a method of packing data in order to save disk storage space or download time.

Zip and mp3 are examples of two common file compression algorithms.

device driver (драйвер пристрою) - software which converts the data from a component or peripheral into data that an operating system can use.

The IT support technician asked the end user if he had recently updated any device drivers.

file permissions (дозволи на файли) - a set of strict rules for controlling read, write, and execute access to a file or directory.

The company switched from FAT32 to NTFS file system because the former did not support file permissions.

format (форматувати) - prepare a device to store data, erasing any existing data.

One must format a flash memory drive or a hard disk drive before it can be used to store data.

GUI (graphical user interface) (графічний інтерфейс користувача) - an icon based link between a computer and its operator.

Most users prefer an icon-based GUI over a command line option.

hidden file (прихований файл) - a file which does not appear by default in a directory listing; normally for security reasons or to spare confusion in end users.

The administrator was upset when the end user found a way to display and then delete several hidden files.

kernel (ядро) **-** the fundamental part of an operating system responsible resource management and file access.

The system administrator needed to upgrade the kernel in order to provide native support for Serial ATA drives.

Linux - an open source version of Unix developed by Linus Torvalds. *Linux was originally developed by Linus Torvalds, who wanted a free Unix-like operating system that ran on standard PC hardware.*

Multitasking (багатозадачність) **-** concurrent execution of two or more tasks by a processor.

UNIX is the original multitasking environment and was designed from the beginning to share resources over a network.

OS (operating system) (операційна система) **-** a GUI or CLI software link between the computer and operator; also provides a framework for productivity software such as an office suite, web browser, or programming languages.

The programmer tested his Java code on many operating systems including: Windows, Linux, and Macintosh.

sign in (увійти) **-** to enter information related to an account name and its password in order to access a computer resource.

The man could not sign in to his Windows computer because he forgot his username and password.

sign out (вийти) **-** to end a session with computer or network resource. *The server was set to automatically sign out users after 10 minutes of inactivity.*

VM (virtual machine) (віртуальна машина) **-** a software program which mimics the performance of one or more hardware devices in order to run software independently of the actual hardware.

In order to run Java programs in Windows one needs to download an appropriate virtual machine.

X11 (X Window System) - a software toolkit for UNIX systems underlying numerous GUI window managers including KDE and Gnome.

The man was surprised at how similar in functionality X11 was to Microsoft Windows.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Learning About Operating Systems

An operating system is a generic term for the **multitasking** software layer that lets you perform a wide array of 'lower level tasks' with your computer. By low-level tasks we mean:

- the ability to **sign in** with a username and password
- **sign out** the system and switch users
- **format** storage devices and set default levels of file **compression**
- install and upgrade device drivers for new hardware
- install and launch applications such as word processors, games, etc
- set **file permissions** and hidden files
- terminate misbehaving applications

A computer would be fairly useless without an OS, so today almost all computers come with an OS pre-installed. Before 1960, every computer model would normally have its own OS custom programmed for the specific architecture of the machine's components. Now it is common for an OS to run on many different hardware configurations.

At the heart of an OS is the **kernel**, which is the lowest level, or core, of the operating system. The kernel is responsible for all the most basic tasks of an OS such as controlling the file systems and device drivers. The only lower-level software than the kernel would be the BIOS, which isn't really a part of the operating system. We discuss the BIOS in more detail in another unit.

The most popular OS today is Microsoft Windows, which has about 85% of the market share for PCs and about 30% of the market share for servers. But there are different types of Windows OSs as well. Some common ones still in use are Windows 98, Windows 2000, Windows XP, Windows Vista, and Windows Server. Each Windows OS is optimized for different users, hardware configurations, and tasks. For instance, Windows 98 would still run on a brand new PC you might buy today, but it's unlikely Vista would run on PC hardware originally designed to run Windows 98.

There are many more operating systems out there besides the various versions of Windows, and each one is optimized to perform some tasks better than others. Free BSD, Solaris, Linux and Mac **OS** X are some good examples of non-Windows operating systems.

Geeks often install and run more than one OS and a single computer. This is possible with dual-booting or by using a virtual machine. Why? The reasons for this are varied and may include preferring one OS for programming, and another OS for music production, gaming, or accounting work.

An OS must have at least one kind of user interface. Today there are two major kinds of user interfaces in use, the command line interface (**CLI**) and the graphical user interface (**GUI**). Right now you are most likely using a GUI interface, but your system probably also contains a command line interface as well.

Typically speaking, GUIs are intended for general use and CLIs are intended for use by computer engineers and system administrators. Although some engineers only use GUIs and some diehard geeks still use a CLI even to type an email or a letter.

Examples of popular operating systems with GUI interfaces include Windows and Mac OS X. Unixsystems have two popular GUIs as well, known as KDE and Gnome, which run on top of X-Windows. All three of the above mentioned operating systems also have built-in CLI interfaces as well for power users and software engineers. The CLI in Windows is known as MS-DOS. The CLI in Max OS X is known as the Terminal. There are many CLIs for Unix and **Linux** operating systems, but the most popular one is called Bash.

In recent years, more and more features are being included in the basic GUI OS install, including notepads, sound recorders, and even web browsers and games. This is another example of the concept of 'convergence' which we like to mention.

A great example of an up and coming OS is Ubuntu. Ubuntu is a Linux operating system which is totally free, and ships with nearly every application you will ever need already installed. Even a professional quality office suite is included by default. What's more, thousands of free, ready-to-use applications can be downloaded and installed with a few clicks of the mouse. This is a revolutionary feature in an OS

and can save lots of time, not to mention hundreds or even thousands of dollars on a single PC. Not surprisingly, Ubuntu's OS market share is growing very quickly around the world.

As an IT professional, you will probably have to learn and master several, if not all, the popular operating systems. If you think this sort of thing is fun and interesting, then you have definitely chosen the right career ;)

We have learned a little about operating systems in this introduction and you are ready to do more research on your own. The operating system is the lowest software layer that a typical user will deal with every day. That is what makes it special and worth studying in detail.(Adapted fromhttps://www.english4it.com/module/core/unit/3/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1.What is the main function of the kernel in an operating system?

a) To run applications such as games and word processors.

b) To control the file system and device drivers.

c) To provide a graphical user interface (GUI).

d) To install and upgrade hardware components.

2. Why might someone use dual-booting or a virtual machine to run multiple operating systems?

a) To avoid paying for a new computer.

b) To perform different tasks optimized for specific operating systems.

c) To create backup copies of their operating systems.

d) To run an OS that doesn’t support a GUI.

3. What is one key feature of Ubuntu as an operating system?

a) It requires a paid subscription for updates.

b) It includes a CLI interface called MS-DOS.

c) It comes pre-installed with a wide range of applications.

d) It cannot run on modern hardware.

Exercise 5. Translate this text into English.

Операційна система (ОС) – це базове програмне забезпечення, яке керує апаратними ресурсами комп’ютера та забезпечує роботу інших програм. Вона виступає посередником між користувачем і комп’ютерним обладнанням, забезпечуючи зручний інтерфейс для виконання завдань.

Основні функції операційної системи:

1. **Керування ресурсами**: ОС розподіляє ресурси комп’ютера, такі як процесор, оперативна пам’ять і місце на диску, між різними програмами.
2. **Керування файлами**: Забезпечує створення, зберігання, редагування та видалення файлів. ОС також організовує дані в ієрархічну структуру.
3. **Керування пристроями**: ОС взаємодіє з периферійними пристроями, такими як клавіатура, миша, принтер тощо, через драйвери.
4. **Інтерфейс користувача**: ОС забезпечує спосіб взаємодії користувача з комп’ютером через графічний (GUI) або текстовий (CLI) інтерфейс.

Основні типи операційних систем:

1. **Настільні ОС**: Найбільш поширені системи для персональних комп’ютерів, такі як Microsoft Windows, macOS та Linux.
2. **Мобільні ОС**: Системи для смартфонів і планшетів, наприклад, Android та iOS.
3. **Серверні ОС**: Спеціалізовані системи для управління серверними ресурсами, наприклад, Windows Server, Ubuntu Server.
4. **Вбудовані ОС**: Використовуються в пристроях, таких як телевізори, автомобілі чи побутова техніка.

Популярні операційні системи:

- **Windows** – одна з найпоширеніших ОС у світі, відома своєю зручністю для звичайних користувачів.
- **macOS** – операційна система для комп’ютерів Apple, популярна серед дизайнерів та розробників.
- **Linux** – відкрита система з багатьма варіантами (дистрибутивами), яка використовується для серверів та ентузіастів.

- **Android** – найпоширеніша мобільна ОС, заснована на Linux.
- **iOS** – операційна система для мобільних пристроїв Apple.

Значення операційних систем:

Операційні системи забезпечують основу для роботи всіх сучасних пристроїв. Вони роблять техніку доступною для звичайних користувачів, автоматизуючи складні процеси та оптимізуючи використання ресурсів. Знання роботи ОС є важливим для всіх, хто працює з комп'ютерами та іншими цифровими пристроями.

Writing Activity

Exercise 6. Pick a question and write about it.

- Have you ever written or modified any software? If so, what were the challenges you faced? If not, why not?
- Name three pieces of software you use frequently. Why do you use them? What would you change about them?
- Pretend you are the world's best programmer and can write computer code as fast as you can think. What kind of software would you write?

Speaking Activity

Exercise 7. Discuss the questions.

- What are the two popular operating system? Which of them do you prefer more and why?
- Name 3 types of hardware devices. Which do you use most frequently and what for?
- What is the relationship between operating systems and computer hardware?
- What inconveniences that a user can face while interacting with a computer system, which is without an operating system?
- What is the future of operating systems? How will it change over the years?

Unit 3. Software Applications

Exercise 1. Think and answer:

What is your favourite software application? Tell about it.

Exercise 2. Study the list of terms and examples of their usage below:

Agile (гнучкий метод розробки) - an iterative and incremental software development approach with work divided into sprints of a predetermined length.

The first book on Agile methodologies is the Manifesto for Agile Software Development written in 2001.

alpha (альфа-версія) - the first version of a software application that is «feature ready» but still very far from «production ready».

Alpha versions are rarely released to the public and frequently contain serious bugs.

Beta (бета-версія) - a software version which is feature-ready, has passed early testing, and ready for more widespread testing.

The beta version of the new program was released yesterday, but final release is still months away.

freemium (фріміум-модель) - functioning software distributed free of charge with the possibility to buy more features later if desired.

The freemium model is increasingly seen because people are very reluctant to buy something without testing it first.

help file (файл довідки) - electronic documentation included with a program.

The help file often contains useful program shortcuts.

MVP (minimum viable product) (мінімально життєздатний продукт) - an early release with only the most important features included.

MVP releases allow a company to get to market quickly by maintaining focus on the core feature set without wasting time on extra features that add little value to the average user.

Patch (патч) - a software update intended primarily to fix bugs or security holes in a software release.

Software patches and other updates should be available on a software web site.

perform a meaningful task (виконувати значущу задачу) - do something useful as opposed to waste time.

Without a CPU a computer is unable to perform any meaningful task.

Productive (продуктивний) - able to deliver a high number of something efficiently.

Office suites such as MS Office or Google Docs are designed to make users more productive at their jobs.

RC (release candidate) (реліз-кандидат) - a piece of software that has passed its testing stages and is close to release.

Apple refers internally to a release candidate as «the golden master.»

rest at ease (заспокоїтися) - to be free from worry.

«Rest at ease, my boy,» said the salesman, «this new graphics card will have you owning alien worlds for years to come.»

SDLC (software development life cycle) (життєвий цикл розробки ПЗ) - a formalized approach to creating and maintaining software.

The SDLC consists of: requirements gathering, design, programming, testing. deployment, and maintenance.

trial version (пробна версія) - a piece of software which is available for evaluation by customers free of charge, normally for a limited amount of time.

Much to the annoyance of potential customers, sometimes a company «cripples» their trial versions by offering less features than the full version.

Upgrade (оновити) - to replace an older version of software or hardware with a newer version.

The man needed to upgrade his office suite so he could export his document files as XML and JSON.

Waterfall (водоспадна модель) - a top down approach to software development with everything decided up front with milestones and distant deadlines.

Waterfall is best suited for projects where there is a clear goal and requirements are unlikely to change.

you should be wary (потрібно бути обережним) - be careful because something might be dangerous or disastrous.

You should be wary of bugs in beta versions and not use them on production servers.

Shareware (умовно-безкоштовне ПЗ) **-** closed source software distributed freely for a limited time until a full version must be purchased.

Many companies have a shareware or «trial version» of their software in order to gain new users.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

The Software Development Cycle

Without software applications, it would be very hard to actually perform any meaningful task on a computer unless one was a very talented, fast, and patient programmer. Applications are meant to make users more **productive** and get work done faster. Their goal should be flexibility, efficiency, and user-friendliness.

Today there are thousands of applications for almost every purpose, from writing letters to playing games. Producing software is no longer the lonely profession it once was, with a few random geeks hacking away in the middle of the night. Software is a big business and the development cycle goes through certain stages and versions before it is released.

Applications are released in different versions, including **alpha** versions, **beta** versions, release candidates, trial versions, full versions, and **upgrade** versions. Even an application's instructions are often included in the form of another application called a help file.

Alpha versions of software are normally not released to the public and have known bugs. They are often seen internally as a 'proof of concept'. Avoid alphas unless you are desperate or else being paid as a 'tester'.

Beta versions, sometimes just called 'betas' for short, are a little better. It is common practice nowadays for companies to release public beta versions of software in order to get free, real-world testing and feedback. Betas are very popular and can be downloaded all over the Internet, normally for free. In general **you should be wary** of beta versions, especially if program stability is important to you. There are exceptions

to this rule as well. For instance, Google has a history of excellent beta versions which are more stable than most company's releases.

After the beta stage of software development comes the release candidates (abbreviated RC). There can be one or more of these candidates, and they are normally called RC 1, RC 2, RC 3, etc. The release candidate is very close to what will actually go out as a feature complete 'release'.

The final stage is a 'release'. The release is the real program that you buy in a shop or download. Because of the complexity in writing PC software, it is likely that bugs will still find their way into the final release. For this reason, software companies will offer **patches** to fix any major problems that end users complain loudly about.

Applications are distributed in many ways today. In the past most software has been bought in stores in versions called retail boxes. More and more, software is being distributed over the Internet, as open source, shareware, freeware, or traditional proprietary and upgrade versions. (Adapted from https://www.english4it.com/module/core/unit/4/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1.What is the primary purpose of software applications?

a) To replace the need for programmers.

b) To make tasks more efficient and user-friendly.

c) To test computer hardware for bugs.

d) To ensure operating system compatibility.

2.What is a key characteristic of beta versions of software?

a) They are feature complete and free of bugs.

b) They are often distributed to the public for testing and feedback.

c) They are only available for purchase in retail stores.

d) They are the final version of the software.

3.Why do software companies release patches after the final release?

a) To add new features to the software.

b) To fix bugs that users report after release.

c) To downgrade the software for older systems.

d) To convert beta versions into alpha versions.

Exercise 5. Translate this text into English.

Цикл розробки програмного забезпечення – це процес створення, тестування та вдосконалення програм, які забезпечують користувачам зручний інструмент для виконання різноманітних завдань. Основна мета програм – підвищення продуктивності, ефективності та зручності використання.

Сьогодні існують тисячі програм для різних цілей: від написання листів до створення відеоігор. Розробка програмного забезпечення стала масштабною галуззю, і кожна програма проходить кілька стадій перед тим, як з’явитися на ринку.

Основні етапи випуску програмного забезпечення:

1. **Альфа-версія**

Альфа-версії зазвичай не доступні для широкого загалу і використовуються всередині компанії для перевірки концепції. Вони часто містять багато помилок і призначені для внутрішнього тестування.

2. **Бета-версія**

Бета-версії розповсюджуються для публічного тестування, щоб зібрати відгуки від реальних користувачів. Хоча ці версії зазвичай мають менше помилок, ніж альфа, вони все одно можуть бути нестабільними. Наприклад, компанії на кшталт Google часто випускають стабільні бета-версії, які працюють краще за деякі релізи інших виробників.

3. **Кандидат на реліз**

Це майже завершена версія програми, яка проходить фінальне тестування. Її можуть позначати як RC 1, RC 2 тощо.

4. **Реліз**

Це остаточна версія програми, яку користувачі купують або завантажують. Однак навіть після релізу програмне забезпечення може містити помилки. Для їх виправлення компанії випускають патчі та оновлення.

Способи розповсюдження програмного забезпечення:

Програми можна придбати в магазинах у вигляді коробкових версій або завантажити через Інтернет. Також доступні різні моделі розповсюдження, зокрема:

- відкрите ПЗ
- умовно-безкоштовне ПЗ
- безкоштовне ПЗ
- власницьке ПЗ

Цикл розробки програмного забезпечення дозволяє створювати продукти, які відповідають потребам користувачів і забезпечують їх ефективну роботу. Вивчення цього процесу є важливою частиною професійної підготовки в IT-сфері.

Writing Activity

Exercise 6. Pick a question and write about it.

1. Describe an app or a program you use on your cellphone or computer.

- what the app or program is
- how you found it
- how you use it
- and how you feel about it.

2. Read the quote and try to write your answer on the question below.

“Verification: Are we building the product right? Validation: Are we building the right product?” – Boehm 1981

The question: What is verification and validation?

Speaking Activity

Exercise 7. Discuss the questions.

- What is the main difference between a computer program and computer software?
- Why is software development life cycle important?
- What are SDLC models available? What are their pros and cons?

Unit 4. IT Careers

Exercise 1. Think and answer:

Do you know any people whose IT careers are mind-blowing? Tell about them.

Exercise 2. Study the list of terms and examples of their usage below:

BA (business analyst) (бізнес-аналітик) - a position responsible for the interpretation of business rules and delivering them to technical teams.

A business analyst spends a lot of time gathering relevant business requirements in the initial stages of an IT project.

CIO (chief information officer) (директор з інформаційних технологій) - an executive position responsible for internal processes and practices; normally has a stronger business administration background than technical background.

The CIO's work is often quite varied, ranging from managing a company's IT budget to overseeing staff outsourcing.

CTO (chief technical officer) (технічний директор) - an executive position responsible for all scientific and technological issues related to a company; normally has a strong technical and somewhat less strong management background.

The CEO decided to fire the CTO after the company's IT department lost a year of financial records due to poor backup procedures.

DBA (database administrator) (адміністратор баз даних) - a person in charge of managing and maintaining relational databases and access rights.

The requirements for the DBA job position were: advanced SQL, performance tuning, and disaster recover for ORACLE systems.

database developer (розробник баз даних) - a position responsible for programming and optimizing databases.

The database developer spent three months refactoring the company database to obtain 3NF (third normal form).

enterprise architect (архітектор підприємства) - a high-level position responsible for understanding a business's overall needs and then designing an IT structure to support it.

The firm hired an enterprise architect to oversee the development of the new software platform.

graphic designer (графічний дизайнер) - a position responsible for the creation of images, typography, mock-ups for an organizations' front end systems; normally works under a company Art Director.

The graphic designer had an iMac with a 27» screen, a graphics tablet instead of a mouse, and the latest version of Adobe Illustrator and Photoshop software.

information architect (інформаційний архітектор) - a position responsible for making complex data structures easy to understand and navigate

The information architect came prepared to the meeting with some wireframes and a glossary of target vocabulary for the project.

IT manager (менеджер з IT) - a job position acting as a bridge between upper management and IT; one who encourages personal development in IT staff; the boss of an IT worker

The IT Manager said the new junior developer's attitude was unacceptable.

IT security manager (менеджер з безпеки IT) - a position responsible for setting best practices for securing wi-fi networks, servers, backups, laptops, and VPNs.

The IT security manager ran a password cracker on his own network users to sniff out weak passwords.

IT support engineer (інженер технічної підтримки) - a position responsible for on-demand support for end users including: fixing hardware, installing software, and troubleshooting minor network issues.

An IT support engineer must be able to troubleshoot and fix almost any hardware, software, or network problem that can affect an end user's PC.

network administrator (адміністратор мережі) - a position responsible for maintenance of all aspects of a computer network; often a specialist in TCP/IP, Linux, and related routing technology such as Cisco.

«The Network is down» is a phrase a good network administrator never wants to hear.

PM (project manager) (менеджер проєктів) **-** a position responsible for organizing and delivering a project on time and on budget; often acts a bridge between developers and stake-holders**.**

Software project managers are increasingly turning to Scrum and other Agile practices to get good results from their teams.

QA manager (quality assurance manager) (менеджер з забезпечення якості) **-** a job title whose responsibilities include ensuring appropriate performance for a software project and organizing and instructing testers.

The QA manager organized a massive usability test to try to squash bugs before the software's release date.

software developer (розробник програмного забезпечення) **-** a position responsible for gathering information around a programming task and performing it.

Software developers often specialize in a specific software framework or paradigm such as Java applications, Python, or CSS.

software tester (тестувальник програмного забезпечення) **-** a job title whose responsibilities include ensuring that a software project meets established quality guidelines.

The software tester spent all day documenting a nasty bug in the ERP software.

technical writer (технічний письменник) **-** a position responsible for the creation and maintenance of documentation relating to an IT project including online help, user guides, white papers, and design specifications.

The technical writer wanted to write novels when she was young, but now she is documenting accounting software applications for IBM.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Choosing an IT Career Path

Most people on English4IT.com are either studying for their first job in IT, or else trying to improve their current IT career. If this is the case with you, well then this unit should really help. Sometimes the hardest part of meeting a goal is to properly define what you are trying to accomplish in the first place. In this article we will discuss the

top IT job positions available around the world right now. So read the rest of the article, reflect on which career most suits your personality... and then go for it!

There are several things to keep in mind when determining what field of IT to go into. Keep an eye on job web sites such as DICE.com or Monster.com to see which jobs are most in-demand. Keep in mind that for many jobs described below, there are several levels of positions available. For instance there are «junior», «senior», and «lead» **software developer** positions available. You probably can't start out you career as a lead developer. You have to know your own limits.

Be honest with yourself. If you don't have previous experience, good contacts, or a good degree from a well-known university, you will be more successful in getting a lower-level job. Also, find out what the job you are applying for typically pays in your area. If you are young, living in a financially depressed area, or really need a job, keep your salary expectations a bit lower than the average. This will make your chances much higher than normal to get hired. Once you have «job experience» then you will be in a good position to ask for more money. Sometimes the best way to get more money is to quit your job and work for another similar company. This may sound cruel or thankless, but that is how business works in the real world.

Everyone who works hard deserves a raise every year. How do you show your IT manager that you are a good performer? Easy. Show up on time, be dependable, be active in the meetings, and always do a little bit more than is asked of you. Also equally important is to be well-liked by members of your team. Read on for more details...

Learn something new every day

IT is an area where people are judged largely by how much they know. If money and a high job position are important to you, you can quickly raise your level by telling your manager that you want harder tasks and more responsibility. IT Managers normally love it when employees ask for more responsibility. When you meet with your manager, set goals for yourself and meet or exceed those goals. Here are some things you can do to increase your worth to your company:

1. learn a new programming language

2. take a certification such a Microsoft, Linux Professional Institute, or Cisco

3. study to be a ScrumMaster or another type of project manager.

Meeting set goals can have beneficial results when it's time to renegotiate salaries, survive a round of layoffs, or get a promotion.

Appearance and attitude is very important!

Take an active interest in things outside IT: such as sports, politics, music, and film. This will make socializing at company events easier for you. If you are disliked in the company then you will not get promotions or important projects.

Be courteous, helpful, and respectful to others

In my own career, I have been in some good IT departments and some bad ones. In a good IT department, the engineers are known for sharing knowledge and helping each other. In bad IT departments, the engineers are secretive and hide knowledge. How can everyone get better if some people are selfish with what they know? Information wants to be free. You must set it free. Despite the fact that I have been to several universities, graduate school, and have collected many IT certifications, I have still learned much more about IT from my fellow engineers than from all my higher education combined. So my advice is to be kind and respect your fellow IT staff. They are your family for eight hours every day, forty hours every week!

When you start a new job, realize how some people are nice to you and some people ignore you. Which kind of person do you want to be? When you get a new junior team member, try to help them and include them in decisions. Make sure they have someone to eat lunch with. If you party after work with your co-workers, invite new employees with you. Being nice to new people can have many rewards, both emotionally and financially.

Back in the 90's we used to have a saying, «Think globally, act locally.» What this means is that just by being nice and pleasant yourself, you can make the whole world a nicer and more pleasant place as well.

Have your own mind and your own opinions

State your opinions in meetings and give good reasons and facts to back up your opinions. But don't be stubborn or insistent if things don't go your way. And whatever you do, please don't be passive-aggressive! Passive-aggressive behavior is when you think something bad about a person or an idea, and then you talk badly behind someone's back (when that person is not around). This is very destructive behavior to both yourself and your IT department.

Okay, now we are ready now to investigate some popular IT job positions. I will rate the following jobs based on the following criteria: respect, qualities, salary, dress, and fun factor. I will also include some notes. These are subjective opinions. Some are even intentionally funny. If you disagree with me, please feel free to flame me.

CTO (Chief Technical Officer), CIO (Chief Information Officer)

Respect: Very High

Qualities: Business savvy, technical mindset, good people skills

Average Salary: $150,000

Dress: Business suit and very clean

Fun Factor: Only fun if you are a workaholic or on a power trip.

Notes: These jobs are highly competitive and usually political, so your chances are low. Sorry.

Enterprise Architect

Respect: High

Qualities: Good technical, business, and design skills

Salary: $100,000

Dress: Clean and presentable with collared shirt and pants

Fun Factor: Fun job because you get to talk to all other departments

Notes: Responsible for all solutions that work; not responsible for ones that don't work

IT Manager

Respect: Medium-High

Qualities: Detail oriented, punctual, critical, supportive

Salary: $70,000

Dress: Business Casual

Fun Factor: Can be fun but often very stressed

Notes: They always seem to be working

Technical Writer

Respect: Medium

Qualities: Excellent writing skills, good technical mind

Salary: $50,000

Dress: Business Casual

Fun Factor: Writers are often good at telling stories

Graphic Designer

Respect: Low-Medium

Qualities: Excellent drawing and illustration skills, good color matching and artistic qualities

Salary: $50,000

Dress: Casual

Fun Factor: Generally fun people and sometimes a bit moody

Notes: Not as 'square' as the rest of the IT department. All good designers seem to have tattoos, piercings, and a fashion sense.

Software Developer

Respect: Medium

Qualities: Creative, persistent, insatiable thirst for knowledge

Salary: $70,000

Dress: Casual Dress is normally the rule (t-shirt and jeans)

Fun Factor: If you don't have fun being a developer then you have the wrong job; other people might not understand your sense of humor though ;)

Notes: Companies have a lot of developers compared to other positions listed. Therefore your chances of becoming a developer are good if you have the skills and more importantly the desire.

Project Manager

Respect: Medium

Qualities: Cooperation, leadership, and organization skills

Salary: $60,000

Dress: Business Casual (collared shirt and nice jeans or pants)

Fun factor: This tends to be a high stress position with long hours. If that sounds fun then go for it!

Database Developer / Database Administrator

Respect: Medium-High

Qualities: Detail-oriented, high business knowledge

Salary: $80,000

Dress: Business Casual

Fun Factor: If processing giant data sets excite you, then this job is for you.

IT Security Manager

Respect: High

Qualities: Military outlook on life, defensive, pro-active

Salary: $70,000

Dress: Smart, clean dress is very important

Fun Factor: Are you kidding me? This guy is basically a cop!

System Administrator

Respect: Medium-High

Qualities: God complex, often eat fast food and drink a lot of soda

Salary: $75,000

Dress: If they were allowed to, they would probably dress as World of Warcraft characters!

Fun Factor: Sysadmins can be patronizing; but they can be fun as well, especially after they have a few beers. They are often eager to show others that they are just «normal people». But this is not usually true.

Notes: Never anger a sysadmin! Why? They have access to everything in the company.

Software Tester

Respect: Low-Medium

Qualities: Detail-oriented, persistent, curious

Salary: $40,000

Dress: Same as developers

Fun Factor: They are normally seen hanging out with developers, trying to talk about bugs.

Notes: Testers play a vital role in software development that cannot be understated. If you don't have a single tester on your team, you are probably in trouble.

IT Support Engineer

Respect: Low (except when someone needs help fixing their computer, then it's really high)

Qualities: Must be good at dealing with technically incompetent people

Salary: $35,000

Dress: Casual

Fun Factor: Often an endless source of funny stories about technically incompetent end-users

Notes: Do not kill the end-users! (Adapted from https://www.english4it.com/module/core/unit/5)

Exercise 4. Pick the correct answer from a list according to the text above.

1. What is a common strategy for increasing your worth in an IT career?

A. Demanding a raise every month

B. Taking on harder tasks and more responsibility

C. Avoiding additional certifications

D. Being passive-aggressive during meetings

2. Which IT position is described as requiring artistic qualities and often involves casual dress?

A. Software Developer

B. Graphic Designer

C. System Administrator

D. Database Developer

3. What is a key quality of an IT Security Manager?

A. Excellent drawing skills

B. Proactive and defensive mindset

C. Detail-oriented and curious

D. An insatiable thirst for knowledge

Exercise 5. Translate this text into English.

Сфера інформаційних технологій (IT) пропонує безліч різноманітних професій, які охоплюють розробку програмного забезпечення, адміністрування систем, кібербезпеку, проектний менеджмент та інші напрямки. Кожна з цих професій має свої особливості, вимагає певних навичок і пропонує унікальні можливості для кар’єрного зростання.

1. Розробник програмного забезпечення

Ця професія включає створення комп’ютерних програм, додатків і систем. Розробники повинні володіти знаннями мов програмування, таких як Python, Java, або C++. Вони працюють над написанням коду, тестуванням, усуненням помилок і впровадженням нових функцій.

2. Системний адміністратор

Сисадміни відповідають за налаштування, обслуговування та захист серверів і комп'ютерних систем. Вони часто вирішують технічні проблеми, що виникають у мережі, і забезпечують стабільну роботу IT-інфраструктури компанії.

3. Менеджер проєктів

Проєктний менеджер організовує роботу команди, слідкує за дотриманням термінів і бюджету, а також забезпечує якісну реалізацію проєкту. Для цієї професії важливі лідерські якості, вміння планувати і комунікувати з різними відділами.

4. Тестувальник програмного забезпечення

Тестувальники перевіряють програмне забезпечення на помилки і оцінюють його зручність для користувачів. Ця робота вимагає уваги до деталей і здатності мислити критично, щоб знаходити можливі проблеми у коді.

5. Фахівець з кібербезпеки

Ці експерти відповідають за захист даних і систем від кіберзагроз. Вони працюють над виявленням уразливостей, розробкою систем захисту та реагуванням на інциденти.

6. Графічний дизайнер

Ця професія поєднує творчість і технології. Графічні дизайнери створюють візуальний контент для вебсайтів, мобільних додатків, реклами та інших продуктів.

7. Інженер технічної підтримки

Фахівці техпідтримки допомагають користувачам вирішувати проблеми з обладнанням і програмним забезпеченням. Вони часто працюють з людьми, які не мають технічних знань, тому важливими є терпіння і комунікабельність.

8. Аналітик даних

Аналітики даних займаються збором, аналізом і інтерпретацією великих обсягів інформації для прийняття бізнес-рішень. Вони працюють з базами даних, створюють звіти та знаходять приховані закономірності.

Отже, IT-сфера є однією з найперспективніших галузей у світі, яка пропонує безліч можливостей для кар'єри. Кожна професія має свої переваги та вимагає різних навичок, тому кожен може знайти те, що підходить саме йому.

Writing Activity

Exercise 6. Research three more IT job positions not from reading in this unit.

Write a short description about the roles and responsibilities.

Speaking Activity

Exercise 7. Discuss the questions.

- Which of the IT careers would you like to pursue? Why?
- Which factors can influence your IT career choice?
- Can IT careers allow you to make the world a better place?
- What kind of personal and professional skills can land you an it job?
- What are the best skills you can self-study that will land you a remote job?

Unit 5. Programming Languages

Exercise 1. Think and answer:

Have you ever tried to learn any programming languages? If no? which one would you try to learn?

Exercise 2. Study the list of terms and examples of their usage below:

NET Framework фреймворк .NET - a software framework by Microsoft which executes code via a virtual machine

The .NET Framework is Microsoft's premium solution for programming applications, and supports over 20 languages including C#, VB.NET, and IronRuby.

Compiler (компілятор) **-** a program that takes human readable code and turns it into machine readable code for running at a later time

The students liked programming python more than C because they didn't have to use a compiler before execution.

ECMAScript (екмаскрипт) **-** the official specification or «standard» for the language commonly known as JavaScript

The professor told us that officially JavaScript should be called ECMAScript, but that few people do so in practice.

Elegant code (елегантний код) **-** concise, clean, and clear code which allows other developers to understand and extend it

The junior programmer wrote elegant code and used variable names with a clear meaning.

Interpreter (інтерпретатор) **-** a program that reads a high-level programming language, converts it into machine code, and then immediately runs that code

Computer languages that require an interpreter often run slower than languages that require a compiler.

Java (мова програмування Java) **-** a high-level, compiled, object-oriented programming language owned by software giant Oracle

The computer science professor predicted that as processor speeds increased, Java would eventually replace C++ in application development.

JavaScript (мова програмування JavaScript) - a popular programming language originally developed by Brendan Eich at Netscape to provide client-side interactivity in Web pages

The web programmer used JavaScript to alert the user about invalid data entered in a required field.

Multi-paradigm language (мультипарадигмальна мова) - a programming language that supports both procedural and object-oriented programming philosophies

PHP started out as a procedural language, but grew into a multi-paradigm language when it added support for objects in version 4.

Object-oriented language (об'єктно-орієнтована мова) - any programming language optimized for modeling real-world objects and concepts

Java is the most popular object-oriented language.

Perl (мова програмування Perl) - a high-level, interpreted programming language written by Larry Wall in 1986 and typically used for system administration

Larry Wall is the inventor of Perl, a widely used programming language that has a very devout following.

PHP (PHP Hypertext Preprocessor) (PHP препроцесор гіпертексту) - a high-level, interpreted programming language written by Rasmus Lerdorf in 1995 and aimed mainly at web developers creating dynamic applications

The professor said that PHP is the most popular web scripting language in the world and supports many advanced object-oriented programming techniques.

Portability (портативність) - a measure of how easily programs can be moved to a new system without having to make any changes

Java and ANSI C are two attempts at making software portability a priority.

Procedural language (процедурна мова) - any programming language that is based on a step-by-step approach to solving a problem

C is the standard example of a procedural language.

Python (мова програмування Python) - a high-level, interpreted programming language developed by Guido van Rossum at CWI in the Netherlands

The motto of Python is «there should be one- and preferably only one- obvious way to do it».

Ruby (мова програмування Ruby) - an object-oriented, high-level, interpreted programming language developed in the 1990's by ace Japanese programmer Yukihiro Matsumoto

The creator of Ruby wanted a scripting language that was more elegant than Perl, and more object-oriented than Python.

Syntax (синтаксис) - rules governing the structure of a programming language

The basic concepts of all programming languages are quite similar, even if the syntax can be quite different.

VB (Visual Basic) (візуальний бейсік) - a Windows-only, multi-paradigm programming language developed by Microsoft and intended for beginners or casual use

The high school programming teacher taught his students Visual Basic in his Introduction to Programming class.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Top 10 Most Popular Programming Languages

There are hundreds of programming languages in use today. How can you know which one to learn first? How do you know which ones are the best for your IT field of choice? Well, I can't answer that question for you. But why not start by learning one of the top 10 most popular ones? That way you will always be able to get a job in the IT industry.

Learning a programming language is not easy, but it can be very rewarding. You will have a lot of questions at first. Just remember to get help when you need it! You can find out the answer to almost everything on Google nowadays.... so there is no

excuse for failure. Also remember that it takes years to become an expert programmer. Don't expect to get good overnight. Just keep learning something new every day and eventually you will be competent enough to get the job done ;)

This article covers the top 10 most popular programming languages as ranked by Tiobe.com in June 2009. I have added some general reviews and comments about each language they listed. Remember these are my own personal opinions. Other IT professionals might have different opinions.

1. **Java.** Java uses a compiler, and is an object-oriented language released in 1995 by Sun Microsystems. Java is the number one programming language today for many reasons. First, it is a well-organized language with a strong library of reusable software components. Second, programs written in Java can run on many different computer architectures and operating systems because of the use of the JVM (Java virtual machine). Sometimes this is referred to as code portability or even WORA (write once, run anywhere). Third, Java is the language most likely to be taught in university computer science classes. A lot of computer science theory books written in the past decade use Java in the code examples. So learning Java syntax is a good idea even if you never actually code in it. Java Strengths: WORA, popularityJava Weaknesses: Slower than natively compiled languages

2. CC is a compiled, procedural language developed in 1972 by Dennis Ritchie for use in the UNIX operating system. Although designed to be portable in nature, C programs must be specifically compiled for computers with different architectures and operating systems. This helps make them lightning fast. Although C is a relatively old language, it is still widely used for system programming, writing other programming languages, and in embedded systems. Strengths: SpeedWeaknesses: Memory management can be difficult to master

3. C++. C++ is a compiled, multi-paradigm language written as an update to C in 1979 by Bjarne Stroustrup. It attempts to be backwards-compatible with C and brings object-orientation, which helps in larger projects. Despite it's age, C++ is used to create a wide array of applications from games to office suites. Strengths:

SpeedWeaknesses: C++ is older and considered more clumsy than newer object-oriented languages such as Java or C#.

4. PHP. PHP uses a run-time **interpreter**, and is a multi-paradigm language originally developed in 1996 by Rasmus Lerdorf to create dynamic web pages. At first it was not even a real programming language, but over time it eventually grew into a fully featured object-oriented programming language. Although PHP has been much criticized in the past for being a bit sloppy and insecure, it's been pretty good since version 5 came out in 2004. It's hard to argue with success. Today, PHP is the most popular language used to write web applications. Even English 4 IT, the program you are currently using, is written in PHP ;) Strengths: Web programming, good documentation Weaknesses: Inconsistent syntax, too many ways to do the same thing, a history of bizarre security decisions

5. **VB** (or Visual Basic) Visual Basic is an interpreted, multi-paradigm language developed by Microsoft Corporation for the Windows platform. It has been evolving over the years and is seen as a direct descendant of Microsoft's old BASIC from the 1970's. Visual Basic is a good language for scripting Windows applications that do not need the power and speed of C#. Strengths: None. Weaknesses: Only runs in Windows

6. **Python**. Python is an interpreted, multi-paradigm programming language written by Guido van Rossum in the late 1980's and intended for general programming purposes. Python was not named after the snake but actually after the Monty Python comedy group. Python is characterized by its use of indentation for readability, and its encouragement for **elegant code** by making developers do similar things in similar ways.

Python is used as the main programming choice of both Google and Ubuntu. Strengths: Excellent readability and overall philosophy Weaknesses: None

7 C#. C# is a compiled, object-oriented language written by Microsoft. It is an open specification, but rarely seen on any non-Windows platform. C# was conceived as Microsoft's premium language in its .NET Framework. It is very similar to Java in

both syntax and nature. Strengths: Powerful and pretty fast Weaknesses: Only really suitable for Windows

8. JavaScript. JavaScript is an interpreted, multi-paradigm language. A very strange one too. Despite it's name, it has nothing whatsoever to do with Java. You will rarely, if ever, see this language outside of a web browser. It is basically a language meant to script behaviors in web browsers and used for things such as web form validation and AJAX style web applications. The trend in the future seems to be building more and more complex applications in JavaScript, even simple online games and office suites. The success of this trend will depend upon advancements in the speed of a browser's JavaScript interpreter. If you want to be correct, the real name of this programming language is ECMAscript, although almost nobody actually calls it this. Strengths: it's the only reliable way to do client-side web programming Weaknesses: it's only really useful in a web browser

9. **Perl.** Perl is an interpreted, multi-paradigm language written by Larry Wall in 1986. It is characterized by a somewhat disorganized and scary-looking syntax which only makes sense to other PERL programmers ;) However, a lot of veteran programmers love it and use if every day as their primary language. 10 years ago, Perl was more popular than it is today. What happened? A lot of newer programmers and even old Perl programmers (such as myself) have switched to other languages such as PHP, Python, and Ruby. Perl is perhaps still the best language for text processing and system administration scripting. I personally do not recommend it however as a primary programming language. Strengths: text processing and system administration Weaknesses: strange syntax, and perhaps too many ways to do the same thing

10. **Ruby**. Ruby is an interpreted, object-oriented language written by Yukihiro Matsumoto around 1995. It is one of the most object-oriented languages in the world. Everything is an object in Ruby, even letters and numbers can have method calls. It's a great language to learn if you love objects. The only negative is that it's love of object-orientation makes it a bit slow, even for an interpreted language. Strengths: Perhaps

the world's most object-oriented language Weaknesses: its superior object model comes at a price... namely speed

Okay! Those are the top 10 programming languages in use today and some personal comments about them. Remember that opinions are like noses, everyone has one and they all smell ;) If you disagree, please feel free to email me or write your own opinions on the forum. (Adapted from https://www.english4it.com/module/core/unit/9/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1. Which programming language is known for its «write once, run anywhere» capability thanks to the Java Virtual Machine (JVM)?

a) C++

b) Java

c) Python

d) JavaScript

2. What is a key weakness of C# as a programming language?

a) It has a complex syntax.

b) It is only really suitable for Windows platforms.

c) It lacks object-oriented features.

d) It has poor readability.

3. Which programming language is particularly well-suited for text processing and system administration, despite having a complex syntax?

a) Ruby

b) Perl

c) PHP

d) Visual Basic

Exercise 5. Translate this text into English.

Python – це одна з найпопулярніших мов програмування у світі, яку створив Гвідо ван Россум наприкінці 1980-х років. Ця мова розроблялася з метою створення простої, зрозумілої та зручної у використанні платформи для

програмістів. Python отримав свою назву не від змії, а на честь комедійної групи «Монті Пайтон».

Основні переваги Python:

1. **Зрозумілий синтаксис**: Програми на Python легко читати та писати завдяки використанню відступів і чітких правил оформлення коду.
2. **Універсальність**: Python застосовується в багатьох галузях, включаючи веб-розробку, науку про дані, машинне навчання, штучний інтелект, автоматизацію процесів, розробку ігор тощо.
3. **Велика кількість бібліотек**: Python має багату екосистему бібліотек, що дозволяє швидко реалізовувати складні проєкти без необхідності писати все з нуля.
4. **Кросплатформеність**: Програми, написані на Python, можуть працювати на різних операційних системах, таких як Windows, macOS та Linux.

Де використовується Python?

- **Google та YouTube**: Ці гіганти використовують Python для розробки частини своїх систем.
- **Наукові дослідження**: Завдяки бібліотекам, таким як NumPy, Pandas і Matplotlib, Python є незамінним інструментом для обробки даних і проведення аналізу.
- **Штучний інтелект**: Фреймворки TensorFlow та PyTorch побудовані на Python, що робить його основною мовою в цій галузі.

Недоліки Python

Попри свої переваги, Python має кілька слабких сторін:

- **Швидкість виконання**: Через інтерпретацію Python може бути повільнішим, ніж компільовані мови, такі як C++.
- **Обмеження в мобільних додатках**: Python менш поширений у розробці для мобільних платформ.

Python – чудовий вибір для тих, хто хоче розпочати свою кар’єру в програмуванні або розширити свої навички. Його гнучкість, зручність і потужність роблять його однією з найкращих мов програмування у світі.

Writing Activity

Exercise 6. Pick a question and write about it.

- Which programming languages do you already know? Have you ever used it?

- Which language do you think is best for making banking software? Web pages? Games? Text processing?

Speaking Activity

Exercise 7. Discuss the questions.

- What programming language would be interesting for you to learn? Why?

- Why Perl isn’t recommended as a primary programming language ?

- Is PHP deservedly considered the most popular language in the world which supports many advanced object-oriented programming techniques?

Unit 6. Computer Types

Exercise 1. Think and answer:

Which types of computers have you had and would like to have?

Exercise 2. Study the list of terms and examples of their usage below:

Barebone (базова конфігурація) - computers which are sold incomplete or in kits that require extra components to be functional

Assembling a barebone computer is usually less expensive and certainly more configurable than buying a ready-made one off the shelf.

Convergence (конвергенція) - the evolution of devices towards common functionality

The iPhone is a good example of convergence, because it has all the features of a PDA, mobile phone, and an MP3 player in one package.

Desktop computer (настільний комп'ютер) - a personal computer typically in the shape of a tower or box with a connected keyboard, mouse, and monitor

Desktop computers are seen in nearly every office and home today.

Embedded system (вбудована система) - a computer using a relatively slow and specialized processor and ROM chip, normally used to control a particular device such as a washing machine or an MP3 player

Nearly all electronics you can buy today are controlled by embedded systems.

Form factor (форм-фактор) - the size, configuration, or physical arrangement of a computing device

It's hard to buy components for a computer if you don't know the particular form factor.

Laptop (ноутбук) - a portable computer with a built-in screen, integrated keyboard, and battery power

As laptop computers have become more powerful and affordable, they are steadily replacing the more traditional desktop computer.

Legacy system (застаріла система) - an older device or application that continues to be used because of the high cost of replacing it

Mainframes and minicomputers used to be cutting edge; now they are known as legacy systems.

PC (personal computer) (персональний комп'ютер) - a computer designed for use by one person at a time

When people use the term 'PC', they are often referring to a desktop computer running Windows.

PDA (personal digital assistant) (кишеньковий комп'ютер) - a legacy handheld computer often running Palm OS or Windows CE and used as a contact organizer, game machine, work tool, or access controller

In the 1990's the modern traveling businessman depended on his PDA to organize all his contacts and appointments.

Print server (друкарський сервер) - hardware or software designed to connect a network device with a printer

The secretary could not get a hard copy because the print server was broken.

Router (маршрутизатор) - a specialized computer which connects two networks

The wifi router allowed the entire office to share a single Internet connection.

Server (сервер) - a type of computer intended primarily for central distribution of data to other computers on the same network

Because servers hold large amounts of centralized data, it is critical to have daily backup routines in place.

Smartphone (смартфон) - a hand-held multimedia computer optimized for communication and featuring a touch screen and internet connection

Almost everyone owns a smart phone today because they serve so many purposes and have become much more affordable over time.

Supercomputer (суперкомп'ютер) - a computer designed to perform intensive tasks such as weather prediction, big data calculations, or space research.

A typical supercomputer may have up to 100 processors and speeds are measured in tens of gigaflops.

Tablet (планшет) - a portable computer shaped in the form of a notebook and capable of advanced handwriting recognition via use of a stylus or on-screen keyboard.

Bill Gates predicted the rise of the tablet PC in 2001, but not much happened until Apple released the first iPad in 2010.

Wearables (носимі пристрої) - computers designed to be worn on the wrist, head, or other places on the body often to improve mobility.

Smart watches and VR goggles are two popular forms of wearables.

Workstation (робоча станція) - a high quality computer, typically with lots of RAM, plenty of CPU power, and a high quality video card

The engineer's latest 3d modeling project was so GPU intensive, that he wouldn't even begin it until his new workstation arrived.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Recognizing Different Types of Computers

In this unit you will learn about different types of computers and what makes them unique.

Computers were not always things you could carry around with you, or even have in your bedroom. Sixty years ago, computers (such as ENIAC) were as big as entire apartments. They were difficult to use and not very powerful by today's standards. They also cost a lot of money to build and operate. So computers were only used by large organizations such as governments, international corporations, and universities.

Throughout the 1950s and 1960s, computers captured the public's imagination in literature, films, and TV. More and more companies wanted computers, even if they didn't always have a good reason to own one. As a result, computers gradually became smaller, cheaper, and more practical to own. This was thanks in part to companies like IBM, which mass-produced computers for the first time and promoted them to medium and large businesses to do things like payroll, accounting, and other number-crunching tasks.

In the 1970s and 1980s a new type of computer started to gain in popularity. It was called the **PC** or personal computer. For the first time in history, computers were now for everyone. The PC started a revolution which affects nearly everything we do today. The ways we work, play, communicate, and access information have all been radically reshaped due to the invention and evolution of the PC.

PCs are everywhere you look today. At home, at the office, and everywhere in between. Many people still mistakenly believe the term PC is synonymous with a **desktop computer** running Windows. This is not true. Really, any computer you use by yourself for general purposes could be called a PC. You probably already own at least one of these types of PCs:

- **laptop** - desktop computer- **PDA** or personal digital assistant - workstation

Besides PCs, there are other types of computers you probably see at work or school. These include:

- file **servers**- print servers- web servers

But not all types of computers are so obvious as the ones above. There are still other kinds of computers that fit inside of other devices and control them. These computers are known as embedded systems.

Embedded systems can be found in traffic lights, TV sets, refrigerators, coffee machines and many more devices. Embedded systems are typically controlled by inexpensive, specialized processors which can only handle very specific tasks.

Types of computers go in and out of fashion as times changes. Older kinds of computers which were very popular in the 20th century (1900's) are now referred to as legacy systems. These include:

- mainframes- minicomputers- IBM clones

New types of computers are always coming out and replacing or augmenting existing computer types. Examples of new types of computers emerging would be netbooks, **tablet**, and even wearable computers.

As you complete this unit, you will learn to differentiate between different computer types. Keep in mind that the lines between computer types are constantly

being blurred. This phenomenon is known as **convergence**. (Adapted from https://studfile.net/preview/7068858/page:5/)

Exercise 4. Pick the correct answer from a list according to the text above.

1. What was one of the key factors that made computers more accessible to medium and large businesses in the 1950s and 1960s?

a) The invention of the PC

b) The mass production of computers by companies like IBM

c) The development of embedded systems

d) The decrease in computer size and weight

2. Which of the following is an example of an embedded system?

a) Desktop computer

b) Laptop computer

c) Coffee machine

d) File server

3.What term is used to refer to older computers that were once popular but are no longer commonly used today?

a) Personal computers

b) Legacy systems

c) Embedded systems

d) Netbooks

Exercise 5. Translate this text into English.

Комп’ютери бувають різних типів і призначень, і кожен з них має свої унікальні особливості. Знання цих відмінностей допомагає краще зрозуміти, як вони працюють і для чого можуть бути використані.

Персональні комп’ютери

Персональні комп’ютери − це пристрої, призначені для індивідуального використання. До цієї категорії входять:

Настільні комп’ютери: Встановлюються на робочому місці та зазвичай використовуються вдома чи в офісі.

Ноутбуки: Портативні комп’ютери, які легко переносити.

Планшети: Компактні пристрої з сенсорними екранами.

Смартфони: Хоча це телефони, вони також є потужними обчислювальними пристроями.

Сервери

Сервери − це комп’ютери, які виконують спеціальні функції, наприклад:

Файлові сервери: Зберігання та надання доступу до файлів.

Веб-сервери: Обробка веб-сайтів та інтернет-запитів.

Принт-сервери: Організація друку в офісній мережі.

Вбудовані системи

Ці комп’ютери є частиною інших пристроїв і виконують лише одну чи кілька конкретних функцій. Вони зустрічаються у:

Світлофорах

Побутовій техніці (холодильниках, кавоварках)

Електронних пристроях (телевізорах, автомобілях)

Старі системи

До цієї категорії належать комп’ютери, які були популярні в минулому, але сьогодні використовуються рідко. Наприклад:

Мейнфрейми: Великі потужні системи, які обслуговували багато користувачів одночасно.

Мінікомп’ютери: Менші за мейнфрейми, але схожі за функціями.

Нові типи комп’ютерів

Сучасні технології постійно розвиваються, і з’являються нові пристрої, такі як:

Нетбуки: Компактні та недорогі ноутбуки.

Носимі пристрої: Розумні годинники або окуляри з комп’ютерними функціями.

Комп’ютери постійно змінюються і вдосконалюються, тому важливо вміти розрізняти їхні типи та розуміти їхнє призначення.

Writing Activity

Exercise 6. Pick a question and write about it.

- What is meant by the term 'convergence'?
- Why did companies in the 1980's shift away from mainframes and move towards personal computers?

Speaking Activity

Exercise 7. Discuss the questions.

- Is it necessary to have daily backup routines in place?
- Why do many people still mistakenly believe the term PC is synonymous with a desktop computer running Windows?
- What are the advantages of laptop computers?

Unit 7. Components

Exercise 1. Think and answer:

Which components does your computer system consist of?

Exercise 2. Study the list of terms and examples of their usage below:

BIOS (basic input output system) (базова система вводу-виводу) **-** this is normally a ROM program that controls the base functionality of the computer such as video, hard drives, optical drives, and keyboard

The technician needed to upgrade the BIOS before installing a faster processor in the workstation.

Chipset (чипсет) **-** a collection of integrated circuits on the motherboard designed to perform certain tasks such as controlling components and system buses

Speccing a low-end chipset is not a good way to save money when building a new PC.

Coprocessor (співпроцесор) **-** any computer processor or portion of the CPU which assists the main processor with a highly specialized task

Many older microprocessors included a coprocessor to speed up floating-point calculations.

Graphics card (графічна карта) **-** a component of a computer which is designed to convert a binary image stored in memory to a display medium

The latest and greatest games normally require the latest and greatest graphics cards.

GPU (Graphics Processing Unit) (графічний процесор) **-** a specialized co-processor designed to handle graphical calculations such as 3D modeling and games

Most computers come specced with a weak GPU embedded into the motherboard and designed for business applications and simple 2D games.

Microprocessor (мікропроцесор) **-** an electronic device constructed from microscopic transistors on a single integrated circuit

The microprocessor is often thought to be the central brain of a computer because it performs most of the calculations.

Motherboard (материнська плата) - The main circuit board in a computer that carries the system buses, sockets for processors, memory modules, etc

The processor, RAM, and PCI cards plug directly into the motherboard.

NIC (network interface card) (мережева карта) - a wired or increasingly wireless PCI or USB device that connects a computer to a network

Almost every computer that ships today has some sort of NIC so it can attach to a network.

PCI (Peripheral Component Interconnect) (шина розширення) - a standard 32-bit bus running at 132 MB/s

The network card used a PCI bus connector which snapped into the motherboard.

PCI Express (Peripheral Component Interconnect Express) (шина розширення PCI Express) - a newer bus type used mainly for graphic cards and running at speeds up to 16,000 MB/s

The boy grinned ear-to-ear when he received a new PCIe graphics card for Christmas.

Power supply (блок живлення) - a separate unit or part of a circuit that supplies the correct amount of electrical current to a computer system

Nick upgraded to a 600 watt power supply when he added a new graphics card and two more hard drives.

SATA (Serial ATA) - a widely used bus for connecting hard disk drives and similar devices to the motherboard

The SSD storage device came with a SATA connection running at 6 Gbit/s.

Sound card (звукова карта) - a hardware component which allows a computer to play and record audio

The musician installed a dedicated sound card to replace the cheap one that came with his computer.

Spec (specification) (специфікація) - a clear set of technical or quantitative requirements

The lead technician became angry when his assistant bought equipment outside the specification.

USB (Universal Serial Bus) - a fast and convenient computer technology allowing the connection common peripherals such as digital cameras, scanners, external hard drives, etc

USB technology is so fast and easy to use, it is hard to even remember what similar connection types existed before it.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Computer Hardware Components

Due to convergence, the traditional categories we divide computing into are blurring. But for practical reasons, IT professionals can still divide hardware into two main classes: components and peripherals.

Components are primarily core internal devices of a computer which help define what type a computer is, what it is capable of doing, and how well it is capable of doing it. Nothing affects the overall quality of a computer more than its components.

Normally the more expensive a component is, the better it performs. This is a general guideline however and not a steadfast rule. Sometimes you can spend a lot more money on a component with only slightly better performance than one costing half as much. Other times a very expensive component might be based on a completely new technology that is not ready for mass production. In these cases, one is often better off buying a more mainstream part.

Being an early adopter is not always the most practical move when speccing components for a new system. Often you can find very powerful hardware at the medium price ranges. There is normally a relatively large sweet-spot in the market.

How can you know if a component is good or bad? You want to be an IT professional, right? IT professionals need good computers without performance

bottlenecks. So do some research. Read articles about components on a website. Where do you find them? Just Google it!

One of my favorite places for objective customer reviews of components is Tom's Hardware Guide. Another place for objective information is on well-known websites such as Amazon.com or NewEgg.com. On sites like these, customers will often write both positive and negative reviews shortly after receiving their products. This can help you decide what to buy!

Imagine you want to build your own computer. It's not that difficult or expensive really. I personally think it's kind of fun, How would you start? If you are experienced, you would start by choosing the components first! Components must be compatible with each other in order to function correctly. For example not all processors are compatible with all motherboards. Research is necessary to solve your dependencies.

If you can't afford the exact parts you want to get all at the same time, you can use old parts or buy cheaper parts at first if you have to. Why? Because certain components can be upgraded to attain increased performance. For example, a video card (or graphics card) can be upgraded to improve the graphics for a CAD/CAM application or 3D gaming experience.

At the heart of the computer lies several key components sitting on the motherboard including the microprocessor, the **chipset**, RAM and a ROM firmware instruction set called the BIOS. These core components are connected by several «buses» made to carry information around the system and eventually out to display devices and other peripherals.

The **CPU** is another name for the 'brain' of the computer and normally includes the microprocessor and RAM. This is what does all the calculations. One or more **coprocessor**s may or may not be needed depending on what the computer is used for. In the 20th century, coprocessors were often used for mathematics such as floating point operations. Today however coprocessors are mostly used for 3D graphics (**GPU**s), sound generation, and physics applications.

As you probably learned in an earlier chapter, RAM is the memory which allows your computer to hold the operating system and all running programs while your

computer is in use. On the contrary, ROM is a kind of permanent memory which is still in tact even when the computer is off. The **BIOS** is a good example of an application using ROM. The BIOS controls very low-level access to the hardware.

Busses and ports are general terms for connectivity components with connect the different parts of the PC together. These include the serial port, parallel port, PCI and PCIe busses, and the Universal Serial Bus (USB) controller. These devices allow communication between different parts of the system. Also network interface cards are now standard on most motherboards, although **USB** and **PCI** versions of the devices are also available.

Your optical drives and hard disk drives are also components in your computer. To allow data interchange between your **CPU** and drives, **SATA**, ATA, and SCSI controllers are still widely used.

The core multimedia components include the sound card and graphics card. They make computing more fun and useful for creative professionals such as designers, gamers, and musicians. Multimedia is definitely a place where high-quality components really matter.

Feeding all these components with a steady supply of energy is another component called the power supply. This is an often overlooked piece of hardware but obviously very important! A low quality power supply can cause havoc in a computer system. On the other hand a bigger than necessary power supply can increase system heat, waste power, and make a lot of noise. Choose wisely!

At the most exterior of the computer we see the computer case. This is meant to look good, protect the components, and provide an easy interface to plug in peripherals. If you are buying or building your own computer, make sure it has a good case.

Apple is well-known for high quality PC and laptop cases, although most major companies have fair to medium quality PC cases. Beware of computers with cheap looking plastic cases. If a computer manufacturer uses a cheap case, it's very likely they are also using other cheap components inside as well. Cheap components equal a slow computer which will break after moderate use. If you intend to use a computer

for several hours every day, it makes sense to buy the very best one which fits your needs and budget.

I hope this reading inspires you to learn more about the components in your computer. Just remember that putting computers back together is harder than taking them apart! (Adapted from https://www.english4it.com/module/core/unit/11/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1.What is the primary function of the CPU in a computer system?

a) Storing data permanently

b) Carrying information between components

c) Performing calculations and processing tasks

d) Controlling access to hardware

2.Which of the following components is responsible for supplying power to all other parts of the computer?

a) Motherboard

b) Power supply

c) BIOS

d) Graphics card

3. What is a key consideration when selecting components for a new computer system?

a) Components must be the most expensive available.

b) Components must be compatible with each other.

c) Components should be purchased from a single retailer.

d) Components should focus solely on multimedia performance.

Exercise 5. Translate this text into English.

Комп’ютерне обладнання складається з фізичних компонентів, які разом утворюють повноцінну комп’ютерну систему. Розуміння цих компонентів допомагає краще зрозуміти, як працює комп’ютер і які частини можна покращити для досягнення кращої продуктивності.

Основні компоненти комп'ютера

1. **Материнська плата** Материнська плата є основою комп’ютера, на якій встановлюються інші компоненти, такі як процесор, оперативна пам’ять, відеокарта та накопичувачі. Вона забезпечує взаємодію між усіма частинами системи.

2. **Центральний процесор**

CPU або центральний процесор — це «мозок» комп’ютера. Він виконує всі обчислення і обробляє дані. Швидкість і продуктивність комп’ютера значною мірою залежать від якості процесора.

3. **Оперативна пам'ять**

Оперативна пам’ять зберігає дані, які використовуються комп’ютером під час роботи. Це тимчасова пам’ять, яка забезпечує швидкий доступ до даних і програм.

4. **Постійна пам'ять**

ROM − це пам’ять, яка зберігає дані навіть після вимкнення комп’ютера. Наприклад, BIOS − це програма, записана у ROM, яка контролює базову роботу обладнання.

5. **Накопичувачі (HDD, SSD)**

Жорсткий диск (HDD): Використовується для довготривалого зберігання даних, таких як файли, програми та операційна система.

Твердотільний накопичувач (SSD): Швидший і надійніший за HDD, використовується для пришвидшення роботи системи.

6. **Відеокарта**

Відеокарта обробляє графіку та відео, що особливо важливо для дизайнерів, геймерів і спеціалістів з 3D-моделювання.

7. **Блок живлення**

Блок живлення забезпечує електроенергію для всіх компонентів комп’ютера. Важливо обрати якісний блок живлення, щоб уникнути проблем з роботою системи.

8. **Корпус**

Корпус захищає компоненти комп’ютера та забезпечує доступ до портів для підключення периферійних пристроїв.

Додаткові компоненти

1. **Звукова карта**

Відповідає за якісний звук, що особливо важливо для музикантів і любителів медіа.

2. **Мережевий адаптер**

Забезпечує підключення до локальних мереж і Інтернету.

3. **Дисководи**

Використовуються для зчитування CD, DVD або Blu-ray дисків.

Периферійні пристрої

Периферійні пристрої підключаються до комп’ютера для розширення його функціональності:

- **Монітор**: Відображає зображення та інформацію.
- **Клавіатура та миша**: Забезпечують введення даних.
- **Принтер, сканер**: Використовуються для роботи з документами.

Отже, кожен компонент відіграє важливу роль у роботі комп’ютера. Якість цих компонентів визначає продуктивність, стабільність і функціональність системи. Для ефективної роботи важливо обирати якісне обладнання, яке відповідає вашим потребам та бюджету.

Writing Activity

Exercise 6. Pick a question and write about it.

- What components are in your computer, laptop or smartphone? What are their main function?

- The more expensive a component is, the better it performs. Do you agree with this statement?

Speaking Activity

Exercise 7. Discuss the questions.

- The microprocessor is often thought to be the central brain of a computer because it performs most of the calculations.

- Why earlier computers were used only by large organizations and now they are accessible to everyone?

- How can you know if a component is good or bad? Which websites do you use to check it?

Unit 8. Peripherals

Exercise 1. Think and answer:

Do you know what computer peripherals mean? Do you have any computer peripherals at your disposal?

Exercise 2. Study the list of terms and examples of their usage below:

associated (асоційований) - connected with something else.

The failure of the printer was found to be closely associated with the upgrade of the operating system.

daisy chain (послідовне підключення) - connecting multiple devices in a row or sequence.

The man daisy chained 3 portable USB drives to increase the storage on his laptop to 3 TB.

DIP switch (перемикач типу DIP) - a set of electric switches using a dual in-line package design**.**

A DIP switch is an attractive alternative to a jumper setup because there are no parts to lose.

display (дисплей) - an output screen which contains visual information; some variants include: LED lights, CRT or LCD monitors, and virtual reality goggles.

The almost man missed his connecting flight because the flight display at the airport showed nothing but a large «fatal error» message.

dongle (адаптер або апаратний ключ) - a hardware adapter that can convert one port type to another; a hardware device acting as a security measure for a software program

The software no longer functioned when the man lost his dongle.

Enhance (покращувати) - increase functionality or quality.

Every year tech manufacturers enhance their latest products to make them more attractive to shoppers

extend (розширювати, подовжувати) - make something go further or last longer.

The technician used an external SSD hard drive to extend the lifespan of the base model Macbook Air.

Generic (загальний, універсальний) - common, general, or non-specific; unbranded.

A generic solution is one meant to apply successfully to many situations.

Jumper (перемичка) - a short length of wire wrapped in plastic used to route a circuit by linking two cross connect termination points.

Instead of software device drivers, older peripherals used to be programmed by setting jumpers.

Modem (модем) - a device which converts digital signals into analog signals, and back again when needed.

You normally need some type of modem to connect to the Internet from home, such as a cable modem or a DSL modem.

monitor (монітор) - a visual interface display between the computer and the operator; normally a large rectangular LCD or LED device.

The old CRT monitor needed to be replaced because it was flickering badly.

Mousepad (килимок для миші) - a small mat optimized for using a computer mouse.

The man's desk was made of rough wood so using a mousepad was not optional.

Peripheral (периферійний пристрій) - an external computer add-on, such as a printer or a scanner; also known as an 'accessory'.

The woman hated the look of all the tangled wires behind her desk, which were caused by so many peripherals.

plug and play (підключи і працюй) - a term used to describe the technology where a peripheral device is automatically recognized and configured when it is detected.

Thanks to plug and play technology, the new printer worked immediately after it was connected.

printer (принтер) - a peripheral device that produces a hard copy, normally paper, from data stored in a computer.

Decent quality printers are very affordable, but the ink can be quite expensive.

scanner (сканер) - a device for capturing a binary digital image from a hard copy.

A scanner can take a photograph or magazine article and digitize it.

speaker (динамік) - an internal or external device which converts electrical impulses into audible sound.

The student turned up the speakers to play a new MP3 for his friends.

Webcam (вебкамера) **-** any specialized video camera designed to transmit video over the internet. *Many websites allow video chat using webcams, but this sometimes brings out the worst behavior in people.*

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Computer Peripherals

Peripherals are a **generic** name for any device external to a computer, but still normally **associated** with its **extend**ed functionality. The purpose of peripherals is to extend and **enhance** what a computer is capable of doing without modifying the core components of the system. A printer is a good example of a peripheral. It is connected to a computer, extends its functionality, but is not actually part of the core machine.

Do not confuse computer peripherals with computer accessories. An accessory can be any device associated with a computer, such as a printer or a **mousepad**. A printer is a peripheral, but a mousepad is definitely not one. A mousepad does not extend the functionality of a computer, it only enhances the user experience.

Peripherals are often sold apart from computers and are normally not essential to its functionality. You might think the **display** and a few vital input devices such as the mouse and keyboard would be necessary, but certain computers such as servers or embedded systems do not require mice, keyboards, or even displays to be functional.

Peripherals are meant to be easily interchangeable, although you may need to install new drivers to get all the functionality you expect out of a new peripheral device. The technology which allows peripherals to work automatically when they are plugged in is called plug and play. A **plug and play** device is meant to function properly without configuration as soon as it is connected. This isn't always the case however. For this reason some people sarcastically refer to the technology as 'plug and pray'.

Still, plug and play was a big deal when it was introduced in the 1990's. Before then, installing a new peripheral could take hours, and could even require changing some **jumper** settings, DIP switches, or even hacking away at drivers or config files. It

was not a fun time except for real hardware geeks. With plug and play technology, all the nasty jumpers and DIP switches moved inside the peripheral and were virtualized into firmware. This was a clear victory for the common, nontechnical person!

Peripherals normally have no function when not connected to a computer. They connect over a wide array of interfaces. Some common ones from the past include: PS2 ports, serial ports, parallel ports, and VGA ports. These are all being replaced by some new standards including USB, Bluetooth, wifi, DVI, and HDMI ports.

The most common peripheral linking device is probably USB technology. Why? USB is good because you can **daisy chain** a lot of peripherals together quickly, it is quite fast and growing ever faster in recent editions, and it even provides enough power to supply some smaller peripheral devices like webcams and flash drives.

Some peripherals are even used for security. A good example of this is the dongle. The dongle is often used to protect very expensive applications from software piracy.

Here is a list of common peripherals you should be familiar with as an IT professional. Keep in mind the list is always changing due to changing technologies:

- **monitor**s or displays
- **scanners**
- **printers**
- external **modem**s
- dongles
- **speakers**
- **webcams**
- external microphones
- external storage devices such as USB-based flash drives and portable hard disk drives
- input devices such as keyboards, mice, etc are normally considered peripherals as well

Now you know a little more about peripherals and what makes them different from components and accessories. I hope you enjoyed the reading! (Adapted from https://www.english4it.com/module/core/unit/12/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1. What is the primary purpose of computer peripherals?

a) To replace core components of a computer

b) To enhance the user experience only

c) To extend and enhance the functionality of a computer

d) To store data permanently

2. Which of the following is considered a peripheral?

a) Motherboard

b) Mousepad

c) Printer

d) BIOS

3. What technology allows peripherals to work automatically when connected to a computer?

a) Plug and play

b) USB

c) Firmware

d) VGA

Exercise 5. Translate this text into English.

Периферійні пристрої − це пристрої, які підключаються до комп'ютера для розширення його функціональності, але не є частиною його основних компонентів. Вони відіграють важливу роль у підвищенні можливостей комп'ютера та забезпеченні зручності його використання.

Типи периферійних пристроїв

1. **Вихідні пристрої (Output Devices):** Ці пристрої виводять інформацію, оброблену комп'ютером, у зрозумілій формі:
   - **Монітори**: Відображають візуальну інформацію.
   - **Принтери**: Друкують документи та зображення.
   - **Колонки та навушники**: Відтворюють звук.

2. **Вхідні пристрої (Input Devices):** Ці пристрої дозволяють користувачеві вводити дані в комп’ютер:
   - **Клавіатури**: Для введення тексту.
   - **Миші**: Для навігації та вибору.
   - **Сканери**: Для перетворення фізичних документів у цифрові.
   - **Веб-камери та мікрофони**: Для відео- та аудіовходу.
3. **Пристрої зберігання даних (Storage Devices):** Ці пристрої дозволяють зберігати інформацію поза межами комп’ютера:
   - **Зовнішні жорсткі диски**
   - **Флеш-накопичувачі USB**
4. **Мережеві периферії (Network Peripherals):** Пристрої, які забезпечують підключення до мережі:
   - **Зовнішні модеми**
   - **Мережеві адаптери**
5. **Спеціалізовані периферії (Specialized Peripherals):** Пристрої для особливих завдань:
   - **Донгли (Dongles):** Використовуються для захисту програмного забезпечення.
   - **Ігрові контролери**

Технології підключення периферійних пристроїв

Сучасні периферійні пристрої підключаються за допомогою різноманітних інтерфейсів:

- **USB (Universal Serial Bus):** Найпоширеніша технологія завдяки швидкості та можливості підключати декілька пристроїв одночасно.
- **Bluetooth:** Бездротове підключення для мишей, клавіатур та колонок.
- **HDMI та DVI:** Для підключення моніторів і телевізорів.
- **Wi-Fi:** Використовується для бездротових мережевих пристроїв.

Переваги периферійних пристроїв

- **Розширення функцій:** Наприклад, принтери додають можливість друку, а веб-камери — відеозв’язок.
- **Гнучкість і зручність:** Периферії легко замінити або оновити.
- **Зручність у використанні:** Завдяки технології *plug and play* більшість пристроїв працюють відразу після підключення.

Отже, периферійні пристрої є невід’ємною частиною сучасного комп’ютера. Вони дозволяють адаптувати систему до різних потреб, покращуючи її продуктивність та зручність. Обираючи периферії, важливо звертати увагу на їхню якість, сумісність і необхідність для конкретних завдань.

Writing Activity

Exercise 6. Pick a question and write about it.

1. Read the statement and give your opinion on it. Do you agree/disagree? Write other examples to support your thoughts.

“*We use peripherals to make our daily lives easier. Inputting information into a computer using a keyboard makes it easier. Sending data to a printer is easier than copying what is on a screen.”*

2. Write the functions of each device as in the example. Add 3 more devices to the list.

- Screen magnifiers – *allow the screen to zoom in for people with visual impairments.*
- Screen readers – allow...
- Foot controlled mouse – allows ...
- Touch screen – ….
- Different types of keyboards – …

Speaking Activity

Exercise 7. Discuss the questions.

- What would the problem be if there were no peripherals?
- How would it change the way computers or digital devices worked if they did not have peripherals?
- What peripherals do you use each day?

- How do peripherals help to send and receive information from a computer?

Unit 9. Networking

Exercise 1. Think and answer:

What specific benefits does networking computers give us? What are the risks involved?

Exercise 2. Study the list of terms and examples of their usage below:

authenticate (автентифікувати) - to verify that someone is really the person they claim to be.

The computer was very slow and took nearly 10 seconds to authenticate a user.

Bluetooth (Блютуз) - an open wireless protocol for exchanging data; primarily used for connecting mobile devices to computers

The man used Bluetooth technology to create a personal area network (PAN) between his mobile phone and his computer.

client-server (клієнт-сервер) - a networking model in which the functions are divided between «child» computers and a single «parent» computer which stores and controls access to data.

A standard protocol such at TCP/IP or IPX is used to define client-server interactions.

distributed computing (розподілені обчислення) - a type of computing in which a computational task is divided into subtasks that execute on a collection of networked computers.

The professor said the Internet could theoretically be used as a giant distributed computing platform.

Domain (домен) - a named group of networked computers that are administered as a unit with common rules and procedures.

System administrators in large organizations use a domain to save time on common administration tasks.

Encryption (шифрування) - the process of making information 'more secure' by rendering it unreadable to anyone but the intended recipient.

The company used SSL encryption on its ecommerce web site to protect its customers while they made online purchases.

ethernet - a baseband local-area network originally developed by Xerox Corp. Normally running between 10 Mbps and 1,000 Mbps.

The man could not connect to the LAN because he forgot to bring his RJ-45 ethernet cable.

firewall - a device or software program designed to prevent unauthorized access to a network.

When the firewall was deactivated, the hacker found it easy to penetrate the system and upload a rootkit virus.

Groupware (групове програмне забезпечення) **-** software applications that facilitate shared work on documents and information.

Since the members of the project were thousands of miles apart, a groupware application was an integral part of the solution.

IP address (IP-адреса) **-** a unique string of numbers that identifies a computer or server on the Internet.

The server's IP address was 215.44.32.134.

LAN (local area network) (локальна мережа) **-** a small computer network normally contained within one room or building.

The network administrator talked with great enthusiasm about upgrading the LAN to Gigabit Ethernet.

node (вузол) **-** any computer or other device connected to a network.

The Networking 101 book said that all connected PCs, servers, and printers are nodes on a network.

packet (пакет) **-** a collection of bits normally sent through a network that contains data surrounded by error correction information.

The network administrator was upset because the network connection kept dropping packets.

P2P (peer-to-peer) (однорангова мережа) **-** a networking model where computers seamlessly share data; used primarily to reduce sever bottlenecks and reduce costs.

BitTorrent is a popular and controversial peer-to-peer networking program used by millions of people to distribute data.

protocol (протокол) - a set of standards or rules which govern communication.

Before the Internet age, corporate LANs often ran on IPX or NetBEUI protocol.

VPN (virtual private network) (віртуальна приватна мережа) - a high security connection to a private network.

The man connected to his work intranet from home using a virtual private network.

Wi-Fi - a trademarked, non-technical term used to describe a certain quality of wireless networking compatibility.

The businessman was upset because he could not get his laptop to connect to the Wi-Fi hotspot at the convention.

WAN (wide area network) (глобальна мережа) - a network which covers more than one building or area

The technical college's WAN connected all the county campuses.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Basic Networking

In the simplest explanation, networking is just computers talking to each other. They do this by sending data **packets** using various **protocols** and transmission mediums such as **ethernet** cable or **Wi-Fi** connections. Computers must also know how to find other computers on the network. To put it briefly, every computer on the network needs a unique address so messages know where to go after they are sent.

Networks exist for many reasons including:

- **distributed computing** in a **client-server** or **peer-to-peer** networking architecture
- centralized data security and authentication
- elimination of risk of computer downtime.
- combining computers into a single **domain** to facilitate **groupware** applications and system administration tasks
- communication and fun!

The types of networks you deal with on a daily basis include **local area networks** (LANs) and **wide area networks** (WANs).

Many people today have LANs in their schools, offices, and even their homes. LANs are especially good for sharing Internet access and commonly used files and databases.

Users can also connect to wide area networks (WANS) as well, which are just large LANS spread out over several physical locations. The Internet itself is basically a large WAN, with each **node** on the network having it's own unique **IP address**.

As you may have read in books or seen in movies, security considerations play a large role when designing networks. Technology such as **firewalls** can both block and filter unwanted network traffic. Virtual private networks (VPNs) are used to connect remote users to office networks without jeopardizing security. VPNs use strong data **encryption** to hide data as it is moving between routers over the Internet.

Networking is not something you can master in a week or even a month. Hundreds of books have been written about the subject and many more hundreds will come in the future as technologies mature and evolve. If you work on networks for a living, you are called a network engineer, and you will probably take certification exams by networking companies such as Cisco.

There are other kinds of networking as well which are not always between PCs and servers. An example is **Bluetooth** technology, which is optimized for networking between common consumer electronics such as mobile phones, mp3 players, and similar devices.

Okay that's it for now! This is just a basic introduction to networking. (Adapted from https://www.english4it.com/module/core/unit/13/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1. What is the primary purpose of a network?

a) To enable computers to talk to each other and share resources

b) To replace hardware components in a computer

c) To eliminate the need for an Internet connection

d) To avoid using unique addresses for computers

2. What is the difference between a LAN and a WAN?

a) LANs are large networks spread across multiple locations, while WANs are local networks.

b) LANs require a unique IP address for each device, but WANs do not.

c) LANs are smaller networks within a single location, while WANs connect multiple locations over larger distances.

d) LANs use Wi-Fi, while WANs use only ethernet cables.

3. What is the purpose of a VPN in networking?

a) To create a unique IP address for each user

b) To connect remote users to a secure network with encryption

c) To block and filter unwanted network traffic

d) To allow computers to bypass firewalls

Exercise 5. Translate this text into English.

Комп'ютерна мережа − це система, яка дозволяє комп'ютерам обмінюватися даними. У найпростішому вигляді це можна описати як «спілкування» комп'ютерів між собою. Для цього вони використовують пакети даних, що передаються за допомогою різних протоколів і середовищ передачі, таких як кабелі Ethernet або з'єднання Wi-Fi. Щоб дані потрапляли за призначенням, кожен комп'ютер у мережі повинен мати унікальну адресу.

Причини створення мереж

Мережі створюються для вирішення багатьох завдань, серед яких:

- **Розподілені обчислення:** Архітектури «клієнт-сервер» або «peer-to-peer».

- **Централізована безпека даних та автентифікація.**
- **Зменшення ризику простою комп’ютерів.**
- **Об’єднання комп’ютерів у єдиний домен:** Для спрощення управління та спільної роботи.
- **Комунікація та розваги.**

Типи мереж

1. **Локальні мережі (LANs):**
   - Локальні мережі є звичайним явищем у школах, офісах і навіть вдома.
   - Вони ідеально підходять для спільного використання Інтернету та файлів.
2. **Глобальні мережі (WANs):**
   - Глобальні мережі об'єднують кілька фізичних локацій.
   - Інтернет є прикладом глобальної мережі, де кожен вузол має унікальну IP-адресу.

Безпека в мережах

Безпека є важливою складовою проєктування мереж:

- **Міжмережеві екрани (firewalls):** Блокують і фільтрують небажаний трафік.
- **Віртуальні приватні мережі (VPNs):** Дозволяють віддаленим користувачам безпечно підключатися до офісних мереж. VPN використовують шифрування для захисту даних під час їх передачі.

Інші види мереж

Мережі не завжди обмежуються зв’язками між ПК і серверами:

- **Bluetooth:** Використовується для підключення споживчої електроніки, наприклад, мобільних телефонів, MP3-плеєрів та інших пристроїв.

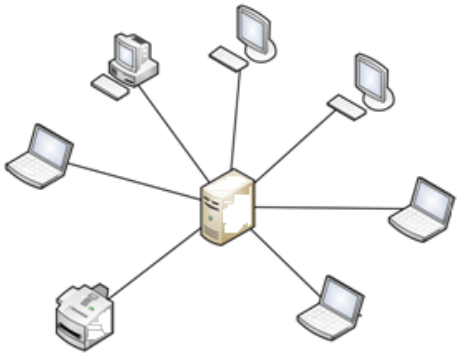
Перспективи вивчення мереж

Вивчення мереж є складною і тривалою справою. Для цього написано сотні книг, і з появою нових технологій з’являтиметься ще більше. Людей, які

працюють з мережами, називають мережевими інженерами. Вони часто проходять сертифікацію в компаніях, таких як Cisco.

Writing Activity

Exercise 6. Pick a question and write about it.

	1. From the picture on your right fill in the blanks. 1. Number of server _____ 2. Number of workstation _____ 3. Name the resource being shared _____ _____

2. Write in your own words why in the section head office of junior section there is only one printer? What is the advantage of using only one printer?

Speaking Activity

Exercise 7. Discuss the questions.

- If a coworker asked you to explain yourself after you used the phrase network of computers, what would you say to her? Elaborate so that a novice would understand completely.
- If you knew that your your employer's plan stipulated that sharing sensitive information was to be strictly controlled, and you agreed with those controls, how would that knowledge affect the degree of data sensitivity that you would be willing to share over that network's resources?
- Although you might choose to share some (or all) of your personal information with selected classmates, would you feel comfortable if you thought your

instructor planned on sharing your whole file freely with everyone in your class without your permission?

- Even if it were not yet true, would the thought of your instructor sharing your information freely affect the amount of information you shared when someone else in authority on the network requested sensitive data?

Unit 10. Memory and Storage

Exercise 1. Think and answer:

What storage capacity do your devices have?

Exercise 2. Study the list of terms and examples of their usage below:

EPROM (erasable programmable read only memory) - a non-volatile (permanent) memory type that is erasable via ultra-violet light and re-programmable.

EPROMs are popular for use in critical firmware applications, such as BIOS's, because they are both non-volatile and they can be updated when necessary.

floppy disk drive (дисковод для дискет) - a legacy storage device which can read and write data slowly from a removable magnetic medium (normally 3.5» in diameter and holding 1.44 MB of data).

Floppy disk drives were once included with almost all computers, but are now considered obsolete.

hard disk drive (жорсткий диск) - a storage device using rotating magnetic platters to store and retrieve data.

A hard disk drive can fail at any time, normally causing data corruption and loss.

magnetic storage (магнітне зберігання) - a storage technique using patterns of charged particles on a metallic surface to store data.

Magnetic drives have been around since the 1950s and are typically used to store large amounts of data.

non-volatile (неволатильний) - long-term, persistent, does not require power to retain its state.

The laptop contained 500MB of non-volatile storage.

optical storage (оптичне зберігання) - a data storage technique using a pattern of markings on a disc that can be read by a laser; examples include CD-ROM, and DVD-ROM technology

Optical storage is a good choice for distributing software packages because it has a good mix of storage size, portability, and a low cost to manufacture.

quantum memory (квантова пам’ять) - an interface between light and matter that allows for the storage and retrieval of entangled photonic qubits.

A practical quantum memory solution must be able to reliably store and recall quantum states on demand at room temperature.

RAM (random access memory) (оперативна пам'ять) - a type of computer memory known for being volatile (temporary) and fast.

My computer was performing slowly when changing programs, so my technician installed more RAM and this solved the problem.

ROM (read-only memory) (постійна пам'ять) - a type of memory which is known for being non-volatile (permanent) and fast.

The boy gave a puzzled look when the man showed him an old Atari 800 computer ROM cartridge.

semiconductor storage (напівпровідникове зберігання) - a type of storage using integrated circuits to store data; examples include RAM, ROM, and flash memory.

All computers created today use at least some form of semiconductor storage.

SSD (solid state drive) (твердотільний накопичувач) - a storage device characterized by high speed, no-moving parts, and low energy consumption.

Many new laptop models use solid state drives, although they are more expensive and have less capacity than traditional hard disk drives.

SAN (storage area network) (мережа зберігання даних) - a cluster of storage devices working together to provide shared network storage.

With cloud hosting, your virtual server's performance is often at the mercy of latency in the SAN.

video memory (відеопам'ять) - memory intended specifically for the graphics processor.

Modern games such as GTA V load textures which can fill gigabytes of dedicated video memory.

volatile (волатильний) - temporary, requires power to retain its state.

Due to its volatility, the RAM lost all the data when the user tripped over the power cord.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Memory and Storage

In this unit you will learn the differences between computer memory and computer storage. Memory and storage are important concepts to master in Information Technology. The two terms are often used interchangeably, so it is important to understand the key differences.

Computer memory needs to be quick. It is constantly feeding the CPU with data to process. Since nobody likes to wait for a computer, high-quality computers will have fast processors with lots of quick memory. This type of memory is commonly known as **RAM**. This type of memory is **volatile** which means that the actual data disappears when the computer loses power.

Because memory needs to be much faster than storage, it is rather more expensive than storage per GB. A typical desktop computer today (in 2016) has between 2GB and 32GB of memory running at speeds of anywhere from 1.2 GHZ to 3.2 GHZ. Speeds tend to go up about 10% every two years.

If you are a gamer, video editor, or physics geek, you may be aware of **video memory**. Video memory is special RAM which is even faster and more expensive than normal system RAM. This RAM is reserved only for the graphics and is thus kept separate from the main system RAM, which sits on the motherboard close to the CPU. A typical dedicated video card will have anywhere between 2GB and 12 GB of dedicated RAM.

But of course computers do not process all the data they have at once. They also need to save some data for long term use. This is where storage comes in. Think of all the video files, mp3s, photos, and documents on your PC. These files are not always being processed by the CPU. They are mostly just hanging around waiting to be used at some point. Storage does not need to be as quick as memory, but there does need to be a lot more of it. And storage of course needs to be **non-volatile**, meaning it will not get erased when you power off or restart your computer. These are the two key differences between memory and storage.... speed and volatility.

Storage today comes in many different types including **semiconductor storage**, **magnetic storage**, and **optical storage**. A typical computer today comes with

anywhere between 128GB to 2TB of storage. Low end computers normally come with a magnetic **hard disk drive** which reads data at around 75 to 200 MB/sec. These devices use rotating, magnetically-charged platters to store data. Hard disk drives are still popular because they can store a lot of data with relatively quick access times very cheaply. Other examples of magnetic storage devices include the **tape drive** and **diskette**, both are which are obsolete. These dinosaurs of storage were painfully slow and prone to data loss with no warning. Ouch.

Another type of storage is network storage, typically referred to as a **SAN**. This storage is usually found in a datacenter. This type of storage goes by other names such as «cloud storage» or «network drive». It is of course highly limited by network speeds. If you are offline and need a file then you are out of luck. That is why it's always a good idea to get as much storage on your PC as you can afford.

In the future all magnetic storage types will become obsolete. They will soon all be replaced by **SSDs** using semiconductor technology. SSDs have 3 key advantages over magnetic storage devices: speed, lack of moving parts, and low power consumption. This makes them ideal for laptops where battery life and durability are huge issues. SSDs are fast as well, and can read data at around 200MB to 800 MB per second. Unfortunately SSDs are currently more expensive than magetic storage per GB, but this should change soon.

Optical storage is another technology which is quickly becoming legacy. Very popular in the 90's and early 2000's, optical storage works by a laser either burning or reading data off a plastic disc coated with various types of light sensitive materials. Due to reliability and speed limitations, optical storage is not used as a primary means of data storage. It is (or increasingly was) used mostly to affordably deliver large datasets like movies, games, and operating systems. In case you still don't understand what optical storage is, typical examples are DVD or Blu-Ray drives.

Well, that about covers it for the current state of memory and storage. But there are some gray areas and exceptions as always, such as **ROMs** and **EPROM**s, which are somewhere in between memory and storage. Embedded systems, BIOS' and older video game machines used these for various reasons, mainly copy protection and cost.

What about the future? Expect a gradual convergence where there is no longer a need for both storage and memory in PCs. Some new technology such as **quantum memory** may arrive which has the advantages of both memory and storage. Until that time arrives, always buy a PC with the most memory and storage you can afford. Computer companies typically charge you a lot more for PCs with a decent amount of RAM and fast storage. Why? Because they know without it, your computer will be slow and completely full very quickly, forcing you to upgrade or buy into their cloud storage solutions like Apple's iCloud, and Microsoft's OneDrive. (Adapted from https://www.english4it.com/module/core/unit/16/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1. What is a key difference between computer memory (RAM) and storage?

A. Memory is non-volatile, while storage is volatile.

B. Memory is faster and volatile, while storage is slower and non-volatile.

C. Storage is faster than memory but is more expensive per GB.

D. Both memory and storage are equally fast and volatile.

2. Why are SSDs expected to replace magnetic storage devices in the future?

A. SSDs are cheaper than magnetic drives.

B. SSDs have moving parts, making them more reliable.

C. SSDs are faster, consume less power, and have no moving parts.

D. SSDs have lower storage capacity than magnetic drives.

3. Which of the following is an example of optical storage?

A. Hard Disk Drive (HDD)

B. Solid-State Drive (SSD)

C. DVD or Blu-Ray Drive

D. Tape Drive

Exercise 5. Translate this text into English.

Пам'ять комп'ютера − це компонент, який використовується для зберігання даних і інструкцій, необхідних для виконання програм. Пам'ять дозволяє комп'ютеру швидко обробляти інформацію, і існує кілька типів пам'яті, які виконують різні функції.

1. **Оперативна пам'ять**

Оперативна пам'ять або **RAM** (Random Access Memory) є основним типом пам'яті в комп'ютері, який використовується для тимчасового зберігання даних, які обробляються процесором. Вона є **вольатильною**, тобто вся інформація в ній зникає після вимкнення комп'ютера. Оперативна пам'ять дозволяє швидко отримувати доступ до даних, що активно використовуються. Чим більше RAM у комп'ютера, тим швидше він може обробляти більшу кількість даних одночасно.

2. **Кеш-пам'ять**

Кеш-пам'ять є спеціалізованим видом пам'яті, який використовується для зберігання часто використовуваних даних і інструкцій. Вона знаходиться між процесором і основною оперативною пам'яттю і забезпечує ще швидший доступ до даних. Кеш-пам'ять має кілька рівнів (L1, L2, L3), де L1 − це найшвидший і найближчий до процесора рівень, а L3 − більший за обсягом, але повільніший.

3. **Постійна пам'ять**

ROM (Read-Only Memory) − це тип пам'яті, який зберігає дані постійно, навіть при вимкненому комп'ютері. Вона використовується для зберігання базових інструкцій, необхідних для запуску комп'ютера, таких як **BIOS** або **UEFI**. Данні в ROM зазвичай не змінюються, тому вона використовується для зберігання важливої інформації, що не потребує частих змін.

4. **Флеш-пам'ять**

Флеш-пам'ять є неволатильною пам'яттю, яка використовується в таких пристроях, як USB-флешки, SSD-диски та карти пам'яті. Вона зберігає дані навіть після вимкнення комп'ютера і має набагато більшу швидкість запису та читання, ніж традиційні жорсткі диски.

5. **Віртуальна пам'ять**

Віртуальна пам'ять − це техніка, яка дозволяє комп'ютеру використовувати частину жорсткого диска як додаткову пам'ять, коли основна оперативна пам'ять заповнена. Це дозволяє запускати більшу кількість програм одночасно, хоча доступ до віртуальної пам'яті значно повільніший, ніж до оперативної.

Пам'ять є одним з найважливіших компонентів комп'ютера, і від її обсягу та швидкості залежить загальна продуктивність системи. Чим більше оперативної пам'яті та швидший доступ до даних, тим ефективніше працює комп'ютер.

Writing Activity

Exercise 6. Pick a question and write about it.

A. Describe the difference between RAM and ROM. Explain the need for ROM in a computer system.

B. Describe the purpose of RAM in a computer system.

C. explain how the amount of RAM in a personal computer affects the performance of the computer.

D. Explain the need for virtual memory

Speaking Activity

Exercise 7. Discuss the questions.

- What do you think would be the effect on the performance of the computer of using a lot of virtual memory rather than upgrading your RAM? Explain why.
- What is Flash memory and what is it used for?
- How changes in memory technologies are leading to innovative computer designs?

Unit 11. Input Devices

Exercise 1. Think and answer:

Which input devices do you know? Why are they called so?

Exercise 2. Study the list of terms and examples of their usage below:

Alphanumeric (алфавітно-цифровий) - consisting of both numbers and letters.

The computer alerted the user that his password must be more than six characters in length and alphanumeric.

barcode reader (сканер штрих-кодів) - a hardware peripheral designed to 'scan' products into an inventory tracking system.

Harry ran into trouble while taking inventory because he forgot to charge his barcode reader.

biometric authentication (біометрична аутентифікація) - a method of identity verification using biological means such as fingerprinting, retinal scanning, iris scanriing, or voice recognition.

In the future biometric authentication will be as common as today's username and password authentication.

bit (binary digit)(біт) - the smallest unit of storage; normally referred to as a '1' or '0'.

The DBA «flipped a bit» in the database, changing a value from a 0 to 1.

click (клік) - to press and release the primary (left-side) mouse button.

The screen prompt read «Click the button to continue.»

cursor (курсор) - a (sometimes) blinking line or other character that indicates the location of the next input on the display screen.

In a modern word processor, you can make the cursor jump position by using the mouse.

double-click (подвійний клік) - to depress and release the left mouse button twice in quick succession.

The Macintosh user did not know that in order to open a file in Windows you need to double-click it.

fingerprint reader (сканер відбитків пальців) - a device which forms a digitised image of a human finger print for the purpose of biometric authentication.

The security firm was so tired of people choosing weak passwords that they installed fingerprint readers to increase the security of their systems.

graphics tablet (графічний планшет) - an input device or mouse alternative using a sensitive rectangular surface and a stylus. *The graphic designer sometimes used a graphics tablet in addition to a mouse.*

input device (пристрій введення) - any peripheral used primarily to enter data into a computer.

The mouse and the keyboard are two examples of common input devices.

joystick (джойстик) - an input device with a vertical rod mounted on a base used to control pointing devices or on-screen objects; normally with one or more buttons

The man used a joystick to control the airplane in the flight simulation.

jumper (джампер) - a short length of wire wrapped in plastic used to route a circuit by linking two cross connect termination points.

Instead of software device drivers, older peripherals used to be programmed by setting jumpers.

light pen (світловий олівець) - an input device using a light-sensitive pen shaped handle which can be used to draw on the computer screen.

Before the mouse became available, some older computers used light pens to increase interactivity.

Microphone (мікрофон) - a hardware device plugged into a computer's sound card optimized for voice commands or other audio recording

The handicapped woman used a microphone to dictate her Word documents instead of typing.

Mouse (миша) - a handheld input device used to move an onscreen pointer by means of sliding on a flat surface.

Before the invention of the mouse, the 'tab key' or 'arrow keys' were often used to navigate the screen.

mouse pointer (вказівник миші) - a special icon (normally an arrow) which indicates the current location of the mouse on the desktop screen.

The mouse pointer changes its shape at times, depending on the application being run and the task being performed.

OCR (optical character recognition) (оптичне розпізнавання символів) - the ability of a computer to determine standard letters and numbers based on fuzzy logic.

Instead of retyping the stack of documents, the secretary decided to use OCR technology to scan them.

right-click (права кнопка миші) - to press the right-side mouse button.

Windows 95 was the first major operating system to popularize the right-click.

speech recognition (розпізнавання мови) - the ability of a computer to convert spoken words from a user into usable data.

The new phone system used speech recognition to help reduce the number of customer service representatives.

Stylus (стилус) - a part of an input device resembling a pen used to draw on the surface of a graphics tablet.

The man lost his stylus and thus could no longer enter text on his PDA until he replaced it.

switch (перемикач) - a block of code or function causing a program to change its default behavior.

The programmer preferred using a switch statement to many if/else statements.

touchpad (сенсорна панель) - an input device consisting of a sensitive rectangular area in which one uses a finger to move a cursor on a display.

Laptops commonly include a built-in touchpad in addition to supporting a mouse.

Touchscreen (сенсорний екран) - a display that also acts as an input device by allowing a user to navigate a program by pressing specific locations of the screen.

Most information kiosks in airports make use of touchscreens instead of keyboards or mice.

Trackball (тракбол) - a pointing device consisting of a ball on top of a base that is rotated to move a cursor on the computer screen.

The injured person used a trackball instead of a mouse to move the cursor.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Some Common Computer Input Devices

We use **input devices** every time we use a computer. Simply speaking, it is these devices which allow us to enter information. Without them, the computer would not know what we want it to do.

Some of the things we do with input devices are: move a **cursor** around the screen, enter **alphanumeric** text, draw pictures, and even enter binary data in the form of graphics or audio wave forms.

Input devices have a history as long as computers themselves. Perhaps the first input device was the simple electronic **switch** (similar to a light switch) which turned **bits** on or off. There were hundreds or even thousands of these switches on larger computers. It used to take a team of programmers hours or even days to set up a computer to perform a single calculation.

Switches and **jumpers** are still used today on computers. For instance, the power button on the computer is a switch which is also an input device telling the computer to power on or power off. Tiny switches called jumpers are also widely used on motherboards to change important settings such as processor clock speed or memory speed.

Most likely in front of you right now are two of the most popular input devices: the keyboard and the mouse. And instead of a mouse on a laptop computer you normally have a **touchpad**.

As computers evolved throughout the late 20th century, computers became more and more interactive. Input devices came and went. Some lasted and some did not. The **light pen** and the **joystick** are almost unknown today, although they were popular before the **mouse** and the gamepad became well-known. Touch screens are already replacing keypads on mobile phones and may come to replace or augment keyboards and mice on PCs and laptops in the near future.

Different people prefer different input devices for doing same task. For instance, many graphic artists prefer to use a stylus and graphics tablet rather than a mouse. It

might offer them a greater deal of artistic freedom, or precision while performing their work.

Sufferers of carpal tunnel syndrome often prefer a **trackball** or **stylus** to a mouse. Handicapped computer users have invented a wide array of input devices designed to replace the mouse including devices controlled by foot or even eye movement.

Not only PCs and mainframes use input devices. Almost all computers feature some kind of input device. Special scanners are used in many stores and warehouses called **barcode readers** to enter stock and sell items at the cashier. These are input devices as well. Even **microphones** can technically be called input devices as a computer can respond to them and interpret them as incoming data.

Corporations and especially government institutions are already implementing the second generation of input devices to improve security. These include retina scanners and/or **fingerprint readers** to replace or improve accuracy of username and password authentication. You will be seeing more of this kind of **biometric authentication** in the coming years as a general remedy for weak passwords or leaked passwords.

In summary, input devices are how you interact with a computer. The computer responds to your input and hopefully does what you need it to do. It seems really simple, and that's the way it was meant to be! (Adapted from https://www.english4it.com/module/core/unit/17/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1. Which of the following is NOT typically considered an input device?

a) Keyboard

b) Mouse

c) Printer

d) Touchpad

2. What input device is commonly used by graphic artists for greater precision?

a) Mouse

b) Keyboard

c) Stylus and graphics tablet

d) Joystick

3. What type of input device is used to improve security by verifying identity?

a) Barcode reader

b) Retina scanner

c) Touchpad

d) Microphone

Exercise 5. Translate this text into English.

Комп'ютерні пристрої вводу − це апаратні засоби, які дозволяють користувачеві взаємодіяти з комп'ютером, вводячи дані та команди. Ці пристрої мають велике значення для функціонування комп'ютерних систем, оскільки саме вони дозволяють користувачам вводити інформацію, яку комп'ютер потім обробляє.

Основні типи комп'ютерних пристроїв вводу включають:

1. **Клавіатура**

Клавіатура − один із найпоширеніших пристроїв вводу. Вона дозволяє вводити текстові дані, а також управляти комп'ютером через спеціальні комбінації клавіш. Клавіатури можуть бути проводовими або бездротовими.

2. **Миша**

Миша використовується для переміщення курсора на екрані та взаємодії з графічним інтерфейсом. Вона може бути стандартною з двома кнопками або мультимедійною з додатковими кнопками для виконання різних функцій.

3. **Трекпад**

Трекпад або сенсорна панель є альтернативою миші і часто використовується на ноутбуках. За допомогою трекпаду користувач може переміщати курсор і

виконувати клацання, натискаючи на саму панель або здійснюючи жести пальцями.

4. **Сенсорний екран**

Сенсорні екрани дозволяють користувачам безпосередньо взаємодіяти з екраном, торкаючись його пальцем або стилусом. Цей тип вводу стає все популярнішим на мобільних пристроях, планшетах та інтерактивних панелях.

5. **Графічний планшет та стилус**

Графічний планшет із стилусом дає змогу художникам і дизайнерам виконувати малюнки та креслення з великою точністю, що є зручнішим, ніж використання миші. Стилус дозволяє здійснювати точніші рухи, ніж палець.

6. **Мікрофон**

Мікрофон дозволяє вводити голосові команди або записувати аудіо. Вони використовуються для голосових помічників, а також для голосового введення тексту в програми.

7. **Сканери**

Сканери, зокрема, зчитувачі штрих-кодів, дозволяють вводити зображення, текст або інші фізичні дані в цифровий формат. Вони широко використовуються в офісах, бібліотеках, магазинах і на складах для швидкого введення даних.

8. **Камера**

Веб-камери та інші типи камер можуть бути використані для введення зображень або відео в комп'ютер. Це дозволяє здійснювати відеозв'язок, а також використовувати камеру для розпізнавання осіб або жестів.

9. **Біометричні пристрої**

Біометричні пристрої вводу, такі як зчитувачі відбитків пальців, сканери сітківки ока або камери для розпізнавання обличчя, використовуються для аутентифікації та підвищення безпеки системи. Вони дозволяють комп'ютеру визначити особу користувача без необхідності введення пароля.

Пристрої вводу є важливою частиною користувацького досвіду, оскільки вони дозволяють ефективно та швидко передавати дані від людини до комп'ютера, забезпечуючи комфортну роботу з комп'ютерними системами.

Writing Activity

Exercise 6. Pick a question and write about it.

1. Describe input devices by filling the table below.

Input Device	How it works	Where it is used	Why used in these applications	
			Advantages	Disadvantages

2. Describe the purpose of a mouse and the role it plays in computing.

Speaking Activity

Exercise 7. Discuss the questions.

- What input devices do you currently use? How do they make it easier to work?
- What is the future of input devices?
- What functions of input devices need to be advanced for more efficient usage of the computer?

Unit 12. IT Slang

Part 1

Exercise 1. Think and answer:

What IT Slang do you already know?

Exercise 2. Study the list of terms and examples of their usage below:

code monkey (кодуючаа мавпочка) - someone who sits and programs all day; someone who is only valued for their programming skills.

The bad software architect said, «Tell the developers to stop thinking and start typing like good little code monkeys!»

early adopter (ранній користувач) - a person who likes new things and often buys them before they are proven.

A geek is a typical early adopter, often trying the latest and greatest technology before their peers have even heard of it.

EOL (end of life) (закінчення терміеу експлуатації) - the point when something is no longer useful or relevant.

Do you think compiled languages have reached their EOL?

fanboy (фанат) - someone obsessed or devoted to a particular item or genre.

Tom is such a fanboy he still uses his old Commodore Amiga as his main computer.

flame (флейм) - to write an intentionally abusive reply to a comment or forum post.

Joe the sysadmin got sent to counseling because he started too many flame wars on the company intranet.

geek (гік) - an individual with a passion for computers, to the exclusion of other normal human interests.

The geek did not have a girlfriend, but he did have a computer named Lisa which he programmed to sing him love songs before bedtime.

horked (сильно зламаний) - completely broken in some mysterious way; corrupted.

The web sites crashed because the server was completely horked.

killer app (вбивчий додаток) - a piece of software which redefines the industry, often causing people to rush out and buy new hardware so they can use it.

Email was the first killer app of the Internet.

leading edge (передова технологія) **-** at the forefront of a new technology; also referred to as «state-of-the-art».

Kyle's company is so leading edge they buy everyone new computers every six months.

PHB (pointy haired boss) (недалекоглядний бос) **-** this is a derogatory reference to a stereotypical IT manager who choose their technology platforms based on the latest TV commercial and have no knowledge at all of what their employees actually do.

This is how the PHB is described on the Dilbert web site: «He wasn't born mean and unscrupulous, he worked hard at it.»

spaghetti code (спагетті-код) **-** a derogatory term for poorly organized programming code.

The lead programmer threw his hands up in the air and screamed when he saw the 10,000 lines of spaghetti code that management asked him to debug by tomorrow.

trendsetter (трендсетер) **-** someone who successfully defines a path for the rest of the industry to follow.

Apple is a company of trendsetters, whom everyone else is just following like lemmings.

troll (троль) **-** a person who writes intentionally controversial posts on a user forum in order to start a flame war.

The forum's popularity among serious users went down when the system operator refused to banish the trolls.

whatever (що завгодно) **-** something sarcastic you can say to non-technical people when they tell repeatedly give you misguided technical instructions that are impossible, impractical, dangerous, or overly time-consuming.

«Whatever, dude,» replied the programmer when his manager asked him to work overtime to recover his deleted files.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

A Controversial IT Purchase

People often use funny slang and abbreviations in chats and other social situations. Typically, there is a relaxed attitude towards the rules of grammar. The following is a chat transcript between an older grumpy senior developer and a younger intermediate developer. They are discussing a recent hardware acquisition.

Please note that you should not use slang with unfamiliar people. Using slang is not professional and can be taken badly by some people. You have been warned!

-- The Bunion --: Did you hear what happened? We had to buy some new hardware last month to run that new **killer app** that management thought was so important.

Deadly Avenger: Really? What was wrong with the old hardware?

-- The Bunion --: It was too old to be used anymore, apparently. I told them the hardware requirements for the new CRM system BEFORE they bought it. Now they need to upgrade the whole internal network as well.

DeadlyAvenger: What are they gonna upgrade to?

-- The Bunion --: You won't believe it. A whole truckload of brand new Dell PowerEdge servers with Intel Xeon processors. And a 10 gbps ethernet connection hooking everything together.

DeadlyAvenger: Cool. That sounds like some real **leading edge** stuff.

-- The Bunion --: **Whatever**. We could have saved a lot of money by staying on the old software platform. It was good enough to make any CRM **geek** happy.

DeadlyAvenger: Let me get this straight. You're complaining about management buying new hardware? The old hardware was totally **EOL**.

-- The Bunion --: It's such a **PHB** move to spend so much on replacing a perfectly functioning CRM platform. The CTO probably read a shiny pamphlet at a trade show in San Francisco, had a few drinks over dinner, and then immediately signed the contract.

DeadlyAvenger: You might be stuck in the past, but the **code monkeys** in the developer room will be salivating over the new equipment. Now they can rewrite their **spaghetti code** into more object-oriented goodness.

-- The Bunion --: I don't mean to be a **pita**, but I checked out the user reviews of that new release and people say it's fundamentally **horked**. Newer does not always mean better. It doesn't pay to be an **early adopter**.

DeadlyAvenger: That may well be true. I thought maybe we should have hosted the new application in the cloud. That's what all the **trendsetters** are doing nowadays. And people from remote offices would like that because they would get quicker load times.

-- The Bunion --: The cloud? Are you serious? What about security? You are such a **fanboy** of every new IT fad that comes along.

DeadlyAvenger: At least I am not a **troll**. You are trying to start a **flame** war on the company intranet discussion forum in order to show upper management how much you are trying to save money.

-- The Bunion --: Well, we didn't exactly have a great financial year so far. Now the whole IT department will be busy upgrading instead of solving our customers' true needs.

DeadlyAvenger: Admit it. If it were up to you, we'd all still be on mainframes programming COBOL.

-- The Bunion --: Hahahaha. Yeah I guess you're right. But saving the company money is the only effective way I know of proving I deserve a raise every year.

DeadlyAvenger: Or you could end up proving that you are a cranky old man and get yourself relocated to the Accounting department ;)

-- The Bunion --: **Whatever**. You guys wouldn't last two weeks without me here!

DeadlyAvenger: Maybe you're right. I gotta go now. Bye.

-- The Bunion --: Howdy.

Exercise 4. Pick the correct answer from a list according to the text above.

1. Pick the correct answer from a list according to the text above.

1. An early adopter is someone who’s obsessed with leading edge technology

a) True b) False

2. Every well-coded app has its fair share of spaghetti code.

a) True b) False

3. Trolls are real pita until the world's EOL.

a) True b) False

Part2

Exercise 1. Think and answer:

Think about situations when certain IT Slang is appropriate and inappropriate?

Exercise 2. Study the list of terms and examples of their usage below:

1337(елітний) **-** one of the best hackers or coders that you know.

You should really watch out for SatanSpawn666, he is totally 1337.

alpha geek (альфа-гік) **-** the guy the other engineers go to when they have a problem they can't solve.

I can't find the source of this bug, so I guess I will ask the alpha geek what he thinks.

ASAP (as soon as possible) (якнайшвидше) **-** now, even yesterday if possible.

Bill Lumbergh told Peter Gibbons that he needed those TPS reports on his desk ASAP!

AFK (away from keyboard) (відійшов від клавіатури) **-** busy in the 'real world'; often used in chats after a long pause in a chats.

Mickey453: sorry was just afk!

BRB (be right back) (зараз повернуся) **-** something you say in a chat message to show you will need to go to the bathroom or go grab a Coke.

i gotta go ... brb!

FTW (for the win) (для перемоги) **-** something to say when you are trying to accomplish something important and you are very near to your goal.

Lizard Team attack the base FTW!

IMHO (in my humble opinion) (на мою скромну думку) **-** something to put in a chat before stating something which is subjective (not a fact).

IMHO software developers should make more money than lawyers.

l8r (later) (пізніше) - something you say when you are leaving a chat.

cu l8r alig8r!

LOL (laughing out loud) (голосно сміюсь) **-** something to type when someone else types something funny.

Dirk: My little brother just swallowed a bug in his soda, lol.

Leech (пиявка) **-** a derogatory term for a person who uses download bandwidth without contributing appropriate upload bandwidth

Many bittorrent users are essentially leeches who are only there to download pirated material.

n00b (нуб) **-** someone who can't hack or program very well.

Bill is such a n00b he can't even navigate the command line without a book.

ROTFL (rolling on the floor laughing) (регочу)**-** something to type when lol simply isn't enough.

You know I was ROTFL when I heard my ex-girlfriend dropped her new iPhone in the toilet.

TIL (today I learned) (сьогодні дізнався) **- s**omething to type in a chat before stating a recently acquired humorous or educational fact.

TIL that sharks cannot stop swimming or they will drown.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Solving a Customer's Server Problem

People often use funny slang and abbreviations in chats and other social situations. Typically, there is also a more relaxed attitude towards grammar as well. The following is a chat transcript between an account manager and a software engineer working at a web hosting company. They are troubleshooting a server problem that affects one of their customers.

Please note that you should not use slang with unfamiliar people. Using slang is not professional and can be taken badly by some people. You have been warned!

Account Manager: **TIL** that certain web page requests are taking forever to load on some of our servers. One client in particular is really angry.

Sysadmin: what is the name of the client?

Account Manager: Svensson Trucking Supply

Sysadmin: **brb** i need to be **afk** for 30 minutes

Account Manager: 30 minutes? this guy is a major client and management says we need to fix this issue **ASAP**. they told me you were the **1337 alpha geek** who could resolve this issue right away.

Sysadmin: **lol**... 1337 alpha geeks need to eat lunch too :P

Account Manager: **LOL** that you would eat lunch right now. Please look into the problem BEFORE lunch!

Sysadmin: okay gimme 5 minutes

Time passes...

Account Manager: Did you find the problem?

Sysadmin: wow did i... **imho** the site was set up by a total **n00b**. there is a major security hole in their iis web server. someone set up a warez site on a subdomain and there are 100 **leech**es sucking up all the bandwidth.

Account Manager: No wonder their site is so slow

Sysadmin: they are using a really old version of iis server. let me upgrade and patch it for them.

Account Manager: Okay. I will call the client and update them on the status.

Time passes...

Sysadmin: okay the server is patched and running normally. now im gonna go eat... 1337 alpha geek **ftw**!

Account Manager: Nice work! Thanks a million!!!!

Sysadmin: how about telling management to give me a raise since im the only one who knows how to solve client problems?

Account Manager: Wow that's funny. I'm literally **ROTFL**

Sysadmin: Get lost.

Account Manager: Hahahaha! I'll pretend I didn't hear that. **L8R** skater. (Adapted from https://www.english4it.com/module/core/unit/20/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1. Pick the correct answer from a list according to the text above.

1. TIL that a n00b is not very 1337.

a) True b) False

2. AFAIK, most people actually roll on the floor when typing ROTFL.

a) True b) False

3. An alpha geek will never be AFK.

a) True b) False

Exercise 5. Translate these sentences into English paying attention to IT slang.

1. «**Що завгодно**, чувак,» — відповів програміст, коли його менеджер попросив його працювати понаднормово, щоб відновити втрачені файли.
2. Популярність форуму серед серйозних користувачів знизилась, коли системний адміністратор відмовився забанити **тролів.**
3. Вебсайти впали, бо сервер був повністю **зламаний.**
4. Том такий **фанат**, що досі використовує свій старий Commodore Amiga як основний комп'ютер.
5. Чи вважаєте ви, що компільовані мови досягли **кінця свого терміну експлуатації**?
6. Білл Ламберг сказав Пітеру Гіббонсу, що йому потрібно отримати ті TPS-репорти на стіл **якнайшвидше**!
7. Білл такий **нуб**, що навіть не може навігувати командним рядком без книги.
8. **Сьогодні я дізнався**, що акули не можуть зупинятися, інакше вони потонуть.

Writing Activity

Exercise 6. Pick a question and write about it.

1. Write a text massage to your colleague using texting slang.

- To remind him/her about a task that needs to be completed
- To ask for advice with your tasks
- To complain about some difficulties with your task

2. Work with your partner to create some TNGAs (The New Great Abbreviations). Think of the phrases or words in English you use very often and TNGAs will help you

to save time or texting space. Be ready to advertise these abbreviating to the other groups.

Speaking Activity

Exercise 7. Discuss the questions.

- Some people say that people who use slang are lazy. Do you agree?
- Do you have similar slang words and expressions in your language?
- How difficult is it to use slang in a foreign language?
- Do you need to use slang to communicate with non-native speakers?
- If you want to find the meaning of a new word what do you do?
- How is your language influenced by other languages?

Unit 13. English for Electronics

Exercise 1. Think and answer:

Which electronics terminology do you remember from school?

Exercise 2. Study the list of terms and examples of their usage below:

Capacitor (конденсатор) - an electrical component which uses electrostatics to store energy

The flash bulb in my camera has an old capacitor and takes a long time to charge up.

Current (струм) - a flow of electric charge (such as electrons or ions)

You can experience the current flow in a copper wire if you hook one end up to a voltage source.

Electrical circuit (електрична схема) - a network or interconnection of electric components in a closed loop

This electrical circuit is not in a closed loop and that is why the current does not flow to the bulb.

EMF (electromotive force) електрорушійна сила) - an energy measured in volts; refers to the potential which causes the flow of electrons (and ions) in a conductor

A simple battery is the most common source of electromotive force in electronic toys, but some modern toys even use solar cells to power up!

IC (integrated circuit) (інтегральна схема)- a set of electronic interconnections in a small «chip» or plate where all the elements are inseparable

ICs get smaller and smaller ever year due to the rapid development of nanotechnology - today there are billions of transistors inside a typical chip.

Light bulb (лампочка) - a source of electric light which uses a filament stored in a glass enclosure

I need a larger light bulb in my table lamp - it is too dark for me when I study at night.

LED (light-emitting diode) (світлодіод) - a semiconductor source of light

LEDs are an increasingly popular source of light in homes nowadays because they consume much less energy than the older style incandescent light bulbs.

Path (шлях, доріжка, проводка) - a continuous trace or wire in electric circuit which connects various components

After a long investigation I found a break in the path which connected the bulb and the battery - that is why the circuit didn't work.

PCB (printed circuit board) (друкована плата) **-** a laminate board which supports electric components in the circuit

We order ready-made PCBs without any components, so our students can practice their soldering skills.

Resistor (резистор) **-** an electrical component, which restricts the flow of electrons in the circuit.

If you add another resistor to your serial connection just before the bulb, the light will be not so blinding.

Solar cell (сонячна батарея) **-** an electrical device which converts the energy from light into electricity

There is a small solar cell in my watch, thanks to that it was working even after the battery died.

Soldering iron (паяльник) **-** a handy tool used to assemble the electrical components on a PCB (printed circuit board).

My home soldering iron is really handy but at work we use soldering stations which allow us to control the heating temperature.

Voltage (напруга) **-** also known as «electric tension», is a difference between electric potential between two points

If there is a voltage between two points of a circuit, and you connect these points with a wire, then the current will flow.

Voltaic cell (гальванічний елемент) – a device that derives energy from chemical reaction, such as in a battery

Car batteries are a good example of voltaic cells: there are electrodes which are submerged in a solution, and due to some chemical reaction which takes place in the solution, the energy is derived.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Basic electronic components

Nowadays most of us cannot imagine life without the electronic devices which surround us. They are everywhere, and often we even forget about their existence: mobile phones, TV sets, mp3 players, electronic watches or even alarm clocks. Most people do not reflect on how and why these everyday appliances work. But being a geek requires at least some basic technical knowledge about the surrounding world.

As you probably remember from early science lessons, the electric **current** is an organized flow of electrons and ions in a conductor. But to make those tiny particles move, there must be some kind of force.

In electronics, this force is called an electromotive force or EMF, which may be also described as voltage. For some, the definition of voltage as the difference of potential in the conductor may be a bit hard to imagine.

So to visualize this process, you may think of the difference in the number of electrons in the conductor. If there are fewer electrons in one part of a wire than another part, then there is a difference of electrons between points A and B of the electrical circuit.

Nature is organized in such a way that it aims at making this potential equal in both points of the circuit. So to make it equal, there is a need to move some of the electrons from one place to another. Current is exactly what we call this movement.

Of course, we need to take our voltage from some electric source. One common source of power in smaller home appliances is batteries, or voltaic cells. The energy stored in batteries comes from chemical reactions that take place inside. But quite often we may replace the batteries with another source of energy, such as a solar cell, which may be found in simple calculators or electronic watches.

So, right now our circuit consists of a source (battery) and a **path** (a conductor in which the electrons flow). We need one more element that will consume the energy stored in the source of voltage. This element, called a load, could be a light bulb. Light bulbs contain a filament inside. The more electrons that flow through the bulb, the hotter the filament gets. At some point, the filament starts to glow with light. But sometimes this light might be too bright for us and we would like to dim it.

To dim the light, it is enough to add an additional element that will take away some energy from our circuit − a **resistor**. The greater the value (resistance) of this element, the bigger drop in **voltage** we experience. So, our first **electrical circuit** is done.

To assemble it we may use a printed circuit board or **PCB**, a laminate board with paths printed on it and the holes drilled to make place for electronic components. To attach the elements to the PCB we need a hand tool called a **soldering iron** or a soldering gun. This device heats the solder and enables us to attach our elements to the path.

Everyday appliances consist of much more sophisticated electronic components than our circuit, like for example **capacitor**s which can store energy for a short time. Capacitors are used for example in photo cameras as a source of energy for flash bulbs. Capacitors store voltage from batteries, and when needed the load is discharged rapidly so you can see the burst of light. There are also light-emitting diodes which are more and more common sources of light in our homes.

Due to the vast development in technology, electronic circuits get smaller and smaller every year. Usually they can be found as integrated circuits or IC - small chips that consist of various components that are inseparable. However, to fully understand the way electronic devices work, it is perhaps best to learn the basics of electronics with simpler and larger models. There is no substitute for manual practice under the supervision of experienced users of the soldering gun. (Adapted from https://www.english4it.com/module/core/unit/21/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1. What is the main function of electromotive force (EMF) in an electrical circuit?

a) To store energy in a battery

b) To move electrons and create current

c) To control the resistance in the circuit

d) To make the circuit light up

2. What component is used to dim the light in an electrical circuit?

a) Battery

b) Capacitor

c) Resistor

d) Light-emitting diode

3. What is the role of a capacitor in electronic devices?

a) To store energy for a short period and release it quickly

b) To convert electrical energy into light

c) To increase the flow of electrons

d) To protect circuits from overcurrent

Exercise 5. Translate this text into English.

Сучасні мобільні телефони є складними електронними пристроями, що містять різноманітні компоненти, які забезпечують їх функціональність. Ось деякі з основних електронних компонентів мобільного телефону:

1. **Процесор (Центральний процесор або CPU)**

Це «мозок» мобільного телефону, який обробляє всі команди та виконує обчислення. Він керує всіма іншими компонентами телефону і забезпечує його роботу.

2. **Оперативна пам'ять (RAM)**

Оперативна пам'ять використовується для тимчасового зберігання даних, які потрібні телефону для виконання поточних операцій. Чим більше RAM, тим швидше і ефективніше працює пристрій.

3. **Пам'ять для зберігання (ROM, Flash Memory)**

Це пам'ять для довготривалого зберігання операційної системи, додатків та особистих файлів, таких як фотографії та відео. У сучасних телефонах зазвичай використовують флеш-пам'ять для зберігання даних.

4. **Батарея**

Батарея постачає енергію для роботи всіх компонентів телефону. Вона зазвичай є літієвою або літій-полімерною і має здатність швидко заряджатися та тривало працювати.

5. **Дисплей (екран)**

Мобільні телефони зазвичай використовують рідкокристалічні (LCD) або органічні світлодіоди (OLED) екрани для відображення інформації. Вони відповідають за візуальне сприйняття і мають високий рівень яскравості та контрасту.

6. **Модуль зв'язку (антенна, Wi-Fi, Bluetooth, 4G/5G)**

Для підключення до мережі мобільного зв'язку, Wi-Fi, Bluetooth та інших бездротових технологій використовуються відповідні модулі. Вони дозволяють телефону передавати та отримувати дані.

7. **Камери**

Більшість мобільних телефонів мають вбудовані камери з різною кількістю мегапікселів, що дозволяють робити фотографії та відео. Камери зазвичай мають також додаткові функції, такі як автофокус та стабілізація зображення.

8. **Сенсори**

Телефони оснащені різними сенсорами, такими як акселерометр (для вимірювання руху), гіроскоп (для визначення орієнтації пристрою), датчик відбитків пальців, барометр, датчик наближення та інші.

9. **Мікрофон і динамік**

Мікрофон дозволяє записувати звук під час розмов, а динамік використовується для відтворення звуків, таких як музика, дзвінки або повідомлення.

10. **Мікросхеми живлення (PMIC)**

Ці мікросхеми відповідають за розподіл енергії в телефоні, забезпечуючи необхідну напругу для кожного компонента та регулюючи процес заряджання батареї.

Усі ці компоненти працюють разом, забезпечуючи нормальну роботу мобільного телефону та його здатність виконувати різноманітні завдання, від дзвінків до інтернет-серфінгу та ігор.

Writing Activity

Exercise 6. Pick a question and write about it.

1. What kinds of skills are necessary to practice electronics? How safe it is?

2. What is, in your opinion, the future of electronic devices? Are they making our life easier or are they dangerous?

Speaking Activity

Exercise 7. Discuss the questions.

1. How important are electronic devices in our lives?

2. Why is it necessary to know at least some basic technical knowledge about the surrounding world?

3. Can you describe the principle of any devices work?

Unit 14. Computer Ethics

Exercise 1. Think and answer:

Why is computer ethics so important in IT?

Exercise 2. Study the list of terms and examples of their usage below:

Commandment (заповідь) **-** a rule to be observed as strictly

The CTO commanded the project managers to ship the product by the end of the year or face the consequences.

CEI (Computer Ethics Institute) (Інститут комп'ютерної етики) **-** a nonprofit research, education, and public policy organization focused on the issues, dilemmas, and challenges of advancing information technology within ethical frameworks

The CEI's mission is to facilitate the examination and recognition of ethical issues in the development and use of modern information technologies.

Consent (дозвіл) **-** permission, allowing

I needed written consent from my boss to change the security settings on the server.

Consequences (наслідки) **-** results; things that happen as a result of a previous action

The server admin was unaware of the dire consequences of installing the wrong version of Apache web server.

Copyright infringement (порушення авторських прав) **-** to use somebody's content without proper permission

The webmaster committed copyright infringement when he copied an article from another website.

Cracking (крэкінг) **-** criminal hacking; exploiting security holes to gain unauthiorized access to computer systems

Ethically speaking, cracking is no different from other forms of illegal trespassing or stealing.

Dependent (залежний) **-** decided or controlled by something else

The software utility was dependent on a specific architecture of CPU in order to run at optimal speed.

Encounter (зустріч) **-** to meet or come across

If you meet an alien from another planet it is called a «close encounter of the third kind».

Ethics (етика) - a branch of philosophy that addresses questions about morality (what is right and wrong)

As computers become more powerful the need for computer ethics classes is increasing.

Internet privacy (приватність в Інтернеті) - general techniques and technologies used to protect sensitive data, communications, and preferences

The US government routinely tries to expand and restrict Internet privacy laws.

Norbert Wiener (Норберт Вінéр) - (1894–1964) mathematician and cybernetics pioneer; considered the forefather of Computer Ethics

The Norbert Wiener Award for Social and Professional Responsibility is awarded annually by CPSR.

Phishing (фішинг) - obtaining sensitive information through a deceptive email campaign, while pretending to be a trusted person or business

The man thought his bank was asking him to change his password, but it was actually a phishing scam trying to trick him into revealing it.

Photoshopping (фотошопінг) - digitally transforming or altering a photograph in order to misrepresent the original

Magazines routinely photoshop pictures of models in order to remove pimples, moles, and wrinkles.

Privacy policy (політика конфіденційності) - a statement or a legal document that discloses the ways an organization gathers, uses, and stores the end user's data.

Every website that collects user information should have their privacy policy posted online.

Reputation (репутація) - the common opinion about someone or something

Being hacked can negatively affect a company's online reputation.

Snoop (шпигувати) - to search for private information without permission

The IT security officer was caught snooping on an employee's private email account in order to steal her personal bank account number.

Social engineering (соціальна інженерія) - a broad term for any online or offline technique that uses deception to trick people into giving out confidential information

The hacker used social engineering methods to trick a bank teller into giving out another user's password.

Tedious (нудний) - boring, repetitive, or slow or long

Debugging code can be a tedious process without the correct knowledge and tools.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

The 10 Commandments of Computer Ethics

Ethics is a branch of philosophy that deals with determining what is right and wrong. Listed below are 10 **commandments** to help you understand computer ethics. Anyone working in the field of IT is going to **encounter** a lot of ethical problems. Why? Because computers are very powerful and IT professionals are like wizards controlling this technology. We all know from watching Harry Potter movies that wizards can be good or bad, and so can IT professionals. Non-technical people are increasingly **dependent** and trusting of information technology. If the next 20 years is anything like the last 20 years, the future is very bright for IT workers. This is why IT professionals need to study and practice computer ethics.

People rely on computers to handle the more **tedious** details of their social, business, and financial lives. Imagine what would happen if an enemy or a business competitor got access to your email account, bank account, facebook profile, and company website? They could destroy your **reputation** in a matter of hours. This is why we need to study Computer Ethics.

The idea of Computer Ethics was invented in 1950 by MIT professor Norbert Wiener. Norbert predicted that the world would undergo “a second industrial revolution” − an “automatic age” with “enormous potential for good and for evil”. Today we live in that world! Why did he say this? Airplane drones can bomb a target while the pilot is safe in a computer room thousands of miles away. Criminals can use **phishing** and other **social engineering** techniques steal your financial information while sitting safely on another country. Companies can create confusing privacy

policies to gather and use information about you without your knowledge or consent. Governments can block access to information using firewalls. These are all very serious ethical violations that happen every day.

The main concerns of computer ethics are: Internet privacy, **cracking**, copyright infringement, and Internet control. The Computer Ethics Institute (**CEI**) has laid these 10 commandments down for people to follow. Below each of the 10 Commandments, I have put some comments to help explain what the commandment means to me.

Rule №1: Thou shalt not use a computer to harm other people.

You should not program a computer to do dangerous things to people. For example, to program a robot to kill people, make viruses, or weapons of mass destruction. What this means is that computers are not an excuse to do bad things to people. The programmer is responsible for the actions of his programs.

Rule №2: Thou shalt not interfere with other people's computer work.

Everyone has the right to use computers. Likewise creating a computer virus is wrong because it wastes people’s time and money and destroys potentially valuable property. As the hippies used to say: “Live and let live!”

Rule №3: Thou shalt not **snoop** around in other people's computer files:

It is illegal to open and read someone's real physical mail, and it is also illegal to snoop on network connections and read someones email, passwords, files, databases, etc.

Rule №4) Thou shalt not use a computer to steal.

This one is obvious. Stealing is just as bad with a computer or without one. Use your IT powers for good!

Rule №5: Thou shalt not use a computer to bear false witness.

This commandment means that you shouldn’t use a computer to create false facts. For instance, **photoshopping** a picture to make an innocent man look guilty. Perhaps another equally important commandment should be, “Don’t believe everything you see on the Internet”. Just like in traditional media, it is a common practice to spread half-truths, exaggerations, lies, and rumors on the Internet. This is wrong.

Rule №6: Thou shalt not copy or use proprietary software for which you have not paid.

Think about how hard it is to write good software. Think about all the people and costs involved. You have two choices: 1) buy a license or 2) find an open source alternative. Using pirated software not only unethical it is dangerous because of hidden malware.

Rule №7: Thou shalt not use other people's computer resources. without authorization or proper compensation.

Is it ethical to use a neighbor's lawn mower or mixer without telling them? How about their Internet connection? No it isn't! Is it legal or ethical to do bad things on a computer while logged in as someone else? No it isn't! Don't do it!

Rule №8: Thou shalt not appropriate other people's intellectual output.

If you copy text or images from a website and post them on your own website it is a crime in most countries, and definitely not ethical. Why? You are causing irreparable damage to the creator of the content. Why not 'get a life' and create your own unique and original content? You will be much happier in the long run creating new things rather than stealing from other people.

Rule №9) Thou shalt think about the social **consequences** of the program you are writing or the system you are designing.

Do you write software that helps people to steal, kill, spy, gamble, or spread pornography? Please ask yourself why. Is there a way you can use your IT talents for good purposes? Wouldn't the world be a better place if everyone was engaged in positive behavior instead of negative behavior? Think about it!

№10 Thou shalt always use a computer in ways that ensure consideration and respect for your fellow humans.

Things that you do 'online' can have real effects in your 'offline' life as well. Perhaps Mark Zuckerberg, the founder of Facebook, said it best, "Having two identities for yourself is an example of a lack of integrity.»

Okay, that was a brief introduction to Computer Ethics. I hope you enjoyed it and it didn't sound too «preachy». I also hope that you continue to think about ethical

questions throughout your IT career. It’s very important. Your career and maybe even your life can be ruined by making the wrong ethical decisions. (Adapted from https://www.english4it.com/module/core/unit/22/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1. What is the main focus of computer ethics?

A) The development of new technologies

B) The ethical use and impact of computers and technology on society

C) The study of computer programming languages

D) The design of computer hardware

2. What is Rule №1 of the Computer Ethics Institute's commandments?

A) Thou shalt not steal intellectual property

B) Thou shalt not use a computer to harm other people

C) Thou shalt not copy proprietary software without permission

D) Thou shalt not snoop around in other people's computer files

3. According to the text, what is the consequence of using pirated software?

A) It is illegal but harmless

B) It can damage the computer's hardware

C) It is unethical and dangerous due to hidden malware

D) It is considered acceptable if it is free

Exercise 5. Translate this text into English.

Комп'ютерна етика − це галузь етики, що вивчає вплив інформаційних технологій на суспільство, а також моральні проблеми, що виникають у зв'язку з використанням комп'ютерів та Інтернету. Вона охоплює різноманітні аспекти, від захисту приватності та конфіденційності до питань авторських прав і зловживань технологіями.

Основні аспекти комп'ютерної етики:

1. **Приватність та захист даних** В умовах глобалізації та швидкого розвитку Інтернету особисті дані стали важливим об'єктом обробки та зберігання. Комп'ютерна етика піднімає питання, як забезпечити захист персональної інформації користувачів і запобігти несанкціонованому доступу до неї. Важливими є питання використання даних у рекламних цілях, а також можливість витоку інформації через хакерські атаки.

2. **Авторські права та піратство** Однією з головних етичних проблем у комп'ютерній сфері є порушення авторських прав. Це стосується як програмного забезпечення, так і медіа-контенту (музики, фільмів, книг тощо). Піратство − незаконне копіювання і поширення контенту без дозволу власника прав − є серйозною етичною проблемою, оскільки порушує права авторів та розробників.

3. **Використання технологій для шкоди** З розвитком комп'ютерних технологій з'являються нові форми зловживання, такі як кіберзлочинність, хакерські атаки, фішинг та інші. Всі ці форми незаконної діяльності не лише порушують закон, але й мають серйозні етичні наслідки, оскільки завдають шкоди іншим користувачам або організаціям.

4. **Віртуальна поведінка та анонімність** Інтернет дозволяє користувачам залишатися анонімними, що може призводити до неетичної поведінки в мережі, такої як хуліганство, тролінг або агресія. Комп'ютерна етика піднімає питання, як зберігати етичні стандарти поведінки в Інтернеті і як забезпечити відповідальність за свої вчинки в цифровому середовищі.

5. **Цифровий розрив і рівність доступу** З розвитком технологій виникає проблема цифрового розриву між різними країнами та соціальними групами. Люди, які не мають доступу до Інтернету або сучасних технологій, опиняються в нерівних умовах. Комп'ютерна етика передбачає створення рівних можливостей для всіх користувачів і боротьбу з нерівністю в доступі до інформаційних технологій.

6. **Штучний інтелект та автоматизація** Використання штучного інтелекту (ШІ) також викликає етичні питання. Це стосується не тільки

потенційної втрати робочих місць через автоматизацію, але й безпеки та контролю над ШІ-системами. Наприклад, хто нестиме відповідальність, якщо автономний автомобіль чи робот завдасть шкоди людині?

Отже**,** комп'ютерна етика є важливою частиною сучасного суспільства, адже вона допомагає зрозуміти моральні аспекти використання технологій і забезпечує безпеку, права людини та гідність в цифровому середовищі. Важливо, щоб кожен користувач технологій розумів свою відповідальність і діяв етично, щоб уникнути негативних наслідків для себе та інших.

Writing Activity

Exercise 6. Pick a question and write about it.

1. Do you believe in studying computer ethics, or do you think it is a waste of time? How can it help you?
2. Why is it necessary to think about the social consequences of the program you are writing or the system you are designing?

Speaking Activity

Exercise 7. Discuss the questions.

1. Why the US government tries to expand and restrict Internet privacy laws?
2. Anyone working in the field of IT is going to encounter many ethical problems. Why?
3. Norbert predicted that the world would undergo “a second industrial revolution” — an “automatic age” with “enormous potential for good and for evil”. Today we live in that world! Why did he say this?

Unit 15. American IT Companies

Exercise 1. Think and answer:

Which IT companies do you know? What are they famous for?

Exercise 2. Study the list of terms and examples of their usage below:

Adobe - San Jose, California multimedia software company founded in 1982 by *John Warnock and Charles Geschk; known for Flash, Photoshop and PDF software technologies*

Our Computer Design and Layout class professor advised us that most industry professionals use Adobe software.

AMD (Advanced Micro Devices) - Santa Clara, California company founded in 1969 by Jerry Sanders; known as the second largest manufacturer of x86 processors and graphics processors

AMD has a strange history of competition, cooperation, and litigation with the x86 platform creator Intel.

Amazon - Seattle company founded by Jeff Bezos in 1994 and specializing in mail order and and cloud computing

Amazon.com started off by selling books, but now they sell everything from food to server space.

Apple - Cupertino, California based computer company founded in 1976 by Steve Jobs and Steve Wozniak

Apple is a large consumer IT company known for its quality products with innovative design and usability concepts.

Cisco - San Jose, California company founded by Len Bosack, Sandy Lerner, and Richard Troiano in 1984 which sells networking and telecommunications products and services

In 2011, Cisco Systems announced it would shut down its consumer electronics devision.

Dell - Round Rock, Texas computer manufacturer founded by «Micheal» in 1984; known for its direct-sales approach, low cost, and highly configurable product offerings

Benjamin Curtis, the popular «Dude, you're getting a Dell» spokesperson was fired in 2003 after he was arrested on drug possession charges in New York City.

Facebook - Menlo Park, California company founded in 1994 by Mark Zuckerberg which calls itself a «social network» and focuses on advertising and interpersonal communication

Over 2 billion people use facebook to communicate with friends and family.

Google - Mountain View, California company founded by Larry Page and Sergei Brin in 1998; known for their search technology, Android mobile OS, web-based productivity tools, and cloud hosting services

Google's unofficial company motto is «Don't be evil», a phrase coined by engineer Paul Buchheit.

HP (Hewlett-Packard) - Palo Alto, California startup founded in a one-car garage in 1939 by two engineering students «Bill» and «Dave»; known for producing high quality computers printers, and other peripherals

HP specializes in manufacturing computers, printers, networking devices, and related services.

Intel - Santa Clara, California company founded in 1968 by Gordon Moore and Robert Noyce; known for manufacturing microprocessors and other high tech products

Intel invented the 4004, the world's first microprocessor in 1971.

IBM (International Business Machines) - Armonk, New York based company founded by Charles Ranlett Flint in 1911; known for building large scale business machines of nearly every type from typewriters to mainframes to personal computers

The man worked for IBM for so long he could remember the days when everyone wore white lab coats to work.

Microsoft - Redmond, Washington software company founded by Bill Gates and Larry Allen in 1977; known for Windows operating system, Office productivity software, and Azure cloud computing platform

The college student wanted to work for Microsoft because it was the biggest and richest software company in the region.

Nvidia - Santa Clara, California company founded by Jensen Huang, Chris Malachowsky, and Curtis Priem in 1993; known for producing GPUs (graphical processing units) for gaming, cryptocurrency mining, and media creation purposes
NVIDIA is very popular with gamers and responsible for the Geforce line of graphics cards.

Oracle - Redwood Shores, California company founded in 1977 by Larry Ellison and known for developing database software and supply chain management software
Oracle Corporation has the best overall reputation for scalability and reliability in the database industry.

Symantec - Mountain View, California software company founded in 1982 by Gary Hendrix; specializing in artificial intelligence and antivirus software
Symantec is famous for their antivirus software which protects computers from malware.

Western Digital - San Jose, California company founded in 1970 by Alvin B. Phillips; the world's largest manufacturer of computer storage devices
In 2017, Western Digital was the first company to ship a 14 terabyte hard drive.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Top 15 American IT Companies

For some reason, most of the large IT companies in the world are American. For instance **Google, Apple, Microsoft,** Facebook, etc. As an IT professional, you are expected to know about big IT corporations, revenue, profits, budgets, and other economic matters.

Why?

Because these things affect your capacity to perform your tasks. The economy tends to go in cycles. In an up cycle, you will get resources to build or improve systems. In a cycle you will be pressured to cut spending and possibly even lay off

staff. This is business as usual» and nothing to get excited about. Over the span of your career you will experience many up and down economic cycles.

Studying what the big IT companies do right and wrong will help you think of ideas how to improve your own IT department. No matter if you work for a small company, non-profit organization, or a government institution, you will learn quickly that most decisions are based on money. The best way to implement your «great idea» is to argue that it will save time or increase profits. Trying to argue a point without keeping efficiency, profits and budgets in mind will be difficult.

Below is a chart of 15 American IT companies I think are important to know about. The more you know about the business of IT, the better IT professional you will become.

Company	Rank	Revenue	Profit	Products
Adobe	443	5,854	871	Graphic Design Software / Flash Media
Advanced Micro Devices	565	4,272	471	CPU / GPU
Apple	3	215,639	14,013	Computer Hardware / Software
Cisco Systems	60	49,247	7,767	Networking Equipment
Dell	41	64,806	2,635	Computer Hardware
Google	27	90,272	8,505	Search / Software
HP	61	48,238	8,761	Computer Hardware / Services
IBM	32	79,919	14,833	Computer Services / Software / Hardware
Intel	47	59,387	11,464	Microprocessors, Components
Microsoft	28	85,320	18,760	Computer Software
Oracle	81	37,047	6,135	Database Software / Services
NVIDIA	387	6,910	-30	GPU / CPU / chipsets
Symantec	550	4,888	1,138	Antivirus / Security Software

Western Digital	217	12,994	1,382	Hard Disk Drives / Computer Hardware

Exercise 4. Pick the correct answer from a list according to the text above.

1. Why is it important for IT professionals to understand the business side of large IT companies?

a) It helps them develop better technical skills.

b) It impacts their ability to perform tasks, especially during economic cycles.

c) It allows them to design more user-friendly products.

d) It helps them avoid working in big corporations.

2. Which of the following companies is ranked 3rd in terms of revenue according to the chart?

a) Apple

b) Google

c) Microsoft

d) HP

3. What is the primary product of the company NVIDIA?

a) Antivirus software

b) Microprocessors

c) GPU / CPU / chipsets

d) Networking equipment

Exercise 5. Translate this text into English.

Google − це одна з найбільших і найвідоміших технологічних компаній у світі, яка надає послуги в галузі пошукових систем, онлайн-реклами, хмарних обчислень та багатьох інших технологічних рішень. Вона була заснована в 1998 році Ларрі Пейджем та Сергієм Бріном, студентами Стенфордського університету, і спочатку мала на меті створити більш ефективний пошуковий алгоритм для Інтернету.

Основною послугою Google є пошукова система, яка дозволяє користувачам знаходити вебсайти, зображення, відео та інші ресурси в Інтернеті. За роки розвитку компанія значно розширила свої можливості, додавши безліч додаткових інструментів, таких як Gmail − популярний сервіс електронної пошти, Google Maps − картографічний сервіс, Google Drive − хмарне сховище для файлів, а також операційну систему Android для мобільних пристроїв.

Google також активно розвиває штучний інтелект, машинне навчання та інші передові технології. Крім того, компанія займається розробкою автономних транспортних засобів і вивчає інші інноваційні напрямки, як-от квантові обчислення.

Сьогодні Google є частиною компанії Alphabet Inc., яка була створена в 2015 році для управління різними бізнесами та проектами, що не відносяться безпосередньо до основної діяльності Google. Це дозволяє компанії зосереджуватися на інноваціях і розвитку нових технологій, водночас підтримуючи свій основний бізнес − пошукові послуги та онлайн-рекламу.

Writing Activity

Exercise 5. Pick a question and write about it.

1. Which company has the most revenue? Why do you think they have so much revenue? Is this number likely to increase or decrease in the future?
2. Why do you think most of the big IT companies in the world are American?

Speaking Activity

Exercise 6. Discuss the questions.

1. Why do you think more and more people prefer Apple products? Is it really popular for its innovations or you are just paying for the brand name?
2. Do you know any unusual facts about American IT Companies?
3. Symantec is famous for their antivirus software which protects computers from malware. What do you think about it? Have you ever used it?

Unit 16. English for Programmers

Exercise 1. Think and answer:

To what extent is English important for IT specialists?

Exercise 2. Study the list of terms and examples of their usage below:

Argument (аргумент) **-** a value passed to a function when it is called

Value1 and value2 are both arguments in the expression sum(value1, value2).

Array (масив) **-** a data structure such as a variable that holds other variables in a particular order; For example: $a = ($b, $c, $d)

The programmer decided to store all her variables in an array so she could loop through them and perform a callback function.

Callback (функція зворотного виклику) **-** code that is passed as an argument to other code

Programmer use callbacks in tedious situations such as asynchronous programming or running an array of values through a function designed to take a string.

For loop (цикл for) - a sequence of instructions set to be repeated a specified amount of times until a condition is met

The student programmed the for loop to run until the value of «x» was greater or equal to 100.

Function (функція) **-** a named sequence of procedural code statements that perform a certain task

The CS student spent all night working on a function to compute the average rainfall in Alaska.

Global scope (глобальна область видимості) **-** usable anywhere in a program

The veteran programmer said using global scope variables is easier but sometimes less secure.

Goto statement (інструкція goto) **-** an instruction in older procedural programming languages that specifies that the instruction execution is to jump to specified location, normally a line number

The professor said that using goto statements is considered bad practice with the possible exception of breaking out of nested loops.

If else statement (умовний оператор if/else) - a decision making process wherein a block of code may or may not be executed based on pre-existing condition

The professor told us the heart of the program logic was simply a long line of if/else statements.

Local scope (локальна область видимості) - usable only in a limited section of program text such as a function

For security reasons the bank password variable had a local scope and was confined to a single function.

Operand (операнд) - a value from which an operator derives another value

«2» and «1» are examples of an operand in the equation «2 + 1 = 3».

Operator (оператор) - a symbol that represents an arithmetical calculation

The plus sign (+) is the operator in the mathematical expression «1 + 3 = 4».

Parameter (параметр) - a variable specified inside a function or subroutine definition which may be set by the code which calls it

The program returned a fatal error because a required parameter was missing.

Regex (regular expression) (регулярний вираз) - a string that describes or matches a set of strings, according to certain syntax rules

Search engines use regular expressions to match many kinds of similar words, even misspelled ones.

Statement (оператор) - a sentence of code in a programming language, usually ending with a semi-colon

A computer program is simply a collection of valid statements which provide a result.

String (рядок) - a series of letters and/or numbers

The teacher asked us to type a search string into the Google text box and press «enter».

Subroutine (підпрограма) - an independent block of code, separate from the main program, which performs a specific task necessary to the program

The Pascal software program was buggy due an incorrectly performing subroutine.

Switch (оператор switch) - a block of code or function causing a program to change its default behavior

The programmer preferred using a switch statement to many if/else statements.

Variable (змінна) **-** a symbol used to represent data which can be changed while the program is running.

Without variables, programs would not be able to accept user input.

While loop (цикл while) **-** a sequence of instructions set to continue until a certain expression is false

The while loop continued until the value of «x» was greater than «y».

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Dialogue: Procedural Programming is not Rocket Science

Dialogue Between a Senior Programmer and a Junior Programmer working at an aerospace startup in Los Angeles, CA.

Commander Keen: Hey kid, how is that new integration going for the booster rocket project?

Space Cadet: Huh? What? Okay I guess.

Commander Keen: So will it be ready to demo next week as scheduled? The client really thinks this product will take off.

Space Cadet: Ha ha. I get it, take off. No wait. It's going terrible. I guess I need some help.

Commander Keen: Okay, what's up?

Space Cadet: Well, firstly, this **statement** is always returning the wrong answer. Why?

Commander Keen: Okay let me see here. Well for starters, you're using single equals (=) signs instead of double equals (==) signs. Assignment operators always return the value that is assigned to the left **operand**.

Space Cadet: Oh yeah. Stupid mistake. Thanks. I guess that's why we still need to employ old people who remember programming the Motorola 6502.

Commander Keen: The 6502. It was a great chip Don't laugh... you could write an entire operating system in 2K of Assembly on that baby. No unit tests though.

Space Cadet: Gee, no wonder the space shuttle crashed 3 times.

Commander Keen: Twice. It only crashed twice. And there was no Motorola 6502 on board. It was an Intel 8086. And we had tests. But it crashed anyway. Twice. I still have nightmares.

Space Cadet: Oh spare me the drama! Well anyway, my second problem is that this function you wrote always seems to return false … no matter what parameters I send in! What gives?

Commander Keen: Weird, dude. Ummm… did you try to dump the **arguments** as they come in and output them to a log?

Space Cadet: Yeah! But they are all null values! I think something is wrong with the compiler. Or maybe we got hacked. Or maybe even… aliens!

Commander Keen: Yeah, right. Possibly aliens. Or maybe you can't code. Let me see here… AHA! These input **variables** are supposed to be strings but you are sending in arrays. That seems to be one of your problems, anyway.

Space Cadet: Strings? Okay. But how do I get the values I need out the **array**?

Commander Keen: You iterate over them and pull them out of the array, duh. It's not rocket science, you know.

Space Cadet: Yeah but the Iterator class I want to use is not in this function's local scope. And I don't want to pollute the global scope with unnecessary cruft.

Commander Keen: Iterator Class? Stop being such a Java nerd and use a procedural function like a for loop, or a while loop.

Space Cadet: Ugh. I hate procedural code. We might as well be using goto statements and COBOL subroutines at this point. Would that make you happy?

Commander Keen: Arrgghh! I can't stand young people whining… if you HAVE to use an OO Iterator then just do a callback to the main object's static function in the global scope. How did you get this job, anyway?

Space Cadet: What? Me? I thought they brought me in to clean up after you! During my interview they asked me to review your code on the Ballistics project. There was a switch statement with 1,000 lines, each with a tiny variation. We had a good laugh over that one. Ever hear of regular expressions, dude?

Commander Keen: Listen mate, that code was still in development. It's easier to tweak things if you can actually read it. Regular Expressions are great, but they are not always very readable or testable. And I'll have you know, I refactored that code the very next day, as planned. I got it down to 35 lines.

Space Cadet: Well, at least you're not being defensive about it.

Commander Keen: Okay yes, sarcasm. I recognize it. I am also human and a bit defensive when people criticize my code. But that's not the point. You need to see the bigger picture. At the end of the day, reality is not an object. It's just a giant if else statement going on into infinity. That's how I see it, anyway.

Space Cadet: Deep. Maybe you should have been a philosopher instead of an aerospace engineer.

Commander Keen: Hardy har har. Maybe you should learn more C and stop doing everything in Java. OO is shit. It's 90% boilerplate and 10% action. Inheritance is just a fancy lie to keep Computer Science professors employed. At the end of the day, it wouldn't hurt you to know a little Assembly language as well! There's no school like the old school!

Space Cadet: Yeah, but Java is actually written in C++, and C++ is written in C, and C was originally written in Assembly. So I guess in some ways I already am an assembly language programmer. After all, old man, we are all standing on the shoulders of giants.

Commander Keen: Wow, I never thought of it that way, kid. Deep. I guess we are both philosophers then.

Space Cadet: Yeah… and all this coding and philosophizing is making me hungry… let's go iterate ourselves over some tacos and beer. I'll treat for helping out with that stupid assignment error.

Commander Keen: Wow. Thanks. And I'm starving… so that's an iteration I can really sink my teeth into.

Space Cadet: ugh (groans at bad pun). (Adapted from https://www.english4it.com/module/core/unit/24/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1. What is the main reason for the Space Cadet's problem with his code in the beginning?

a) Incorrect usage of the for loop

b) Using single equals (=) instead of double equals (==)

c) Missing semicolons

d) Incorrect use of the Iterator class

2. How does Commander Keen suggest Space Cadet fix the problem with null values being passed into the function?

a) Use a debugger to trace the error

b) Check the function’s arguments and ensure they are strings, not arrays

c) Rewrite the entire function

d) Use a regular expression to handle the parameters

3. What does Space Cadet think about the use of procedural code like for loops or while loops?

a) He prefers them over object-oriented code.

b) He finds them outdated and prefers object-oriented approaches.

c) He doesn’t understand them.

d) He suggests using them exclusively in all projects.

Exercise 5. Translate this text into English.

Процедурне програмування − це парадигма програмування, яка базується на концепції виклику процедур, також відомих як підпрограми, функції чи методи. У цьому підході код організований у вигляді послідовності процедур, які працюють з даними. Це одна з найдавніших та найпростіших форм

програмування, яка широко використовується в різних сферах, особливо там, де важливі продуктивність та простота.

Основні концепції процедурного програмування

1. **Процедури (функції/підпрограми)**
    - Процедура − це багаторазовий блок коду, який виконує певну задачу.
    - Вона може приймати вхідні дані (параметри) та повертати результат.
    - Приклади: main(), print() або користувацькі функції, як-от calculateArea().
2. **Послідовне виконання**
    - Код зазвичай виконується у тому порядку, в якому він написаний, якщо інше не визначено структурами управління, такими як цикли чи умовні оператори.
3. **Модульність**
    - Код ділиться на менші, зручні для управління та повторного використання блоки − функції. Це спрощує тестування та обслуговування.
4. **Локальні та глобальні змінні**
    - Локальні змінні доступні лише в межах процедури, тоді як глобальні можуть бути використані в різних процедурах.
5. **Структури управління**
    - Включають цикли (for, while, do-while) та умовні оператори (if, else, switch), які дозволяють контролювати потік виконання програми.

Переваги процедурного програмування

- **Простота**: Легко зрозуміти та впровадити, особливо для новачків.
- **Повторне використання**: Функції можна використовувати повторно у різних частинах програми.
- **Продуктивність**: Прямий контроль над потоком програми та пам'яттю забезпечує швидке виконання.

- **Легке налагодження**: Модульна структура полегшує тестування та виправлення помилок.

Недоліки процедурного програмування

- **Масштабованість**: Складно застосовувати для великих, комплексних систем.
- **Уразливість даних**: Глобальні змінні можуть бути змінені будь-якою функцією, що може спричинити помилки.
- **Відсутність об’єктно-орієнтованості**: Складніше моделювати реальні об'єкти та їх взаємодії порівняно з об’єктно-орієнтованим програмуванням.

Приклади мов процедурного програмування

1. **C**
    - Відома своєю швидкістю та гнучкістю.
2. **BASIC**
    - Розроблена для початківців.
3. **Pascal**
    - Часто використовується для навчання основ програмування.
4. **Fortran**
    - Популярна в наукових обчисленнях.

Сфери використання

Процедурне програмування широко застосовується в системному програмуванні, вбудованих системах і там, де необхідна взаємодія з апаратним забезпеченням.

Приклади:

- Розробка операційних систем (наприклад, Unix написаний на C)
- Ігрові движки, що потребують високої продуктивності
- Наукові та математичні обчислення

Процедурне програмування, хоча й вважається класичним, залишається потужною та актуальною парадигмою. Володіння ним дає міцну основу для розуміння інших підходів, таких як об’єктно-орієнтоване чи функціональне програмування.

Writing Activity

Exercise 6. Pick a question and write about it.

1. Assembly language. What do you know about it?

2. Are there any alternatives to procedural and object-oriented programming?

Speaking Activity

Exercise 7. Pick a question and write about it.

1. The professor said that using goto statements is considered bad practice with the possible exception of breaking out of nested loops. Do you agree with him?

2. For security reasons the bank password variable had a local scope and was confined to a single function. Do you think this is the right decision?

Unit 17. What is Cryptocurrency?

Exercise 1. Think and answer:

What do you know about cryptocurrency?

Exercise 2. Study the list of terms and examples of their usage below:

Altcoin - any cryptocurrency other than Bitcoin

The majority of early altcoins were just forks of Bitcoin with minor changes to the proof-of-work algorithm.

Blockchain - a decentralised digital ledger of transactions maintained by consensus

The Bitcoin blockchain is maintained by a network of specialized computers designed for mining.

Cipher (шифр) **-** a coded message or secret way of writing

The boy had a difficult time decoding the cipher in which his father's diary was written.

Consensus process (процес консенсусу) **-** a group of peers responsible for maintaining a distributed ledger

A blockchain's consensus process is a vital part of its success because it must be fast, reliable, and verifiable by everyone using it.

Daps (decentralized applications) (децентралізовані додатки) **-** programs designed to work without a central owner, server, or controller

One can split dapps into three major types: financial transactions, traditional commerce, and "other" including everything from games to governance.

Double spend problem (проблема подвійного витрачання) **-** a single financial transaction potentially happening in two places at once

The double spend problem is difficult to solve without a blockchain or trusted third party like a bank.

Fiat - any currency established by a government without inherent value or scarcity such as the US dollar or EU euro

The value of fiat currency is tied to governmental authority, convenience, and trust in market forces such as supply and demand.

Fintech (фінансові технології) - innovation that aims to compete with traditional delivery of financial services

This rise of fintech has had a positive impact for society, including increased competition, a reduction in prices, and wider access to financial services.

Fungible (фунгібельний) - interchangeable and replaceable

Monero is a fungible cryptocurrency, meaning that every unit can be substituted by another unit without fear of a possibly tainted past.

Lambo (Lamborghini) - a sports car closely associated with cryptocurrency millionaire memes

Some early adopters of cryptocurrency are in love with blockchain technology, but others just want to be seen on Youtube driving a lambo.

Mining (майнінг) - being rewarded with cryptocurrency for providing routine bookkeeping and verification services

Cryptocoin mining is not really a reliable way to make substantial money for most people.

Money laundering (відмивання грошей) - the illegal process off making criminal income look legitimate

Successful criminals need to be experts at money laundering and often make deals with «cash only» businesses or crooked banks to conceal the true source of their income.

Mooning (місячний ріст) - the term when a crypto coin's market activity and price go way up in a short amount of time

ZoomCoin has been mooning for the past 24 hours, but it's likely just a «pump and dump scheme» and will fall back to Earth when reality hits.

Paper wallet (паперовий гаманець) - a printed version of a cryptocoin's private key

The woman did not trust technology so she printed out her paper wallet with 25 Bitcoin on it and put it in her safe.

Private key (приватний ключ) - a long random string of text used to create your digital signature

If someone knows your private key, they can fake your signature and steal your cryptocurrency.

Pump and dump (памп і дамп) **-** a coordinated effort to quickly raise the price of a coin and then sell it

A pump and dump is a coordinated scheme to make money from a failing or stagnant coin.

Satoshi (сатоші) - a common measurement of the worth of altcoins, measuring one hundred millionth of a Bitcoin.

The satoshi value of altcoins may vary, but one Bitcoin is always worth 100,000,000 satoshis.

Stable coin - a cryptocurrency whose sole purpose is to be pegged to a fiat pair like the USD, thus providing a safe place in times of volatility.

There is some debate over whether stablecoins are really backed by fiat and aren't just used to artificially pump other crypto prices.

The dark web (темна мережа) **-** the portion of the web that has been intentionally hidden and is inaccessible through standard web browsers

The dark web first became known as the home of The Silk Road marketplace which sold guns and drugs via mail and took payments in bitcoin.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

MegaHyperCoin, a Fintech Startup Short Story

Daniel took a sip of his coffee, taking in a deep breath. The coffee felt cold and a bit too sweet. Convincing his old friend about the new investment was taking longer than he expected. He sighed, set his cup of coffee down on the table, and decided to give it one last try.

«Look Martin, I know that you've heard stories about the dark web, **pump and dump** schemes, and money laundering. But, that's not what **blockchain** is all about. MegaHyperCoin can replace a lot of inefficient systems such as cross-border payments. And it can be **fungible** as well, making it just as anonymous and private as cold hard cash.»

Martin looked back at his friend with a raised eyebrow.

«OK, so let's say I go with your new **altcoin mining** idea. How do I know it's not another...»

«Bad coin?»

«Yeah, bad coin. You're showing me these figures but online people are talking about MegaHyperCoin mooning and how it's inevitable. Inevitable! Nothing is inevitable except the heat death of the universe. I know you believe in what you're doing, but I am still feeling highly skeptical. I've got a reputation to protect. I can't be seen shilling some new shitcoin.»

Daniel tapped his fingers on the table.

«Here is the thing. I am not asking you to shill anything. All I'm asking you to do is help with the mining. You've got some good computers doing nothing. It's not too late to jump in. I'm not saying that we'll be driving new lambos tomorrow either. We are just trying to make a couple hundred of dollars per day at first. We won't make any real money until there is mass adoption.»

Martin's face relaxed a bit.

«Go on.»

Daniel Continued.

«I understand why you're worried. The difference here is, we're dealing with two things: a well-known stable coin called Tether and a relatively new altcoin. While the market does tend to be volatile, we're sort of hedging our bets. In effect, we shouldn't be more at risk than anybody else trading regular **fiat** currencies.»

Martin's forehead wrinkled, making him look his age. The years he sacrificed to build up his business were beginning to show. He squinted his eyes and directed his gaze straight into Daniel's pupils.

«If I do this, it's not for the lambos and it's not to help an old friend. It's because I have nothing to lose and I need to learn more about this blockchain stuff. I believe **daps** might be a potential threat to my business one day. Everyone says that decentralized applications are the future. The only thing is, I don't understand what

they are talking about. I just need to understand this technology better. This project is how I can justify doing that.»

Daniel felt his pulse speed up. If Martin agreed to this deal, he would have 100 state-of-the-art computers working for him. And, he would be making money in sleep. He leaned closer and pushed a stack of paper in front of Martin.

«Here. Read this white paper. It explains everything, It shows how the **double spend problem** was finally solved in scalable fashion. It shows how to make a **paper wallet** just using a web browser and a common printer. That way you can keep your mad gains and private keys in cold storage and thus off the Internet. On page 15 there is a chart which explains the revolutionary new **consensus process** called proof of stake. Come on, if things start to look bearish we can always cut our losses and trade back into Bitcoin. Everyone could use a few more **satoshis** for a rainy day, am I right?'«

Martin's laughter filled the room. He could smell bullshit from a mile away. After all, one doesn't build a successful **fintech** company without some amount of sharp wit. Together with his team, he created a **cipher** that made is easier to make international transactions safely. He knew finance and software in the traditional sense, but cryptocurrency was still something mystical to him.

«Daniel, my friend... As long as you understand that this is not the traditional way I do things, then I'm on board with all this. I can sign these documents and do this deal, but this means that you take full responsibility for the outcome. I'll cover the hardware costs and the energy bills for the first 12 weeks. But if things aren't working out as you predicted after that time, well then you're on your own.»

«Yes, Sir! Thanks! That's more than enough time to get us going! You won't regret this.»

«I hope not. Any final details before we put pen to paper and close this thing?»

Daniel knew better than to say anything else but, «Just thank you». Right now, all that mattered was that he could start tomorrow.

Martin looked at his watch.

«Okay see you tomorrow, I've got to get to my kid's ballgame now.»

Daniel swallowed his last sip of cold coffee which had somehow turned from sickly sweet to gritty and bitter. He had a dark feeling that 12 weeks was not going to be enough time. (Adapted from https://www.english4it.com/module/core/unit/25/reading)

Exercise 4. Pick the correct answer from a list according to the text above.

1. What was Daniel's main goal in the conversation?

a. To sell Martin a new type of blockchain software.

b. To convince Martin to invest in MegaHyperCoin mining.

c. To promote Tether as a stable coin for trading.

d. To persuade Martin to learn about decentralized applications.

2. Why was Martin skeptical about MegaHyperCoin?

a. He had no interest in cryptocurrencies.

b. He believed blockchain technology was a scam.

c. He was concerned about its reputation and volatility.

d. He thought the technology was too outdated.

3. What agreement did Martin make regarding the project?

a. He agreed to cover all costs indefinitely.

b. He would fund the hardware and energy for 12 weeks.

c. He agreed to mine MegaHyperCoin on the condition of immediate profits.

d. He decided to buy and hold MegaHyperCoin without mining.

Exercise 5. Translate this text into English.

Криптовалюта − це цифрова або віртуальна валюта, яка використовує криптографію для забезпечення безпеки. На відміну від традиційних валют, таких як долар чи гривня, криптовалюти не мають фізичної форми і не контролюються центральними банками чи урядами. Вони працюють на основі

технології блокчейн, яка забезпечує прозорість, безпеку та децентралізоване управління.

Основні особливості криптовалюти:

1. **Децентралізація:**

Блокчейн — це розподілений реєстр, який знаходиться на багатьох комп'ютерах одночасно. Це означає, що немає центрального органу, який контролює криптовалюту.

2. **Анонімність:**

Транзакції з криптовалютою не прив'язані до особистої інформації користувача. Хоча всі операції є публічними, ідентифікувати користувачів складно.

3. **Криптографія:**

Для захисту транзакцій і створення нових монет використовується складна математична технологія.

4. **Обмежена кількість:**

Більшість криптовалют мають обмежену пропозицію. Наприклад, біткоїн має максимум 21 мільйон монет. Це робить криптовалюти подібними до рідкісних ресурсів, таких як золото.

Популярні криптовалюти:

1. **Біткоїн (Bitcoin):** Перша криптовалюта, створена у 2009 році. Вона є найпопулярнішою і найбільш впізнаваною.

2. **Ефіріум (Ethereum):** Відомий своєю платформою для створення децентралізованих додатків (DApps) і смарт-контрактів.

3. **Тезер (Tether):** Стабільна криптовалюта (стейблкоїн), яка прив'язана до вартості долара США.

Використання криптовалюти:

- **Інвестиції:** Багато людей купують криптовалюту як засіб збереження і примноження капіталу.
- **Оплата товарів і послуг:** Деякі компанії приймають криптовалюту як спосіб оплати.

- **Технологічні рішення:** Використовується для створення децентралізованих додатків, автоматизації бізнес-процесів через смарт-контракти тощо.

Переваги:

- Швидкі міжнародні перекази.
- Відсутність посередників, таких як банки.
- Захист від інфляції завдяки обмеженій пропозиції.

Недоліки:

- Волатильність (швидкі зміни ціни).
- Можливість використання у незаконній діяльності.
- Високі вимоги до електроенергії для майнінгу.

Криптовалюта змінює світ фінансів, пропонуючи нові можливості для інвесторів, бізнесу та простих користувачів. Вона відкриває двері до децентралізованого майбутнього, але також вимагає обережності та розуміння ризиків.

Writing Activity

Exercise 6. Pick a question and write about it.

1. Crypto-related websites and channels. What do you know about them? What interesting things could you learn for yourself?

2. What do you know about 'mining'? Do you think you could make substantial profits from mining?

Speaking Activity

Exercise 7. Pick a question and write about it.

1. What do you think is the advantage of the Mega Hyper Coin?

2. If someone knows your private key, they can fake your signature and steal your cryptocurrency.

3. Everyone says that decentralized applications are the future. What do you think?

Unit 18. Artificial Intelligence

Exercise 1. Think and answer:

What are pros and cons of AI becoming part of our life?

Exercise 2. Study the list of terms and examples of their usage below:

Diverse (різноманітний) –including many different types of people or things

IT teams are increasingly diverse, including experts from various fields to tackle complex problems.

Search engines (пошукові системи) – Software systems that search for information on the internet or in databases, allowing users to find content by entering keywords or queries.

Modern search engines use advanced algorithms to deliver accurate results within seconds.

Chatbots (чат-боти) – Computer programs designed to simulate conversation with human users, often used in customer service or online assistance.

Chatbots can handle simple customer service inquiries, freeing up human agents for complex cases.

Procedures (процедури) – Established methods or steps for completing a particular task or process.

Following security procedures is critical to protect sensitive data in IT systems.

Reasoning (міркування) – The process of forming conclusions, judgments, or inferences based on facts or logical principles.

Machine learning models rely on statistical reasoning to identify patterns in data.

Rote learning (механічне запам'ятовування) – A memorization technique based on repetition, often without understanding the underlying meaning or concepts.

Rote learning alone is insufficient in IT, as understanding concepts is crucial for problem-solving.

Inductive (індуктивний) – A reasoning approach that moves from specific observations to broader generalizations, often leading to hypotheses or theories.

Inductive methods in data science help generate hypotheses by observing patterns in data.

Premises (припущення) – Statements or propositions that are assumed to be true and are used as the basis for reasoning or drawing conclusions.

Developers operate on the premise that users want faster and more efficient software.

Inferences (висновки) – Logical conclusions drawn based on evidence and reasoning, often involving some degree of interpretation.

Inferences from user data can help tailor digital products to better meet customer needs.

Tentative models (попередні моделі) – Initial or provisional representations or explanations that are subject to further testing and refinement.

AI developers often work with tentative models that are iteratively improved over time.

Approach (підхід) – A method or strategy used to address a particular problem or task.

Our team's approach to cybersecurity includes both preventive measures and rapid response.

Prominent (помітний) – Highly noticeable or important; easily recognized or well-known.

Data privacy has become a prominent concern in the field of information technology.

Indistinguishable (непомітний, практично нічим не відрізняється) – Impossible to distinguish or tell apart due to extreme similarity.

Some AI-generated images are nearly indistinguishable from real photographs.

Gross (грубий) – Conspicuously incorrect or lacking in refinement; used to describe something very obvious or crude.

The software's design shows a gross misunderstanding of user needs.

Oversimplification (перегляд в бік спрощення) – The process of simplifying something to the point where it loses essential details or complexity, potentially leading to misunderstandings.

Oversimplification of security protocols can leave systems vulnerable to attacks.

Exercise 3. Read and translate the following text into Ukrainian paying attention to active vocabulary:

Artificial Intelligence

Artificial intelligence (AI), the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. The term is frequently applied to the project of developing systems endowed with the intellectual processes characteristic of humans, such as the ability to reason, discover meaning, generalize, or learn from past experience. Since their development in the 1940s, digital computers have been programmed to carry out very complex tasks − such as discovering proofs for mathematical theorems or playing chess − with great proficiency. Despite continuing advances in computer processing speed and memory capacity, there are as yet no programs that can match full human flexibility over wider domains or in tasks requiring much everyday knowledge. On the other hand, some programs have attained the performance levels of human experts and professionals in executing certain specific tasks, so that artificial intelligence in this limited sense is found in applications as **diverse** as medical diagnosis, computer search engines, voice or handwriting recognition, and **chatbots**.

What is intelligence in general? Psychologists generally characterize human intelligence not by just one trait but by the combination of many diverse abilities. Research in AI has focused chiefly on the following components of intelligence: learning, **reasoning**, problem solving, perception, and using language.

There are a number of different forms of learning as applied to artificial intelligence. The simplest is learning by trial and error. For example, a simple computer program for solving mate-in-one chess problems might try moves at random until mate is found. The program might then store the solution with the position so that, the next time the computer encountered the same position, it would recall the solution. This simple memorizing of individual items and **procedures** − known as **rote learning** − is relatively easy to implement on a computer. More challenging is the problem of implementing what is called generalization. Generalization involves applying past experience to analogous new situations. For example, a program that learns the past tense of regular English verbs by rote will not

be able to produce the past tense of a word such as jump unless the program was previously presented with jumped, whereas a program that is able to generalize can learn the “add -ed” rule for regular verbs ending in a consonant and so form the past tense of jump on the basis of experience with similar verbs.

To reason is to draw **inferences** appropriate to the situation. Inferences are classified as either deductive or **inductive**. An example of the former is, “Fred must be in either the museum or the café. He is not in the café; therefore, he is in the museum,” and of the latter is, “Previous accidents of this sort were caused by instrument failure. This accident is of the same sort; therefore, it was likely caused by instrument failure.” The most significant difference between these forms of reasoning is that in the deductive case, the truth of the **premises** guarantees the truth of the conclusion, whereas in the inductive case, the truth of the premises lends support to the conclusion without giving absolute assurance. Inductive reasoning is common in science, where data are collected and **tentative models** are developed to describe and predict future behavior − until the appearance of anomalous data forces the model to be revised. Deductive reasoning is common in mathematics and logic, where elaborate structures of irrefutable theorems are built up from a small set of basic axioms and rules.

There has been considerable success in programming computers to draw inferences. However, true reasoning involves more than just drawing inferences: it involves drawing inferences relevant to the solution of the particular problem. This is one of the hardest problems confronting AI.

Problem solving, particularly in artificial intelligence, may be characterized as a systematic search through a range of possible actions in order to reach some predefined goal or solution. Problem-solving methods divide into special purpose and general purpose. A special-purpose method is tailor-made for a particular problem and often exploits very specific features of the situation in which the problem is embedded. In contrast, a general-purpose method is applicable to a wide variety of problems. One general-purpose technique used in AI is means-end analysis−a step-by-step, or incremental, reduction of the difference between the current state and the final goal. The program selects actions from a list of means−in the case of a simple robot, this

might consist of PICKUP, PUTDOWN, MOVEFORWARD, MOVEBACK, MOVELEFT, and MOVERIGHT− until the goal is reached.

Many diverse problems have been solved by artificial intelligence programs. Some examples are finding the winning move (or sequence of moves) in a board game, devising mathematical proofs, and manipulating "virtual objects" in a computer-generated world.

In perception the environment is scanned by means of various sensory organs, real or artificial, and the scene is decomposed into separate objects in various spatial relationships. Analysis is complicated by the fact that an object may appear different depending on the angle from which it is viewed, the direction and intensity of illumination in the scene, and how much the object contrasts with the surrounding field. At present, artificial perception is sufficiently advanced to enable optical sensors to identify individuals and enable autonomous vehicles to drive at moderate speeds on the open road.

A language is a system of signs having meaning by convention. In this sense, language need not be confined to the spoken word. Traffic signs, for example, form a mini-language, it being a matter of convention that ⚠ means "hazard ahead" in some countries. It is distinctive of languages that linguistic units possess meaning by convention, and linguistic meaning is very different from what is called natural meaning, exemplified in statements such as "Those clouds mean rain" and "The fall in pressure means the valve is malfunctioning."

An important characteristic of full-fledged human languages − in contrast to birdcalls and traffic signs − is their productivity. A productive language can formulate an unlimited variety of sentences.

Large language models like ChatGPT can respond fluently in a human language to questions and statements. Although such models do not actually understand language as humans do but merely select words that are more probable than others, they have reached the point where their command of a language is **indistinguishable** from that of a normal human. What, then, is involved in genuine understanding, if even a computer that uses language like a native human speaker is

not acknowledged to understand? There is no universally agreed upon answer to this difficult question.

AI research follows two distinct, and to some extent competing, methods, the symbolic (or "top-down") **approach**, and the connectionist (or "bottom-up") approach. The top-down approach seeks to replicate intelligence by analyzing cognition independent of the biological structure of the brain, in terms of the processing of symbols − whence the symbolic label. The bottom-up approach, on the other hand, involves creating artificial neural networks in imitation of the brain's structure − whence the connectionist label.

During the 1950s and '60s the top-down and bottom-up approaches were pursued simultaneously, and both achieved noteworthy, if limited, results. During the 1970s, however, bottom-up AI was neglected, and it was not until the 1980s that this approach again became prominent. Nowadays both approaches are followed, and both are acknowledged as facing difficulties. Symbolic techniques work in simplified realms but typically break down when confronted with the real world; meanwhile, bottom-up researchers have been unable to replicate the nervous systems of even the simplest living things. Caenorhabditis elegans, a much-studied worm, has approximately 300 neurons whose pattern of interconnections is perfectly known. Yet connectionist models have failed to mimic even this worm. Evidently, the neurons of connectionist theory are **gross oversimplifications** of the real thing.

In the early 21st century faster processing power and larger datasets ("big data") brought artificial intelligence out of computer science departments and into the wider world. Moore's law, the observation that computing power doubled roughly every 18 months, continued to hold true. The stock responses of the early chatbot Eliza fit comfortably within 50 kilobytes; the language model at the heart of ChatGPT was trained on 45 terabytes of text. (Adapted from https://www.britannica.com/technology/artificial-intelligence/Is-artificial-general-intelligence-AGI-possible)

Exercise 4. Pick the correct answer from a list according to the text above.

1. Which of the following best describes artificial intelligence (AI)?

A. A system that can surpass human intelligence in all tasks

B. A technology limited to simple tasks like solving math problems

C. A digital system capable of performing tasks associated with intelligent beings

D. A technology exclusively used in gaming applications

2. What is the simplest form of learning in artificial intelligence, as mentioned in the text?

A. Deductive reasoning

B. Trial and error

C. Perception

D. Language processing

3. Which type of reasoning involves conclusions that are not guaranteed to be true, even if the premises are?

A. Deductive reasoning

B. Problem-solving reasoning

C. Inductive reasoning

D. Symbolic reasoning

Exercise 5. Translate into English.

Штучний інтелект (ШІ) – це симуляція людського інтелекту в машинах, які здатні думати, навчатися та приймати рішення подібно до людей. Системи ШІ можуть аналізувати дані, розпізнавати шаблони та виконувати завдання з мінімальним втручанням людини. Ця технологія стає революційною у багатьох сферах, включаючи охорону здоров'я, освіту, транспорт і розваги.

Види ШІ:

1. **Низький ШІ (Harrow AI):**

Призначений для виконання конкретних завдань. Прикладами є віртуальні помічники, як-от Siri, системи рекомендацій на Netflix або алгоритми виявлення шахрайства.

2. **Загальний ШІ (General AI):**

Теоретичний рівень, коли машини можуть виконувати будь-яке інтелектуальне завдання, яке виконує людина. Загальний ШІ поки залишається метою для дослідників.

3. **Суперінтелект (Super AI):**

Це концепція майбутнього, де машини перевершать інтелект людини. Це піднімає етичні та екзистенційні питання щодо майбутнього людства.

Основні компоненти ШІ:

1. **Машинне навчання (Machine Learning):** Підрозділ ШІ, де машини навчаються з даних без чіткого програмування. Такі системи покращуються з часом, аналізуючи більше інформації.

2. **Обробка природної мови (Natural Language Processing):** Дозволяє машинам розуміти людську мову та відповідати на неї. Це використовують у чат-ботах, голосових помічниках та автоматичному перекладі.

3. **Комп'ютерний зір (Computer Vision):** Дозволяє машинам інтерпретувати візуальну інформацію, як-от розпізнавання облич, об'єктів або почерку.

4. **Робототехніка (Robotics):**

Поєднує ШІ з механічними системами для створення інтелектуальних роботів, здатних виконувати завдання в різних галузях, таких як виробництво чи медицина.

Застосування ШІ:

- **Охорона здоров'я:** ШІ допомагає в діагностиці захворювань, прогнозуванні результатів лікування та розробці персоналізованих терапій.
- **Транспорт:** Автономні автомобілі використовують ШІ для навігації, розпізнавання перешкод і прийняття рішень на дорозі.
- **Фінанси:** ШІ виявляє шахрайські дії, прогнозує тенденції ринку та автоматизує торгівлю.
- **Обслуговування клієнтів:** Чат-боти й віртуальні помічники обробляють запити клієнтів 24/7.

- **Розваги:** ШІ персоналізує рекомендації фільмів, музики та ігор відповідно до уподобань користувачів.

Переваги ШІ:

- **Підвищення ефективності:** Автоматизація повторюваних завдань економить час і зменшує кількість помилок.
- **Покращення прийняття рішень:** ШІ аналізує великі обсяги даних, надаючи цінні інсайти для ухвалення рішень.
- **Інновації:** Сприяє прогресу в медицині, відновлюваній енергетиці та космічних дослідженнях.

Виклики та ризики:

1. **Етичні питання:** Використання ШІ у спостереженні, військових технологіях або упереджених рішеннях викликає дискусії.
2. **Заміна робочих місць:** Автоматизація може призвести до втрати роботи в певних галузях.
3. **Ризики конфіденційності:** ШІ потребує збору й аналізу особистих даних, що може порушити конфіденційність.
4. **Контроль і залежність:** З'являються побоювання щодо втрати контролю над складними системами ШІ.

Майбутнє ШІ:

Штучний інтелект продовжує розвиватися. Дослідники вивчають нові напрямки, такі як розпізнавання емоцій, передові робототехнічні системи та квантові обчислення. Попри великий потенціал, розвиток ШІ має бути врегульований етичними нормами, щоб він приносив користь людству.

Штучний інтелект уже змінює наше сьогодення й відкриває двері до нових можливостей. Головне – забезпечити, щоб інновації були відповідальними та справедливими для всіх.

Writing Activity

Exercise 6. Pick a question and write about it.

- Describe the two main approaches in AI research, the symbolic (top-down) and connectionist (bottom-up) approaches, including their strengths and limitations. Why do researchers continue to pursue both methods?
- How does artificial intelligence use reasoning to solve problems, and what are the key differences between deductive and inductive reasoning? Give examples of how each type is used in AI applications.
- Explain the challenges AI faces in achieving generalization and perception. How do these challenges limit AI's ability to replicate full human intelligence?

Speaking Activity

Exercise 7. Discuss the questions.

- In what ways do you think AI's ability to learn and generalize differs from human learning? Do you believe AI will ever achieve a level of understanding similar to human intelligence?
- How do you feel about the current applications of AI in fields like medical diagnosis and autonomous driving? What potential benefits and risks do you see as AI becomes more integrated into our daily lives?
- Between the symbolic (top-down) and connectionist (bottom-up) approaches in AI, which one do you think has a greater potential for replicating human-like intelligence? Why?

Texts for independent class and home translation

(Adapted from the book "The Pragmatic Programmer: From Journeyman to Master, Andrew Hunt, David Thomas")

Text 1. The Cat Ate My Source

Code The greatest of all weaknesses
is the fear of appearing weak.
J. B. Bossuet, Politics from Holy Writ, 1709

One of the cornerstones of the pragmatic philosophy is the idea of taking responsibility for yourself and your actions in terms of your career advancement, your project, and your day-to-day work. A Pragmatic Programmer takes charge of his or her own career, and isn't afraid to admit ignorance or error. It's not the most pleasant aspect of programming, to be sure, but it will happen − even on the best of projects. Despite thorough testing, good documentation, and solid automation, things go wrong. Deliveries are late. Unforeseen technical problems come up.

These things happen, and we try to deal with them as professionally as we can. This means being honest and direct. We can be proud of our abilities, but we must be honest about our shortcomings − our ignorance as well as our mistakes.

Text 2. Take Responsibility

Responsibility is something you actively agree to. You make a commitment to ensure that something is done right, but you don't necessarily have direct control over every aspect of it. In addition to doing your own personal best, you must analyze the situation for risks that are beyond your control. You have the right not to take on a responsibility for an impossible situation, or one in which the risks are too great. You'll have to make the call based on your own ethics and judgment.

When you do accept the responsibility for an outcome, you should expect to be held accountable for it. When you make a mistake (as we all do) or an error in judgment, admit it honestly and try to offer options.

Don't blame someone or something else, or make up an excuse. Don't blame all the problems on a vendor, a programming language, management, or your coworkers. Any and all of these may play a role, but it is up to you to provide solutions, not excuses.

If there was a risk that the vendor wouldn't come through for you, then you should have had a contingency plan. If the disk crashes − taking all of your source code with it − and you don't have a backup, it's your fault. Telling your boss «the cat ate my source code» just won't cut it.

Tip 3 Provide Options, Don't Make Lame Excuses

Before you approach anyone to tell them why something can't be done, is late, or is broken, stop and listen to yourself. Talk to the rubber duck on your monitor, or the cat. Does your excuse sound reasonable, or stupid? How's it going to sound to your boss?

Run through the conversation in your mind. What is the other person likely to say? Will they ask, «Have you tried this…» or «Didn't you consider that?» How will you respond? Before you go and tell them the bad news, is there anything else you can try? Sometimes, you just know what they are going to say, so save them the trouble.

Instead of excuses, provide options. Don't say it can't be done; explain what can be done to salvage the situation. Does code have to be thrown out? Educate them on the value of refactoring. Do you need to spend time prototyping to determine the best way to proceed ? Do you need to introduce better testing or automation to prevent it from happening again? Perhaps you need additional resources. Don't be afraid to ask, or to admit that you need help.

Try to flush out the lame excuses before voicing them aloud. If you must, tell your cat first. After all, if little Tiddles is going to take the blame….

Text 3. Software Entropy

While software development is immune from almost all physical laws, entropy hits us hard. Entropy is a term from physics that refers to the amount of «disorder» in a system. Unfortunately, the laws of thermodynamics guarantee that the entropy in the

universe tends toward a maximum. When disorder increases in software, programmers call it «software rot.»

There are many factors that can contribute to software rot. The most important one seems to be the psychology, or culture, at work on a project. Even if you are a team of one, your project's psychology can be a very delicate thing. Despite the best laid plans and the best people, a project can still experience ruin and decay during its lifetime. Yet there are other projects that, despite enormous difficulties and constant setbacks, successfully fight nature's tendency toward disorder and manage to come out pretty well.

What makes the difference?

In inner cities, some buildings are beautiful and clean, while others are rotting hulks. Why? Researchers in the field of crime and urban decay discovered a fascinating trigger mechanism, one that very quickly turns a clean, intact, inhabited building into a smashed and abandoned derelict.

Text 4. A broken window.

One broken window, left unrepaired for any substantial length of time, instills in the inhabitants of the building a sense of abandonment—a sense that the powers that be don't care about the building. So another window gets broken. People start littering. Graffiti appears. Serious structural damage begins. In a relatively short space of time, the building becomes damaged beyond the owner's desire to fix it, and the sense of abandonment becomes reality.

The «Broken Window Theory» has inspired police departments in New York and other major cities to crack down on the small stuff in order to keep out the big stuff. It works: keeping on top of broken windows, graffiti, and other small infractions has reduced the serious crime level.

Tip 4 Don't Live with Broken Windows

Don't leave «broken windows» (bad designs, wrong decisions, or poor code) unrepaired. Fix each one as soon as it is discovered. If there is insufficient time to fix it properly, then board it up. Perhaps you can comment out the offending code, or

display a «Not Implemented» message, or substitute dummy data instead. Take some action to prevent further damage and to show that you're on top of the situation.

We've seen clean, functional systems deteriorate pretty quickly once windows start breaking. There are other factors that can contribute to software rot, and we'll touch on some of them elsewhere, but neglect accelerates the rot faster than any other factor.

You may be thinking that no one has the time to go around cleaning up all the broken glass of a project. If you continue to think like that, then you'd better plan on getting a dumpster, or moving to another neighborhood. Don't let entropy win.

Text 5.Putting Out Fires

By contrast, there's the story of an obscenely rich acquaintance of Andy's. His house was immaculate, beautiful, loaded with priceless antiques, objets d'art, and so on. One day, a tapestry that was hanging a little too close to his living room fireplace caught on fire. The fire department rushed in to save the day—and his house. But before they dragged their big, dirty hoses into the house, they stopped—with the fire raging − to roll out a mat between the front door and the source of the fire.

They didn't want to mess up the carpet.

A pretty extreme case, to be sure, but that's the way it must be with software. One broken window−a badly designed piece of code, a poor management decision that the team must live with for the duration of the project— is all it takes to start the decline. If you find yourself working on a project with quite a few broken windows, it's all too easy to slip into the mindset of «All the rest of this code is crap, I'll just follow suit.» It doesn't matter if the project has been fine up to this point. In the original experiment leading to the «Broken Window Theory,» an abandoned car sat for a week untouched. But once a single window was broken, the car was stripped and turned upside down within hours.

By the same token, if you find yourself on a team and a project where the code is pristinely beautiful—cleanly written, well designed, and elegant—you will likely take extra special care not to mess it up, just like the firefighters. Even if there's a fire

raging (deadline, release date, trade show demo, etc.), you don't want to be the first one to make a mess.

Text 6.Stone Soup and Boiled Frogs

The three soldiers returning home from war were hungry. When they saw the village ahead their spirits lifted—they were sure the villagers would give them a meal. But when they got there, they found the doors locked and the windows closed. After many years of war, the villagers were short of food, and hoarded what they had.

Undeterred, the soldiers boiled a pot of water and carefully placed three stones into it. The amazed villagers came out to watch.

«This is stone soup,» the soldiers explained. «Is that all you put in it?» asked the villagers. «Absolutely−although some say it tastes even better with a few carrots….» A villager ran off, returning in no time with a basket of carrots from his hoard.

A couple of minutes later, the villagers again asked «Is that it?»

«Well,» said the soldiers, «a couple of potatoes give it body.» Off ran another villager.

Over the next hour, the soldiers listed more ingredients that would enhance the soup: beef, leeks, salt, and herbs. Each time a different villager would run off to raid their personal stores.

Eventually they had produced a large pot of steaming soup. The soldiers removed the stones, and they sat down with the entire village to enjoy the first square meal any of them had eaten in months.

There are a couple of morals in the stone soup story. The villagers are tricked by the soldiers, who use the villagers' curiosity to get food from them. But more importantly, the soldiers act as a catalyst, bringing the village together so they can jointly produce something that they couldn't have done by themselves—a synergistic result. Eventually everyone wins.

Every now and then, you might want to emulate the soldiers.

You may be in a situation where you know exactly what needs doing and how to do it. The entire system just appears before your eyes—you know it's right. But ask

permission to tackle the whole thing and you'll be met with delays and blank stares. People will form committees, budgets will need approval, and things will get complicated. Everyone will guard their own resources. Sometimes this is called «start-up fatigue.»

It's time to bring out the stones. Work out what you can reasonably ask for. Develop it well. Once you've got it, show people, and let them marvel. Then say «of course, it would be better if we added….» Pretend it's not important. Sit back and wait for them to start asking you to add the functionality you originally wanted. People find it easier to join an ongoing success. Show them a glimpse of the future and you'll get them to rally around.

While doing this, you may be comforted by the line attributed to Rear Admiral Dr. Grace Hopper: «It's easier to ask forgiveness than it is to get permission.»

Tip 5 Be a Catalyst for Change

Text 7.The Villagers' Side

On the other hand, the stone soup story is also about gentle and gradual deception. It's about focusing too tightly. The villagers think about the stones and forget about the rest of the world. We all fall for it, every day. Things just creep up on us.

We've all seen the symptoms. Projects slowly and inexorably get totally out of hand. Most software disasters start out too small to notice, and most project overruns happen a day at a time. Systems drift from their specifications feature by feature, while patch after patch gets added to a piece of code until there's nothing of the original left. It's often the accumulation of small things that breaks morale and teams.

Tip 6 Remember the Big Picture

We've never tried this—honest. But they say that if you take a frog and drop it into boiling water, it will jump straight back out again. However, if you place the frog in a pan of cold water, then gradually heat it, the frog won't notice the slow increase in temperature and will stay put until cooked.

Note that the frog's problem is different from the broken windows issue discussed in before. In the Broken Window Theory, people lose the will to fight entropy because they perceive that no one else cares. The frog just doesn't notice the change.

Don't be like the frog. Keep an eye on the big picture. Constantly review what's happening around you, not just what you personally are doing.

Text 8. Good-Enough Software

Striving to better, oft we mar what's well.

King Lear 1.4

There's an old(ish) joke about a U.S. company that places an order for 100,000 integrated circuits with a Japanese manufacturer. Part of the specification was the defect rate: one chip in 10,000. A few weeks later the order arrived: one large box containing thousands of ICs, and a small one containing just ten. Attached to the small box was a label that read: «These are the faulty ones.»

If only we really had this kind of control over quality. But the real world just won't let us produce much that's truly perfect, particularly not bug-free software. Time, technology, and temperament all conspire against us.

However, this doesn't have to be frustrating. As Ed Yourdon described in an article in IEEE Software, you can discipline yourself to write software that's good enough—good enough for your users, for future maintainers, for your own peace of mind. You'll find that you are more productive and your users are happier. And you may well find that your programs are actually better for their shorter incubation.

Before we go any further, we need to qualify what we're about to say. The phrase «good enough» does not imply sloppy or poorly produced code. All systems must meet their users' requirements to be successful. We are simply advocating that users be given an opportunity to participate in the process of deciding when what you've produced is good enough.

Text 9. Involve Your Users in the Trade-Off

Normally you're writing software for other people. Often you'll remember to get requirements from them. But how often do you ask them how good they want their software to be? Sometimes there'll be no choice. If you're working on pacemakers, the space shuttle, or a low-level library that will be widely disseminated, the requirements will be more stringent and your options more limited. However, if you're working on a brand new product, you'll have different constraints. The marketing people will have promises to keep, the eventual end users may have made plans based on a delivery schedule, and your company will certainly have cash-flow constraints. It would be unprofessional to ignore these users' requirements simply to add new features to the program, or to polish up the code just one more time. We're not advocating panic: it is equally unprofessional to promise impossible time scales and to cut basic engineering corners to meet a deadline. That was supposed to be a joke!

The scope and quality of the system you produce should be specified as part of that system's requirements.

Tip 7 Make Quality a Requirements Issue

Often you'll be in situations where trade-offs are involved. Surprisingly, many users would rather use software with some rough edges today than wait a year for the multimedia version. Many IT departments with tight budgets would agree. Great software today is often preferable to perfect software tomorrow. If you give your users something to play with early, their feedback will often lead you to a better eventual solution (see Tracer Bullets).

Text 10. Know When to Stop

In some ways, programming is like painting. You start with a blank canvas and certain basic raw materials. You use a combination of science, art, and craft to determine what to do with them. You sketch out an overall shape, paint the underlying environment, then fill in the details. You constantly step back with a critical eye to view what you've done. Every now and then you'll throw a canvas away and start again.

But artists will tell you that all the hard work is ruined if you don't know when to stop. If you add layer upon layer, detail over detail, the painting becomes lost in the paint.

Don't spoil a perfectly good program by overembellishment and over-refinement. Move on, and let your code stand in its own right for a while. It may not be perfect. Don't worry: it could never be perfect.

Запитання для самоконтролю

1. Choose the correct variant

1. What does System Software consist of?

A. The operating System

B. Graphics software

C. Private software

D. Drivers

2. What is the correct definition of Application Software?

A. The everyday programs you use

B. Programs that are rarely used

C. Programs made specifically for a client

D. Things included in your computer

3. Google has a free open source program

A. True

B. False

4. Which of the following AREN'T applications

A. Drivers

B. USB

C. Games

5. Software only works when there is Hardware

Discuss

A. True

B. False

6. Which of the following statements about formatting is true?

A. It is used to erase all existing data on the disk and so a new disk does not need to be formatted.

B. Formatting initialises the filing system and so prepares a disk for use. All existing data on the disk is erased.

C. Formatting erases everything on the disk except the operating system.

D. All the above are true.

7. Which of the following utilities can make internet downloading safer?

A. Scandisk

B. Antivirus

C. Defragmentation

D. Winzip

8. Why is a File-Compression utility ideal for use when files are being transferred over a network connection?

A. Zipped files are smaller and so sending them is faster

B. Zipped files are always virus-free

C. It is impossible to send unzipped files over a network connection

D. None of the above are true

9. Which of these utilities can help us get rid of malicious software?

A. Scan disk, antivirus

B. Anti-spyware software, antivirus

C. Scan disk, antivirus, anti-spyware software

D. Antivirus, trojan, anti-spyware software

10. Which of the following statements is/are correct?

A. A virus copies itself automatically between and within computers by attaching itself to another file.

B. A worm spreads very fast over a network and often slows down a system.

C. Trojans are not viruses because they do not copy themselves automatically

D. All the above are true

2. Choose the correct answer from a list according to the text above

1. 'PC' or 'personal computer' is:

A. A portable computer shaped in the form of a notebook and capable of advanced handwriting recognition via use of a stylus or on-screen keyboard

B. A computer designed for use by one person at a time

C. A high quality computer, typically with lots of RAM, plenty of CPU power, and a high quality video card

2. 'Embedded system' is:

A. A computer using a relatively slow and specialized processor and ROM chip, normally used to control a particular device such as a washing machine or an MP3 player

B. The evolution of devices towards common functionality

C. The evolution of devices towards common functionality

3. 'Workstation' is:

A. Hardware or software designed to connect a network device with a printer

B. A computer designed to perform intensive tasks such as weather prediction, big data calculations, or space research

C. A high quality computer, typically with lots of RAM, plenty of CPU power, and a high quality video card

4. 'Print server' is:

A. A computer using a relatively slow and specialized processor and ROM chip, normally used to control a particular device such as a washing machine or an MP3 player

B. Hardware or software designed to connect a network device with a printer.

C. A computer designed for use by one person at a time

5. 'Legacy system' is:

A. A high quality computer, typically with lots of RAM, plenty of CPU power, and a high quality video card

B. A type of computer intended primarily for central distribution of data to other computers on the same network

C. An older device or application that continues to be used because of the high cost of replacing it

6. 'Server' is:

A. A type of computer intended primarily for central distribution of data to other computers on the same network

B. A hand-held multimedia computer optimized for communication and featuring a touch screen and internet connection

C. The size, configuration, or physical arrangement of a computing device

7. 'PDA' or 'personal digital assistant' is:

A. A legacy handheld computer often running Palm OS or Windows CE and used as a contact organizer, game machine, work tool, or access controller

B. A specialized computer which connects two networks

C. Hardware or software designed to connect a network device with a printer

8. 'Supercomputer' is:

A. Computers designed to be worn on the wrist, head, or other places on the body often to improve mobility

B. Hardware or software designed to connect a network device with a printer

C. A computer designed to perform intensive tasks such as weather prediction, big data calculations, or space research

9. 'Desktop computer' is:

A. Hardware or software designed to connect a network device with a printer

B. A specialized computer which connects two networks

C. A personal computer typically in the shape of a tower or box with a connected keyboard, mouse, and monitor

10. 'Tablet' is:

A. A high quality computer, typically with lots of RAM, plenty of CPU power, and a high quality video card

B. A portable computer shaped in the form of a notebook and capable of advanced handwriting recognition via use of a stylus or on-screen keyboard

C. A computer designed for use by one person at a time

11. 'Coprocessor' it is:

A. A wired or increasingly wireless PCI or USB device that connects a computer to a network

B. A hardware component which allows a computer to play and record audio

C. Any computer processor or portion of the CPU which assists the main processor with a highly specialized task

12. 'Power supply' it is:

A. An electronic device constructed from microscopic transistors on a single integrated circuit

B. A separate unit or part of a circuit that supplies the correct amount of electrical current to a computer system

C. A hardware component which allows a computer to play and record audio

13. 'Microprocessor' it is:

A. A newer bus type used mainly for graphic cards and running at speeds up to 16,000 MB/s

B. A component of a computer which is designed to convert a binary image stored in memory to a display medium

C. An electronic device constructed from microscopic transistors on a single integrated circuit

14. 'PCI' or 'Peripheral Component Interconnect' it is:

A. A specialized co-processor designed to handle graphical calculations such as 3D modeling and games
B. A standard 32-bit bus running at 132 MB/s
C. Normally a ROM program that controls the base functionality of the computer such as video, hard drives, optical drives, and keyboard

15. 'SATA' or 'Serial ATA' it is:

A. A widely used bus for connecting hard disk drives and similar devices to the motherboard
B. A clear set of technical or quantitative requirements
C. A specialized co-processor designed to handle graphical calculations such as 3D modeling and games

16. 'Spec' or 'specification' it is:

A. A clear set of technical or quantitative requirements
B. A newer bus type used mainly for graphic cards and running at speeds up to 16,000 MB/s
C. A wired or increasingly wireless PCI or USB device that connects a computer to a network

17. 'GPU' or 'Graphics Processing Unit' it is:

A. A clear set of technical or quantitative requirements
B. A separate unit or part of a circuit that supplies the correct amount of electrical current to a computer system
C. A specialized co-processor designed to handle graphical calculations such as 3D modeling and games

18. 'Sound card' it is:

A. A newer bus type used mainly for graphic cards and running at speeds up to 16,000 MB/s
B. A component of a computer which is designed to convert a binary image stored in memory to a display medium
C. A hardware component which allows a computer to play and record audio

19. 'Graphics card' it is:

A. A component of a computer which is designed to convert a binary image stored in memory to a display medium

B. Any computer processor or portion of the CPU which assists the main processor with a highly specialized task

C. A hardware component which allows a computer to play and record audio

20. 'USB' or 'Universal Serial Bus' it is:

A. A high-level computer language developed in the 1960's used in business, finance, and administrative systems for companies and governments

B. A fast and convenient computer technology allowing the connection common peripherals such as digital cameras, scanners, external hard drives, etc

C. Mountain View, California company founded by Larry Page and Sergei Brin in 1998; known for their search technology, Android mobile OS, web-based productivity tools, and cloud hosting services

3. Select the best definition of the word

1. error

a) something a computer program is «supposed» to do; these are often reasons to use a particular program or upgrade to a more recent version

b) an incorrect action attributable to poor judgment, ignorance, or inattention

c) a rule or law which limits or controls something

2. compatible

a) capable of being used without modification

b) an electronic, digital device that stores and processes information

c) the largest known public network in the world, connecting millions of computers around the world

3. bug

a) an incorrect action attributable to poor judgment, ignorance, or inattention

b) a computer failure due to faulty hardware or a serious software bug

c) an error in a computer program

4. restriction

a) a rule or law which limits or controls something
b) software in which the license stipulates that the user cannot see, edit, or manipulate the source code of a software program
c) an application normally consisting of a source code editor, a compiler and/or interpreter, build-automation tools, and a debugger

5. open source

a) a measure of how easy or efficient a program is to use
b) software in which the license stipulates that the user cannot see, edit, or manipulate the source code of a software program
c) a program in which the code is distributed allowing programmers to alter and change the original software as much as they like

6. crash

a) a computer failure due to faulty hardware or a serious software bug
b) an application normally consisting of a source code editor, a compiler and/or interpreter, build-automation tools, and a debugger
c) software in which the license stipulates that the user cannot see, edit, or manipulate the source code of a software program

7. abort

a) to start a program on a computer
b) to end a program or a process before its completion
c) an electronic, digital device that stores and processes information

8. end user

a) a person who uses a product or service on a computer
b) a person who writes or modifies computer programs or applications
c) an electronic, digital device that stores and processes information

9. programmer

a) a person who writes or modifies computer programs or applications
b) a person who uses a product or service on a computer
c) an electronic, digital device that stores and processes information

10. 'IDE' or 'integrated development environment'

a) something a computer program is «supposed» to do; these are often reasons to use a particular program or upgrade to a more recent version
b) software in which the license stipulates that the user cannot see, edit, or manipulate the source code of a software program
c) an application normally consisting of a source code editor, a compiler and/or interpreter, build-automation tools, and a debugger

11. usability

a) a computer failure due to faulty hardware or a serious software bug
b) a rule or law which limits or controls something
c) a measure of how easy or efficient a program is to use

12. proprietary

a) an error in a computer program
b) something a computer program is «supposed» to do; these are often reasons to use a particular program or upgrade to a more recent version
c) privately developed and owned technology

Список термінів-метафор в IT

Домени	Термін	Пояснення та переклад
Релігійні	**God object**	В українській мові можна перекласти як «об'єкт-бог». Це термін із програмування, який означає об’єкт, що має занадто багато відповідальностей або занадто великий вплив на систему. Такий об’єкт порушує принципи гарного дизайну, зокрема принцип єдиної відповідальності, що робить код важким для розуміння, тестування та підтримки
	Exodus	Процес переміщення від застарілої технології чи старої платформи до нової, що може бути складним, але необхідним кроком для розвитку. У контексті програмування ***Exodus*** (від лат. *exodūs* – «вихід») зазвичай перекладається як **«Вихід»** і використовується для позначення процесу або етапу, коли система, програма або команда переходить від однієї технології чи платформи до іншої **Значення в програмуванні:** **1. Перехід від застарілих технологій до нових:** «Exodus» може позначати перехід від старої технології або платформи до нової. Це може бути, наприклад, міграція з застарілого програмного забезпечення або бази даних на сучасне, більш ефективне рішення. **2. Міграція або виведення даних:** У контексті баз даних або програмного забезпечення це може означати процес виведення даних з однієї системи в іншу, щоб адаптуватися до нових вимог або змінити платформу. Термін «Exodus» має біблійне коріння і оригінально означає **«вихід»** або **«мігрантський шлях»**. У Біблії «Exodus» позначає вихід єврейського народу з Єгипту. У програмуванні цей термін застосовується метафорично, описуючи великі зміни або важливі етапи, коли система, організація чи команда «виходить» з одного стану в інший. Загалом, у програмуванні **Exodus** — це метафора для великого, часто складного переходу або змін у проекті чи технології
	Айті-євангеліст (або «IT-проповідник / пропагандист»)	людина, яка займається пропагандою (нерідко професійно) в сфері IT
Кулінарні	**Spaghetti code**	Погано структурований код, де немає чіткої логіки, і він заплутаний, наче макарони, що ускладнює його розуміння та супровід
	Lasagna code	Багатошаровий код, де логіка розділена на багато рівнів або «шарів», що робить його важким для розуміння, але дещо краще організованим, ніж

		«спагеті-код»
	Swiss cheese model	Модель безпеки чи ризиків, яка показує, як невеликі дірки в різних шарах системи можуть спричинити серйозні проблеми, коли ці дірки виявляються в одному місці
	Cookie	Невеликий файл, збережений на комп'ютері користувача для зберігання інформації, такої як налаштування користувача чи сесії
	Honeypot	Спеціально створене вразливе середовище для того, щоб привернути увагу хакерів і відвернути їх від справжніх систем
	Breadcrumbs	Сліди навігації, які допомагають користувачеві зрозуміти, де він знаходиться в ієрархії сторінок чи меню програми
	Hot potato	Проблема або задача, якою ніхто не хоче займатися і яку «перекидають» між членами команди
	Onion architecture	Структура програмного забезпечення, де код розподілений на шари, подібно до шарів цибулі, щоб зробити систему гнучкішою та легшою для супроводу
	Dog fooding	Процес, коли компанія сама використовує свій продукт, щоб краще зрозуміти його сильні та слабкі сторони
	Salt	Випадкові дані, додані до паролів перед хешуванням для підвищення їхньої безпеки
	Pepper	Додаткове випадкове значення, яке додається до хешів паролів, зазвичай зберігається окремо від самого хешу
Предмети побуту	**Garbage in, garbage out (GIGO)**	Принцип, згідно з яким неправильно введені або погані дані призводять до помилок або непридатного результату
	Garbage collection	Автоматичний процес видалення непотрібних об'єктів з пам'яті, щоб звільнити ресурси для інших операцій
	Fork	Копія проєкту, яку створюють для внесення змін, часто для розвитку нових функцій або для експериментів, не змінюючи основну версію
	Forking the code	Створення копії існуючого коду для початку розробки нового напрямку або нової версії програмного забезпечення
	Pod **Переклад:** Под	Найменший і найпростіший об'єкт у Kubernetes, який представляє набір контейнерів, що працюють на вашому кластері
	Container	Метод пакування і безпечного запуску застосунків у середовищі віртуалізації додатків
	Black-box testing	Метод тестування програмного забезпечення, при якому перевіряється робота програми без знання її внутрішньої побудови та схеми роботи. Іншими словами, не маючи доступу до коду програми

	Keys	Унікальний ідентифікатор, який дозволяє іншим програмам автентифікувати користувача, розробника або програму при зверненні до API, щоб підтвердити їхню автентичність
Анімалістичні	**Monkey patching**	Процес внесення змін або додавання функціоналу до вже існуючого коду або бібліотеки, іноді нестандартними методами
	Rubber duck debugging	Метод налагодження, коли програміст пояснює свій код або проблему простому об'єкту (наприклад, гумовій качечці), щоб знайти помилку або зрозуміти логіку коду
	Code monkey Переклад: Мавпа, що пише код	Жартівливий термін для програміста, який просто пише код, не беручи участі в плануванні чи проєктуванні, а лише виконує завдання, як «робот»
	Duck typing	Принцип програмування, де об'єкт визначається не через його тип, а через поведінку (якщо виглядає і працює, як качка, то це качка)
	One-trick pony	Поні з одним трюком» — це принизливий термін, який використовується для опису людини, яка має лише одну обмежену здатність або навичку та не може ефективно виконувати будь-які інші завдання
	Yak shaving	Ситуація, коли для досягнення кінцевої мети потрібно виконати серію непрямих і часто нудних задач. Термін підкреслює складність і неочевидність такого процесу
	Gorilla arm	Ситуація, коли інтерфейс або технологія змушують користувача чи розробника постійно виконувати фізично важку або неприємну роботу, наприклад, тримати руку в піднятій позиції
	Canary release	Метод тестування нової версії програмного забезпечення шляхом поступового її випуску для обмеженої кількості користувачів, щоб перевірити, чи не з'являються критичні помилки
Транспорт	**Bus factor / truck factor**	У сфері розробки програмного забезпечення bus factor (чи truck factor) проєкту — міра зосередження інформації серед окремих членів проєкту. Bus factor показує кількість розробників команди програмістів, після втрати яких проєкт не може бути продовжений далі
	Parking lot	Метод залишення ідей або задач на потім, зазвичай до завершення більш важливих справ, ніби вони «паркуються» до кращого часу
	Helicopter view	Високий рівень огляду проблеми чи проєкт, коли ви маєте загальне уявлення про всі аспекти, не занурюючись в деталі
	Pipeline	Серія етапів обробки даних, де вихід одного етапу є вхідними даними для наступного
Антропоморфні	**Singleton**	Шаблон проєктування, коли існує лише один екземпляр класу, який відповідає за певну

		функціональність
	Cowboy coding	Стиль програмування без належного планування, документації або дотримання правил, коли програміст діє як «ковбой», роблячи все на свій розсуд
	Greedy algorithm	Алгоритм, який на кожному етапі вибирає оптимальне (жадібне) рішення, яке здається найкращим на той момент, хоч і не завжди веде до глобально оптимального результату
	Handshake	Процес встановлення зв’язку або з’єднання між двома системами чи комп'ютерами, подібно до вітання перед початком спілкування.
	Head (branch)	Іменоване посилання на коміт, що знаходиться на кінці гілки
	Master (branch)	Гілка за замовчуванням для розробки в системі контролю версій GIT
	Host	Будь-який апаратний пристрій, що дозволяє доступ до мережі за допомогою інтерфейсу користувача, спеціалізованого програмного забезпечення, мережевої адреси чи протоколів
	Listener	Об'єкт, який очікує і реагує на події у програмуванні, орієнтованому на події.
	Multi-tenant	Тип архітектури, що використовується у хмарних обчисленнях, яка дозволяє запускати кілька окремих екземплярів програмного забезпечення з одного фізичного сервера
	Parent class	У ООП, батьківський клас – клас, від якого успадковуються атрибути та функції дочірніми класами
	Child class	У ООП, дочірній – клас, що успадковує всі атрибути й функції батьківського класу
Танатологічні	**Blue screen of death (BSOD)**	Критичний системний збій в операційній системі Windows, коли комп'ютер показує синій екран з повідомленням про помилку і вимагає перезавантаження
	Dead code	Частини коду, які більше не виконуються або не мають впливу на програму, але все ще присутні у вихідному коді
	Kill switch	Механізм, що дозволяє швидко відключити програму чи систему в разі серйозних помилок або загроз
	Kill chain	Покроковий процес або стратегія, що використовується для досягнення цілі, наприклад, в атаці на систему, де кожен етап є важливим для успіху
	Killer feature	Унікальна або дуже потужна функція, яка робить продукт або програмне забезпечення особливо привабливим для користувачів
	EOL (End of Life)	У програмних продуктах це означає, що програма

		досягла кінця свого терміну служби. Це може означати, що доступна нова версія, яка замінює наявний продукт. Або це може означати припинення підтримки (EOS), коли виробник більше не надає оновлень
Архітектурні	**Broken window**	Погана практика або незначна помилка у коді, яка залишена без виправлення і може призвести до зниження якості всього проєкту
	Spaghetti architecture	Погано спроєктована архітектура програмного забезпечення, у якій модулі чи компоненти переплетені настільки, що це ускладнює зміни та обслуговування
	Laying the groundwork	Початкова стадія будівельних робіт, під час якої зносяться наявні будівлі, викорчовуються дерева і т.д. та готується поверхня землі
	Backdoor	Секретний або несанкціонований спосіб доступу до системи, зазвичай для обходу стандартних процедур безпеки
	Walled garden	Закрита система або середовище, яке обмежує взаємодію з іншими системами або додатками, щоб зберегти контроль
	Sandbox	Ізольоване тестове середовище, де код може виконуватися без впливу на основну систему
	Gateway	Інструмент управління API, що слугує посередником між клієнтом і набором різних бекенд-сервісів
	Monolithic architecture (**monolith**)	Єдина, велика обчислювальна система з однією базою коду, яка об'єднує всі бізнес-функції разом
	Bridge	Патерн проєктування, що розділяє абстракцію і її реалізацію, дозволяючи їм змінюватися незалежно
Природничі	**Greenfield project**	Проєкт, що починається з нуля, без необхідності підтримки або врахування попередніх версій чи обмежень
	Brownfield project	Проєкт, який вимагає роботи з наявними системами або кодом, що потребує оновлення або переписування
	Low-hanging fruit	Легко здійснені завдання або проблеми, які можна вирішити без особливих зусиль, часто починаючи з них для швидкого результату
	Waterfall model	Модель розробки програмного забезпечення, де процес проєктування розділений на фази, і кожна фаза повинна завершитись перед початком наступної

English-Ukrainian Vocabulary

Altcoin – альткоін

Approach − підхід

array – набір

artificial intelligence – штучний інтелект

backwards-compatible – сумісний з попередніми версіями

barebone – напівзібранний комп`ютер

BIOS – базова система вводу\виводу

blockchain – розподілена база даних

blurred – розпливчатий

bulb – колба

callback – зворотнє з`єднання

capacitor - конденсатор

chipset – набір схем

cipher - шифр

circuit - ланцюг

clumsy – незручний

commandment - наказ

compiled - складений

compiler – компилятор

computer case – корпус комп`ютера

conductor – провідник

consensus process – процедура прийняття рішень

consent – згода

consequences - наслідки

consume – споживати

convergence – конвергенція

coprocessor - копроцесор

copyright infrigment – порушення авторських прав

current – потік

Daps (decentralized applications) – протокол DAP

descendent – нащадок

desktop computer – настільний комп`ютер

diverse – різноманітний

electrical circuit – електричний ланцюг

elegant code – елегантний код

embedded system – вбудована система

EMF (electromotive force) – електрорушійна сила

encounter – зустріти

fiat - постанова

filament – волокно

fintech – фінансово-технологічний сектор

firewall – система обмеення доступу

flash bulb – спалахова лампа

for loop – цикл з лічильником

form factor– конструктивний параметр

fully featured – повнофункціональний

fungible - взаємозамінний

geek – комп`ютерний експерт

global scope – глобальна галузь бачення

goto statement – оператор goto

GPU (Graphics Processing Unit) – ядро процесора

graphics card – графічний адаптер

gross – огидний

havoc – хаос

IC (integrated circuit) – інтегральна схема

indistinguishable – непомітний

nferences –висновки

internet privacy – конфідеційність в інтернеті

interpreter – інтерпретатор

laptop – ноутбук

LED (light-emitting diode) – світловипромінювальний діод

legacy system – традиційна система

light bulb - електролампа

mainframe - комп`ютер звичайного розміру

microprocessor – мікропроцесор

mining – майнінг

money laundering – легалізація незаконних доходів

motherboard – материнська плата

multi-paradigm language – багатопарадигмальна мова програмування

NIC (network interface card) – мережева карта

number-crunching task – вирішення числових задач великого об`єму

object-oriented language – об`єктно-орієнтована мов

operand – компонент операції

oversimplification – спрощення

path - маршрут

payroll – платіжна відомість

PC – персональний комп`ютер

PCB – блок зв`язку програм

PCI (Peripheral Component Interconnect)–взаємодія переферійних компонентів

PCI Express (Peripheral Component Interconnect Express) – послідовний інтерфейс

PDA – персональний цифровий асистент

phishing – злочинна діяльність інтернет-шахраїв

portability – портативність

power supply – джерело електроживлення

premises − припущення

print server – сервер печатки

private key – секретний ключ

privacy policy – політика конфідеційності

procedural language – процедурно-орієнтована мова

prominent − помітний

pump and dump – схема «накачування і скидання»

reflect on – замислюватися над

Regex (regular expression) – регулярний вислів

resistance - опір

reusable software components – програмні компоненти для багаторазового використання

Rote learning − механічне запам'ятовування

router - роутер

run on – працює на

SATA (Serial ATA) – послідовний ATA

serial port – послідовний порт

server - сервер

sloppy – неохайний

smartphone - смартфон

snoop around – шпигувати

social engineering – психологічна атака

solar cell – сонячна батарея

soldering gun – паяльний пістолет

soldering iron – паяльник

sound card – звуковий адаптер

spec (specification) – специфікація

string – послідовність

subroutine – стандартна підпрограма

supercomputer - суперкомп`ютер

syntax – синтаксична структура

tablet – планшет

tedious – нудний

The dark web – тіньова мережа

USB (Universal Serial Bus) – універсальна послідовна шина

validation – перевірка

voltage - напруга

voltaic cells – вольтові ланцюги

waste power – витрата електроенергії

wearables – натільні пристрої

wearable computer – мініатюрний комп`ютер

while loop – цикл з перевіркою умов

wire - дріт

write once, run anywhere – що написано один раз, виконується будь-де

Ukrainian-English Vocabulary

А
альткоін – Altcoin
Б
багатопарадигмальна мова програмування – Multi-paradigm language
базова система вводу/виводу – BIOS
блок зв’язку програм – PCB
Blockchain – розподілена база даних
булга – Bulb
розпливчатий – Blurred
багатофункціональний – Fully featured
блок живлення – Power supply
блок прийняття рішень – Consensus process
В
велика кількість обчислень – Number-crunching task
взаємозамінний – Fungible
вбудована система – Embedded system
видалення – Erasure
високий рівень доступу – Global scope
використання – Usage
витрата електроенергії – Waste power
вирішення задач – Solution solving
вольтові ланцюги – Voltaic cells
Г
графічний адаптер – Graphics card
глобальна галузь бачення – Global scope
глибокі висновки – Inferences
гнучкий – Flexible
Д
довідка – Reference
дріт – Wire
друкований сервер – Print server
джерело електроживлення – Power supply
Е
елегантний код – Elegant code
електричний ланцюг – Electrical circuit
електролампа – Light bulb
електрорушійна сила – EMF (Electromotive Force)
З
загальні положення – General terms
звуковий адаптер – Sound card
згода – Consent
зворотне з’єднання – Callback
І

інтегральна схема – IC (Integrated Circuit)
інтернет-конфіденційність – Internet privacy
інтерпретатор – Interpreter
інтелект – Intelligence
К
конденсатор – Capacitor
конструктивний параметр – Form factor
компілятор – Compiler
комп’ютер звичайного розміру – Mainframe
корпус комп’ютера – Computer case
Л
легалізація незаконних доходів – Money laundering
легкість використання – Usability
ланцюг – Circuit
М
майнінг – Mining
механічне запам’ятовування – Rote learning
мережева карта – NIC (Network Interface Card)
мікропроцесор – Microprocessor
мініатюрний комп’ютер – Wearable computer
материнська плата – Motherboard
Н
набір – Array
набір схем – Chipset
накачування й скидання (схема) – Pump and dump
настільний комп’ютер – Desktop computer
натільні пристрої – Wearables
незручний – Clumsy
непомітний – Indistinguishable
наслідки – Consequences
О
об’єктно-орієнтована мова – Object-oriented language
опір – Resistance
оператор goto – Goto statement
операнд – Operand
огидний – Gross
П
пам’ять – Memory
пароль – Password
платіжна відомість – Payroll
планшет – Tablet
портативність – Portability
припущення – Premises
процедура прийняття рішень – Consensus process

програмні компоненти для багаторазового використання – Reusable software components
приватний ключ – Private key
психологічна атака – Social engineering
Р
регулярний вираз – Regex (Regular Expression)
роутер – Router
розподілена база даних – Blockchain
розпливчатий – Blurred
робота на – Run on
С
сервер – Server
світловипромінювальний діод – LED (Light-Emitting Diode)
специфікація – Spec (Specification)
сонячна батарея – Solar cell
стандартна підпрограма – Subroutine
суперкомп’ютер – Supercomputer
схема – Scheme
схема перекачування і скидання – Pump and dump
Т
тіньова мережа – The Dark Web
традиційна система – Legacy system
точка доступу – Access point
У
універсальна послідовна шина – USB (Universal Serial Bus)
упорядкування – Order
Ф
фінансово-технологічний сектор – Fintech
файл – File
форми факторів – Form factor
Х
хаос – Havoc
Ц
цикл із перевіркою умов – While loop
цикл із лічильником – For loop
Ш
шифр – Cipher
Я
ядро процесора – GPU (Graphics Processing Unit)

Список рекомендованої літератури

1. Англо-український словник у 2 т. / М. І. Балла (скл.). – Київ : Освіта, 1996. – Т. 1 – 152 с., т. 2 – 712 с.

2. Бадан А. А. Практичний курс з перекладу термінології комп'ютерних систем з англійської на українську мову : навч. посібник / А. А. Бадан, С. О. Царьова ; Нац. техн. ун-т «Харків. політехн. ін-т». – Харків : НТУ «ХПІ», 2010. – 83 с.

3. Бадан А. А., Царьова С. О. «Англійська мова» для студентів економічних та комп'ютерних спеціальностей : навч.-метод. посіб. / А. А. Бадан, С. О. Царьова. – Вінниця : Нова книга, 2004. – 272 с.

4. Балабан Т. Англо-український словник-довідник інженерії довкілля / Т. Балабан. – Львів : Вид-во Держ. університету «Львівська політехніка», 2000. – 400 с.

5. Карабан В. І. Переклад англійської наукової і технічної літератури. Ч.І, ІІ / В. І. Карабан. – Вінниця : Нова книга, 2001. Ч.І – 271 с., ч.ІІ. – 302 с.

6. Інформаційні технології в перекладі : навчальний посібник / Т. І. Коваль, П. Г. Асоянц, Л. М. Артемчук та ін.; за заг. ред. д.п.н. Т. І. Коваль. – Київ : Вид. центр КНЛУ, 2010. – 261 с.

7. Корунець І. В. Теорія і практика перекладу / І. В. Корунець. – Вінниця : Нова книга. – 2000.

8. Коллін С.М.І. Англо-український словник комп'ютерних термінів / С.М.І. Коллін; пер. з англ. В. В. Воробйова. – Харків : Кн. Рекламне агентство «Ра», 2002. – 480 с.

9. Мірам Г. Алгоритми перекладу : Вступ. курс з формалізації перекладу (англ .. мовою) .- К .: Твім інтер , 1998.- 176 с .

10. Новікова М.О. Нове в теорії перекладу. Нові підходи до вивчення й викладання філології у вищій школі: наукова монографія. За ред.д-ра філол.н., проф. А.М.Науменка. Запоріжжя : ЗДУ, 1998.-Т. 3. С.70–73.

11. Свідер І. А. Інформаційні технології у філології та перекладі : навчально-методичний посібник для філологічних спеціальностей / І. А. Свідер. – Кам’янець-Подільський: ТОВ «Друкарня «Рута», 2021. – 184 с.

12. Тарасенко Р. О. Інформаційні технології в перекладацькій діяльності : навч. посіб. / Р. О. Тарасенко, С. М. Амеліна. – Київ : ЦП «КОМПРИНТ», 2017. – 253 с.

13. Чередниченко О. І., Коваль Я. І. Теорія і практика перекладу / О. І. Чередниченко, Я. І. Коваль. – Київ : Либідь, 1995. – 320 с.

14. Черноватий Л.М. Переклад англомовної технічної літератури. Електричне та електронне побутове устаткування. Комунікаційне устаткування. Виробництво та обробка металу : навч. посіб. / Л.М. Черноватий, В.І. Карабан, Омелянчук О.О. – Вінниця : Нова книга, 2006. – 296 с.

15. Черноватий Л.М. Принципи складання вправ для навчання галузевого перекладу / Л.М. Черноватий // Вісник Сумського державного університету. Серія «Філологічні науки». – 2002. – № 3 (36). – С. 101–105.

Навчальне видання

АКОП’ЯНЦ Нуну Михайлівна

ПЕРЕКЛАД У ГАЛУЗІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Навчальний посібник
для бакалаврів
спеціальності 035 «Філологія»,
спеціалізація 035.041 «Германські мови та літератури
(переклад включно), перша – англійська»
денної та заочної форм навчання

Англійською та українською мовами

Відповідальний за випуск доц. Голікова О.М.
Роботу до видання рекомендувала доц. Карачова Д. В.

В авторській редакції

План 2024, поз. 170

Гарнітура Times New Roman. Ум. друк. арк. 6

Видавничий центр НТУ «ХПІ»
Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.
61002, Харків, вул. Кирпичова, 2

Електронне видання