

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ
УНІВЕРСИТЕТ**

Факультет культури й виховання
Кафедра іноземних мов

АНГЛІЙСЬКА МОВА

Навчальний посібник

для аудиторної та самостійної роботи здобувачів
першого (бакалаврського) рівня вищої освіти
спеціальності F3(122) «Комп'ютерні науки»
денної форми здобуття вищої освіти

Миколаїв
2025

УДК 811.111'243(075.8)

A64

Друкується за рішенням науково-методичної комісії факультету культури й виховання від 27 травня 2025 р., протокол № 9

Укладачі:

- Т. А. Ганніченко – канд. пед. наук, доцент кафедри іноземних мов, Миколаївський національний аграрний університет
- О. О. Жебко – асистент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій, Миколаївський національний аграрний університет

Рецензенти:

- І. О. Калініна – д-р. тех. наук, доцент, професор кафедри інтелектуальних інформаційних систем Чорноморського національного університету імені Петра Могили;
- К. В. Тішечкіна – канд. філол. наук, доцент, завкафедри іноземних мов Миколаївського національного аграрного університету.

Англійська мова : навчальний посібник для аудиторної та самостійної роботи здобувачів першого (бакалаврського) рівня вищої освіти спеціальності F3(122) «Комп'ютерні науки» денної форми здобуття вищої освіти / уклад. Т. А. Ганніченко, О. О. Жебко. Миколаїв : МНАУ, 2025. 236 с.

УДК 811.111'243(075.8)

© Миколаївський національний аграрний університет, 2025

CONTENT

ПЕРЕДМОВА	5
LESSON 1 COMPUTING BASICS AND COMPUTER ARCHITECTURE	9
LESSON 2 OPERATING SYSTEMS AND SOFTWARE	18
LESSON 3 PROGRAMMING LANGUAGES AND ALGORITHMS	28
LESSON 4 COMPUTER NETWORKS AND THE INTERNET ..	38
LESSON 5 DATABASES AND DATA MANAGEMENT	48
LESSON 6 CYBERSECURITY AND DATA PROTECTION	57
LESSON 7 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING	67
LESSON 8 CLOUD COMPUTING AND EMERGING TECHNOLOGIES	77
LESSON 9 SOFTWARE DEVELOPMENT METHODOLOGIES AND VERSION CONTROL	87
LESSON 10 HUMAN-COMPUTER INTERACTION AND UX DESIGN	100
LESSON 11 COMPUTER GRAPHICS, MULTIMEDIA, AND SIGNAL PROCESSING	113
LESSON 12 ETHICS, SOCIETY, AND THE FUTURE OF COMPUTING	126
LESSON 13 EMBEDDED SYSTEMS AND THE INTERNET OF THINGS	140
LESSON 14 BLOCKCHAIN AND DISTRIBUTED LEDGER TECHNOLOGY	151
LESSON 15 QUANTUM COMPUTING	162

LESSON 16 PARALLEL AND DISTRIBUTED COMPUTING	173
LESSON 17 MOBILE APPLICATION DEVELOPMENT	184
LESSON 18 INFORMATION RETRIEVAL AND SEARCH ENGINES	195
LESSON 19 COMPILER DESIGN AND FORMAL LANGUAGES	206
LESSON 20 BIOINFORMATICS AND COMPUTATIONAL SCIENCE.....	217
REFERENCES.....	229

ПЕРЕДМОВА

Навчальний посібник з англійської мови призначений для аудиторної та самостійної роботи здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 122 «Комп'ютерні науки» денної форми здобуття вищої освіти. Посібник розроблений у рамках навчальної дисципліни «Іноземна мова за професійним спрямуванням» (ESP – English for Specific Purposes) відповідно до освітньо-професійної програми підготовки бакалаврів та навчального плану підготовки фахівців. Теоретичне й методичне підґрунтя посібника становлять компетентнісний, комунікативний та системно-діяльнісний підходи до навчання іноземної мови.

Посібник складається з 20 уроків, що забезпечують повний курс іншомовної підготовки здобувачів бакалаврського рівня. Рівень мовної підготовки відповідає рівню B2 Загальноєвропейських рекомендацій з мовної освіти (CEFR – Common European Framework of Reference for Languages): Independent User, Upper-Intermediate. На цьому рівні студент здатний читати й розуміти тексти за фахом, що містять спеціалізовану лексику; брати участь в усному та писемному обміні думками на теми, пов'язані зі своєю спеціальністю; готувати анотації та реферати англійською мовою; вести

листування й спілкуватися на фахові теми у стандартних академічних та ділових ситуаціях.

Головною метою посібника є формування іншомовної комунікативної компетентності студентів у сфері комп'ютерних наук: розвиток навичок читання та розуміння англomовних фахових текстів, засвоєння галузевої термінології, повторення та закріплення граматичних конструкцій у фаховому контексті, розвиток навичок усного та писемного мовлення на теми, пов'язані зі спеціальністю. Автентичні тексти сучасного змісту слугують основним джерелом фахової лексики і відображають актуальний стан розвитку інформаційних технологій.

Усі тексти посібника є оригінальними авторськими текстами, написаними відповідно до вимог рівня B2 та адаптованими під тематику спеціальності «Комп'ютерні науки». При написанні текстів автори спиралися на матеріали авторитетних академічних і технічних джерел, зазначених у списку використаних джерел. Тематика текстів відповідає сучасному стану розвитку комп'ютерних наук: вони охоплюють актуальні технології (генеративний штучний інтелект, великі мовні моделі, пост-квантова криптографія, IoT, хмарні обчислення нового покоління), а також фундаментальні концепції, знання яких є необхідним для розуміння сучасної фахової літератури. Тексти першого та другого в кожному уроці

різняться за функціональним спрямуванням: перший текст орієнтований на теоретичне ознайомлення з темою, другий – на її практичне, прикладне або соціальне вимірювання.

Граматичний матеріал посібника не є самоціллю, а підпорядкований завданню опанування фахового дискурсу. Конструкції, що відпрацьовуються у вправах, є типовими для наукового та технічного стилю англійської мови і регулярно зустрічаються у фаховій літературі.

Під час аудиторної роботи рекомендується виконувати вправи 1–6 (лексика першого тексту, переклад, заповнення пропусків, зіставлення, граматична вправа). Вправи 7–10 (другий текст, True/False, переклад, комунікативні завдання) можуть бути запропоновані для самостійної підготовки з наступним обговоренням на занятті. Вправу 10 (обговорення) рекомендується проводити у парах або малих групах із подальшим коротким підведенням підсумків.

Для самостійної роботи рекомендується: виконати вправи на переклад письмово до аудиторного заняття; після аудиторного заняття самостійно скласти 5–7 речень із новою лексикою уроку; підготувати короткий (5–7 речень) усний виступ на тему одного з питань вправи 10. Словниковий запас уроку найефективніше засвоюється через систематичне

повторення: рекомендується повертатися до таблиці лексики (вправа 1) перед кожним наступним заняттям.

Для розширення словникового запасу рекомендується використовувати онлайн-ресурси: Cambridge Dictionary (dictionary.cambridge.org), Merriam-Webster (merriam-webster.com), а також спеціалізовані технічні глосарії (Techopedia, Computer Hope).

Список літератури містить джерела, на основі яких укладено навчальний посібник, а також джерела, рекомендовані студентам для подальшого самостійного читання за тематикою кожного розділу. Для зручності список розподілено за тематичними блоками, що відповідають урокам посібника.

LESSON 1

COMPUTING BASICS AND COMPUTER ARCHITECTURE

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
hardware	апаратне забезпечення
software	програмне забезпечення
central processing unit (CPU)	центральний процесор
random access memory (RAM)	оперативна пам'ять
read-only memory (ROM)	постійна пам'ять
motherboard	материнська плата
input / output device	пристрій введення / виведення
binary system	двійкова система числення
bit / byte	біт / байт
integrated circuit	інтегральна схема
microprocessor	мікропроцесор
graphics processing unit (GPU)	графічний процесор
cache memory	кеш-пам'ять
bus	шина передачі даних

English term / word combination	Ukrainian equivalent
peripheral device	периферійний пристрій

2. Read and translate the text.

What Is a Computer?

The word 'computer' originally referred to a person who performed calculations. Today, however, it universally denotes an electronic device capable of processing, storing, and displaying information. The first computers were enormous machines that occupied entire rooms; modern devices range from smartphones small enough to fit in a pocket to supercomputers that fill dedicated facilities and can perform quadrillions of calculations per second.

All computers, regardless of size or purpose, rely on the same fundamental principle: data is represented in binary form, using only two values – 0 and 1. Each binary digit is called a bit; eight bits form a byte. Using combinations of bits, a computer can encode numbers, text, images, audio, video, and instructions. The binary system is used because electronic components can easily distinguish between two voltage levels – on and off – making it a reliable and efficient basis for computation.

The physical components of a computer constitute its hardware. The central processing unit (CPU) is the component that executes instructions; it fetches each instruction from memory, decodes it, and performs the required operation. Modern CPUs contain billions of transistors on a single chip and can execute several billion instructions per second. The CPU communicates with other components through a set of electrical pathways known as a bus.

Memory is divided into several types. Random access memory (RAM) is the working memory of the computer – it holds the data and program instructions that the CPU is currently using. RAM is volatile, meaning its contents are lost when power is removed. Read-only memory (ROM) contains permanent instructions, such as the firmware needed to start the computer. Long-term storage – hard disk drives (HDDs) and solid-state drives (SSDs) – retains data permanently, but is accessed more slowly than RAM.

The motherboard is the main circuit board that connects the CPU, memory, storage, and peripheral devices. Peripherals are devices connected to the computer to extend its capabilities: keyboards and mice for input, monitors and printers for output, and external drives or USB devices for additional storage or connectivity. The graphics processing unit (GPU), originally designed for rendering images, is now also widely used for parallel numerical computation, including tasks in artificial intelligence.

3. Give the Ukrainian equivalents to the following words and word combinations:

to perform calculations; to process information; to fit in a pocket; to rely on; binary form; voltage levels; to encode numbers; to execute instructions; to fetch an instruction; to decode; billions of transistors; electrical pathways; volatile memory; to retain data permanently; long-term storage; to extend capabilities; parallel numerical computation.

4. Fill in the gaps with the words from the list:

CPU, RAM, ROM, motherboard, binary, bit, byte, bus, GPU, peripheral

1. A _____ is the smallest unit of digital information, representing either 0 or 1.
2. Eight bits make one _____.
3. The _____ is the component that fetches, decodes, and executes instructions.
4. Working memory that holds currently active data and programs is called _____.
5. Permanent firmware instructions are stored in _____.
6. The _____ is the main circuit board connecting all internal components.

7. A set of electrical pathways that transfers data between components is called a _____.
8. All computer data is ultimately represented in _____ form, using 0s and 1s.
9. A keyboard, mouse, and monitor are examples of _____ devices.
10. The _____ was originally designed for rendering graphics but is now also used for AI computation.

5. Match the terms in column A with their definitions in column B:

#	Term		Definition
1	hardware	a	instructions and programs that run on a computer
2	software	b	memory that retains its contents without power
3	RAM	c	the physical components of a computer system
4	ROM	d	volatile memory used for active data and programs
5	CPU	e	a device connected externally to extend computer capabilities
6	peripheral	f	the component that executes program instructions

6. Insert the correct article (a / an / the) where necessary. Write '-' if no article is needed.

1. ___ CPU is ___ brain of every computer.
2. ___ binary system uses only two digits: 0 and 1.
3. A solid-state drive is ___ faster alternative to ___ hard disk drive.
4. ___ RAM is ___ volatile memory, meaning its contents are lost when ___ power is cut.
5. Modern ___ microprocessors contain billions of ___ transistors on ___ single chip.
6. ___ motherboard connects ___ CPU, ___ memory, and all ___ peripheral devices.

7. Read and translate the text.

Inside the CPU: Architecture and Performance

The performance of a CPU depends on several key factors: clock speed, the number of cores, cache size, and architecture. Clock speed, measured in gigahertz (GHz), indicates how many instruction cycles the processor can complete per second. A processor running at 3 GHz can perform three billion cycles per second. However, raw clock speed alone does not determine overall performance, because different architectures accomplish varying amounts of work per cycle.

Modern CPUs are multi-core processors, meaning they contain two or more independent processing units on a single chip.

Each core can execute its own stream of instructions simultaneously, allowing the processor to handle multiple tasks in parallel. Consumer processors typically have between four and sixteen cores; server-grade processors can have dozens. For software to take full advantage of multiple cores, it must be written to support parallel execution – a technique known as multithreading.

Cache memory is a small but extremely fast type of memory located on or very close to the CPU chip. It stores copies of frequently accessed data and instructions so that the CPU does not have to wait for slower RAM to supply them. Cache is organised into levels – L1, L2, and L3 – with L1 being the smallest and fastest, and L3 being larger but slightly slower. Having sufficient cache can dramatically improve performance in data-intensive tasks.

Instruction set architecture (ISA) defines the set of instructions a processor can execute and the way it interacts with memory and other hardware. The two dominant ISA families today are x86 (used by Intel and AMD processors in most personal computers and servers) and ARM (used in smartphones, tablets, and increasingly in laptops and servers due to its energy efficiency). Apple's M-series chips, for example, use an ARM-based architecture and have demonstrated that ARM processors can match or exceed x86 performance in many tasks.

The fetch-decode-execute cycle is the fundamental operational cycle of every CPU. In the fetch stage, the processor retrieves the next instruction from memory. In the decode stage, it interprets what the instruction requires. In the execute stage, it performs the operation – which may involve arithmetic, logic, memory access, or flow control. This cycle repeats billions of times per second, and the efficiency with which a processor completes each stage is at the heart of computer performance.

8. Define the sentences as True (T) or False (F). Correct the false sentences:

1. A higher clock speed always guarantees better overall CPU performance regardless of architecture.
2. A multi-core processor can execute multiple instruction streams simultaneously.
3. L1 cache is the largest and slowest level of cache memory.
4. ARM architecture is known for its energy efficiency and is used in smartphones and tablets.
5. The fetch-decode-execute cycle is the fundamental operational cycle of a CPU.
6. Software must support multithreading to benefit fully from a multi-core processor.

9. Give the English equivalents to the following words and word combinations:

тактова частота; багатоядерний процесор; незалежний потік команд; паралельне виконання; кеш-пам'ять; рівень кеш-пам'яті; набір команд; енергоефективність; цикл вибірки-декодування-виконання; арифметична операція; керування потоком; продуктивність комп'ютера; виробник процесорів; серверний процесор; потокова обробка даних.

10. Answer the questions:

- What are the main factors that determine CPU performance?
- What is the purpose of cache memory and how is it organised?
- What is the difference between x86 and ARM instruction set architectures?
- Why must software be written to support multithreading to benefit from multi-core processors?
- Describe the three stages of the fetch-decode-execute cycle.
- What is the role of the motherboard in a computer system?

LESSON 2

OPERATING SYSTEMS AND SOFTWARE

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
operating system (OS)	операційна система
kernel	ядро операційної системи
process	процес
thread	потік виконання
file system	файлова система
driver	драйвер
application software	прикладне програмне забезпечення
system software	системне програмне забезпечення
user interface (UI)	інтерфейс користувача
command line interface (CLI)	інтерфейс командного рядка
graphical user interface (GUI)	графічний інтерфейс користувача
virtual memory	віртуальна пам'ять
multitasking	багатозадачність

English term / word combination	Ukrainian equivalent
open source	відкрите програмне забезпечення
patch / update	патч / оновлення

2. Read and translate the text.

The Role of the Operating System

An operating system (OS) is the fundamental software layer that manages a computer's hardware resources and provides an environment in which application software can run. Without an operating system, each application would need to communicate directly with the hardware – an impractical arrangement given the enormous variety of hardware configurations in use. The OS abstracts the hardware, presenting applications with a consistent set of services regardless of the underlying physical components.

The kernel is the core of the operating system. It runs in a privileged mode with direct access to hardware and is responsible for the most critical tasks: managing processes and threads, allocating memory, controlling access to storage devices, and handling input and output operations. The kernel is always running in the background, mediating between hardware and the software that uses

it. Applications, by contrast, run in user mode and must make requests to the kernel whenever they need hardware resources.

Process management is one of the OS's most important functions. A process is a running instance of a program, complete with its own memory space and resources. A thread is a lighter unit of execution within a process; a single process may contain multiple threads that share memory but execute concurrently. Modern operating systems support multitasking – the ability to run many processes simultaneously by rapidly switching the CPU's attention between them. This switching happens so quickly that users perceive multiple tasks as running at the same time.

The file system is the component responsible for organising and managing data on storage devices. It defines how files are named, stored, retrieved, and protected. Common file systems include NTFS (used by Windows), ext4 (used by many Linux distributions), and APFS (used by macOS and iOS). The OS also provides memory management, including the concept of virtual memory – a technique that allows the system to use a portion of the hard drive as an extension of RAM, enabling programs larger than physical RAM to run.

The three dominant desktop operating systems today are Microsoft Windows, macOS, and Linux. Windows is the most widely used in consumer and enterprise environments. macOS is developed

by Apple exclusively for its own hardware. Linux is an open-source operating system available in many distributions, widely used in servers, embedded systems, and as the basis for Android. Each OS offers a graphical user interface for everyday tasks and a command-line interface for advanced operations and scripting.

3. Give the Ukrainian equivalents to the following words and word combinations:

hardware resources; application software; to abstract the hardware; privileged mode; to allocate memory; to handle input/output; to mediate between; user mode; running instance; memory space; to execute concurrently; to switch CPU attention; to organise data; to retrieve files; virtual memory; extension of RAM; enterprise environment; open-source distribution; to support multitasking; advanced scripting.

4. Fill in the gaps with the words from the list:

kernel, process, thread, multitasking, virtual memory, driver, CLI, GUI, file system, open source

1. The _____ is the core component of an OS that runs in privileged mode and manages hardware directly.

2. A _____ is a running instance of a program with its own memory and resources.
3. A _____ is a lighter unit of execution that shares memory with other threads in the same process.
4. _____ allows an OS to run many programs simultaneously by rapidly switching CPU attention.
5. _____ uses disk space to extend the effective size of available RAM.
6. A _____ is software that allows the OS to communicate with a specific hardware device.
7. A _____ allows users to interact with a computer through text commands.
8. A _____ provides a visual environment with windows, icons, and menus.
9. The _____ organises and manages how data is stored on and retrieved from a storage device.
10. Linux is an _____ operating system, meaning its source code is freely available.

5. Match words with similar meaning:

1) to manage	a) to simplify / hide complexity
2) to retrieve	b) simultaneous
3) to allocate	c) to control / to handle
4) concurrent	d) to assign / to distribute
5) to abstract	e) elevated / with special rights
6) privileged	f) to access / to fetch

6. Use the correct form of the verb in brackets:

1. The operating system (manage) _____ all hardware resources on behalf of applications.
2. Applications (run) _____ in user mode and cannot access hardware directly.
3. When a computer (start up) _____, the kernel (load) _____ into memory first.
4. Virtual memory (allow) _____ programs (use) _____ more memory than is physically available.
5. Modern operating systems (support) _____ multithreading, so a single application can (perform) _____ several tasks at once.
6. Linux distributions (use, widely) _____ in servers because of their stability and low cost.

7. Read and translate the text.

Software: Types, Development, and Licensing

Software is a set of instructions that tells a computer what to do. It is broadly categorised into system software, which manages and operates the hardware (including operating systems, device drivers, and utilities), and application software, which performs specific tasks for the user (including word processors, web browsers, games, and development tools). The distinction between these categories has

become less clear-cut as modern operating systems include more and more built-in applications.

Software development is the process of designing, writing, testing, and maintaining source code. Source code is human-readable text written in a programming language. It must be translated into machine code – binary instructions the CPU can execute – before a computer can run it. This translation is performed either by a compiler, which converts the entire program at once before execution, or by an interpreter, which translates and executes the code line by line at runtime.

The software development lifecycle (SDLC) describes the stages a software project passes through from initial conception to retirement. Common stages include requirements analysis, system design, implementation (coding), testing, deployment, and maintenance. Different development methodologies organise these stages differently: the Waterfall model completes each stage sequentially before moving to the next, while Agile approaches work in short iterative cycles, delivering small increments of working software frequently.

Software licensing governs the terms under which software may be used, copied, modified, and distributed. Proprietary software is owned by its developer; users pay for a licence and are generally not permitted to access or modify the source code. Open-source

software is distributed with its source code under a licence that grants users the right to study, modify, and redistribute it. Widely used open-source licences include the MIT licence, the Apache licence, and the GNU General Public Licence (GPL). Free and open-source software (FOSS) has become a cornerstone of modern computing infrastructure.

Software security is a growing concern. Vulnerabilities in software can be exploited by attackers to gain unauthorised access, steal data, or disrupt services. Developers are responsible for writing secure code: validating all user input, avoiding common vulnerabilities such as buffer overflows and SQL injection, applying the principle of least privilege, and keeping dependencies up to date. Regular security patches are essential to address newly discovered vulnerabilities before they can be exploited.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. System software performs specific tasks for the user, such as word processing or web browsing.
2. A compiler translates source code into machine code before execution.
3. The Agile methodology completes all development stages sequentially before delivery.

4. Open-source licences grant users the right to study, modify, and redistribute the software.
5. Buffer overflow and SQL injection are examples of common software vulnerabilities.
6. Proprietary software typically allows users to view and modify its source code.

9. Give the English equivalents to the following words and word combinations:

вихідний код; машинний код; компілятор; інтерпретатор; виконання програми; цикл розробки програмного забезпечення; аналіз вимог; системне проєктування; реалізація (програмування); тестування; розгортання; ліцензування програмного забезпечення; власницьке програмне забезпечення; відкрите програмне забезпечення; вразливість програмного забезпечення; ін'єкція SQL; переповнення буфера; принцип найменших привілеїв; патч безпеки; залежності програми.

10. Answer the questions:

- What is the difference between system software and application software?
- How does a compiler differ from an interpreter?
- What are the main stages of the software development lifecycle?

- What is the difference between proprietary and open-source software licences?
- What responsibilities do developers have regarding software security?
- Why is the Agile methodology often preferred over the Waterfall model in practice?

LESSON 3

PROGRAMMING LANGUAGES AND ALGORITHMS

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
programming language	мова програмування
algorithm	алгоритм
syntax	синтаксис
variable	змінна
function / method	функція / метод
loop	цикл
conditional statement	умовний оператор
data type	тип даних
object-oriented programming (OOP)	об'єктно-орієнтоване програмування
class / object	клас / об'єкт
inheritance	успадкування
recursion	рекурсія
data structure	структура даних
time complexity	часова складність

English term / word combination	Ukrainian equivalent
debugging	налагодження / виправлення ПОМИЛОК

2. Read and translate the text.

Programming Languages: From Low to High Level

A programming language is a formal notation system that allows humans to write instructions for a computer. Programming languages can be classified on a spectrum from low-level to high-level. Low-level languages are closer to the hardware: assembly language, for example, uses short mnemonics that correspond directly to machine instructions and gives programmers fine-grained control over hardware resources. High-level languages such as Python, Java, and C++ abstract away the hardware details, allowing programmers to express logic in terms closer to human reasoning.

Interpreted languages such as Python execute code through an interpreter at runtime, translating each instruction as it runs. This makes them flexible and easy to use interactively but generally slower than compiled languages. Compiled languages such as C and C++ are translated into machine code by a compiler before execution; the resulting executable runs directly on the CPU and is typically much

faster. Some languages, such as Java and C#, use a hybrid approach: source code is compiled to an intermediate bytecode, which is then executed by a virtual machine.

Programming paradigms describe different approaches to organising and expressing code. Procedural programming organises code into procedures (functions) that are called in sequence. Object-oriented programming (OOP) structures code around objects – instances of classes that encapsulate data and behaviour together. Four fundamental principles of OOP are encapsulation (hiding implementation details), inheritance (deriving new classes from existing ones), polymorphism (one interface, multiple implementations), and abstraction (focusing on what an object does, not how it does it).

Functional programming treats computation as the evaluation of mathematical functions and avoids changing state or mutable data. Languages such as Haskell are purely functional, while languages such as Python and JavaScript support functional programming as one of several paradigms. Scripting languages such as JavaScript, Bash, and PHP are used to automate tasks, manipulate text, and build web applications. The choice of programming language depends on the task, performance requirements, available libraries, and team expertise.

Syntax is the set of rules that defines the structure of valid statements in a programming language. A syntax error occurs when code violates these rules and prevents compilation or interpretation. Semantics refers to the meaning of statements – code may be syntactically correct but semantically incorrect if it does not do what the programmer intended. Debugging is the process of finding and fixing errors in code. Modern development environments provide powerful debugging tools that allow developers to step through code, inspect variable values, and identify the source of errors.

3. Give the Ukrainian equivalents to the following words and word combinations:

formal notation system; assembly language; fine-grained control; to abstract away; human reasoning; to execute at runtime; executable file; intermediate bytecode; virtual machine; to organise code; encapsulation; implementation details; mutable data; to automate tasks; to manipulate text; performance requirements; available libraries; syntax error; semantics; to step through code.

4. Fill in the gaps with the words from the list:

syntax, variable, loop, function, class, inheritance, recursion, data type, debugging, OOP

1. _____ is the set of rules that defines the valid structure of statements in a programming language.
2. A _____ is a named storage location that holds a value which can change during execution.
3. A _____ defines the kind of value a variable can hold, such as integer, float, or string.
4. A _____ is a block of code that performs a specific task and can be called multiple times.
5. A _____ is a control structure that repeats a block of code while a condition is true.
6. In _____, code is organised around objects that combine data and behaviour.
7. A _____ is a blueprint from which objects are created.
8. _____ allows a new class to acquire the properties and methods of an existing class.
9. _____ is a technique where a function calls itself to solve a problem by breaking it into smaller subproblems.
10. _____ is the process of identifying and correcting errors in a program.

5. Choose the most suitable answer:

1. Assembly language is classified as a low-level language because ____.
a) it is easy to learn and use b) it corresponds closely to machine instructions and hardware
c) it cannot be compiled into machine code d) it is only used in educational settings
2. Which of the following best describes polymorphism in OOP?
a) A class can only have one method with a given

- name b) Data is hidden inside an object and cannot be accessed externally c) One interface can have multiple implementations d) A new class is derived from an existing class
3. A compiled language differs from an interpreted language in that ____.
- a) it requires an internet connection to run
 - b) its source code is translated into machine code before execution
 - c) it can only be used for web development
 - d) it does not support object-oriented programming
4. Functional programming is characterised by ____.
- a) heavy use of loops and mutable variables
 - b) organising code into classes and objects
 - c) treating computation as the evaluation of mathematical functions
 - d) direct manipulation of hardware registers

6. Read and translate the text.

Algorithms and Data Structures

An algorithm is a finite, well-defined sequence of steps designed to solve a specific problem or perform a computation. Algorithms are at the heart of computer science: any task a computer performs – sorting a list, searching a database, routing a network packet, or training a machine learning model – can be described as an algorithm. The quality of an algorithm is measured in terms of correctness (does it produce the right answer?) and efficiency (how much time and memory does it require?).

The time complexity of an algorithm describes how its running time grows as the input size increases. It is typically expressed using Big O notation. An $O(1)$ algorithm executes in constant time regardless of input size. An $O(n)$ algorithm's running time grows linearly with input size. An $O(n^2)$ algorithm's time grows quadratically – acceptable for small inputs but problematic for large ones. Sorting algorithms provide a classic illustration: bubble sort has $O(n^2)$ complexity, while merge sort and quicksort achieve $O(n \log n)$, making them far more efficient for large datasets.

A data structure is a way of organising data in memory so that it can be used efficiently. The choice of data structure has a major impact on the performance of algorithms. An array stores elements in contiguous memory locations and allows direct access by index. A linked list stores elements as nodes, each containing data and a pointer to the next node; it allows efficient insertion and deletion but slow random access. A stack follows last-in-first-out (LIFO) order, while a queue follows first-in-first-out (FIFO) order.

Trees and graphs are more complex data structures. A binary search tree organises data hierarchically, allowing efficient searching, insertion, and deletion in $O(\log n)$ time on average. A graph represents entities (nodes) and the relationships between them (edges), and is used to model networks, maps, social connections, and many other real-world structures. Searching a graph efficiently

requires algorithms such as breadth-first search (BFS) and depth-first search (DFS).

Hash tables are data structures that map keys to values using a hash function. They provide $O(1)$ average-case time for insertion, deletion, and lookup, making them extremely fast for many common tasks. They are used in implementing dictionaries, database indexes, and caches. Understanding data structures and algorithms is fundamental to writing efficient software and is a core topic in computer science education and technical interviews.

7. Define sentences as True (T) or False (F). Correct the false sentences:

1. An algorithm with $O(n^2)$ complexity is more efficient than one with $O(n \log n)$ for large inputs.
2. A linked list allows efficient insertion and deletion but slow random access.
3. A stack follows first-in-first-out order.
4. A binary search tree allows efficient searching in $O(\log n)$ time on average.
5. Hash tables provide constant-time average-case performance for insertion, deletion, and lookup.
6. Graphs are used to represent entities and relationships such as social networks or maps.

8. Match the terms with their definitions:

#	Term		Definition
1	algorithm	a	a data structure mapping keys to values with fast lookup
2	Big O notation	b	a finite sequence of steps for solving a problem
3	array	c	a LIFO data structure used for managing function calls
4	hash table	d	a way of describing how an algorithm's time grows with input size
5	stack	e	a structure storing elements in contiguous memory locations
6	graph	f	a data structure representing entities and relationships between them

9. Give the English equivalents to the following words and word combinations:

кінцева послідовність кроків; ефективність алгоритму; часова складність; нотація O-велике; константний час виконання; лінійна складність; квадратична складність; сортування злиттям; масив; зв'язний список; покажчик на наступний вузол; стек; черга; дерево двійкового пошуку; граф; обхід у ширину; обхід у глибину; хеш-таблиця; хеш-функція; пошук за ключем.

10. Answer the questions:

- What two criteria are used to measure the quality of an algorithm?
- What does Big O notation describe and why is it important?
- What is the difference between an array and a linked list?
- In what situations would you use a hash table rather than an array?
- What is a binary search tree and what is its average-case time complexity for search operations?
- Name two real-world applications where graph data structures are used.

LESSON 4

COMPUTER NETWORKS AND THE INTERNET

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
local area network (LAN)	локальна мережа
wide area network (WAN)	глобальна мережа
protocol	протокол
IP address	IP-адреса
router	маршрутизатор
packet switching	пакетна комутація
TCP/IP	протокол управління передачею / Інтернет-протокол
DNS (Domain Name System)	система доменних імен
bandwidth	пропускна здатність
firewall	брандмауер / міжмережевий екран
HTTP / HTTPS	протокол передачі гіпертексту / захищений протокол
OSI model	модель OSI
Wi-Fi	бездротова мережа Wi-Fi

English term / word combination	Ukrainian equivalent
VPN (Virtual Private Network)	віртуальна приватна мережа
latency	затримка / час відгуку

2. Read and translate the text.

How Computer Networks Work

A computer network is a collection of interconnected devices that share resources and communicate with each other. Networks are classified by geographical scale: a local area network (LAN) connects devices within a limited area such as a home, office, or campus; a wide area network (WAN) spans larger distances, potentially connecting sites across different cities or countries. The internet is the largest WAN – a global network of networks that uses the TCP/IP protocol suite to enable communication between billions of devices worldwide.

Communication across a network requires agreement on a common set of rules known as protocols. The Internet Protocol (IP) is responsible for addressing and routing: every device on the internet has a unique IP address that identifies it, and routers use these addresses to forward data packets toward their destinations. IPv4 uses 32-bit addresses and supports approximately 4.3 billion unique

addresses, a number that has proved insufficient for the modern internet. IPv6 uses 128-bit addresses, providing a vastly larger address space.

The Transmission Control Protocol (TCP) works above IP, providing reliable, ordered delivery of data. When a large file or web page is transmitted, TCP breaks it into small packets that travel independently through the network and are reassembled at the destination. If packets are lost or corrupted in transit, TCP automatically requests retransmission. This makes TCP suitable for applications where data integrity is critical, such as file transfers and web browsing. The User Datagram Protocol (UDP), by contrast, prioritises speed over reliability – it sends packets without guaranteeing delivery and is used in real-time applications such as video streaming and online gaming.

The OSI (Open Systems Interconnection) model is a conceptual framework dividing network communication into seven layers, from the Physical layer (which deals with the transmission of raw bits over a medium) up to the Application layer (which provides network services to software applications). Each layer communicates only with the layers directly above and below it. This layered architecture allows different technologies to be combined and substituted independently: for example, a Wi-Fi physical connection

can be replaced by a fibre-optic cable without changing the higher-level protocols.

The Domain Name System (DNS) is the internet's addressing service, translating human-readable domain names such as `www.university.edu` into numerical IP addresses. Without DNS, users would need to memorise numbers to visit websites. A firewall is a security system that monitors and filters incoming and outgoing network traffic according to predefined rules, blocking communications that do not comply. VPNs encrypt all traffic between a device and a remote server, providing privacy and allowing users to access corporate networks securely over the public internet.

3. Give the Ukrainian equivalents to the following words and word combinations:

to share resources; geographical scale; to span distances; to enable communication; common set of rules; unique IP address; to forward data packets; 32-bit address; address space; reliable ordered delivery; to reassemble packets; data integrity; real-time application; raw bit transmission; physical medium; layered architecture; to substitute independently; human-readable domain name; to monitor network traffic; to comply with rules.

4. Fill in the gaps with the words from the list:

protocol, router, DNS, firewall, VPN, bandwidth, latency, TCP, UDP, OSI

1. A _____ is a set of rules that governs how devices communicate on a network.
2. A _____ is a device that forwards data packets between networks using IP addresses.
3. _____ ensures reliable, ordered delivery of data by managing retransmission of lost packets.
4. _____ prioritises speed and is used for real-time applications such as video streaming.
5. The _____ model divides network communication into seven layers.
6. _____ translates domain names into IP addresses.
7. A _____ monitors and filters network traffic to block unauthorised communications.
8. A _____ encrypts traffic between a device and a remote server to provide privacy.
9. _____ is the maximum rate at which data can be transmitted over a network connection.
10. _____ is the delay between sending a request and receiving a response.

5. Arrange the steps of a web page request in the correct order (1 = first):

1. The browser receives the IP address and sends an HTTP request to the web server.

2. The web server returns the requested page, which the browser displays.
3. The user types a URL into a browser.
4. TCP breaks the response into packets that travel through the network and are reassembled.
5. The browser queries the DNS server to resolve the domain name to an IP address.

6. Read and translate the text.

Wireless Networks, the Web, and Network Security

Wireless networking has transformed connectivity. Wi-Fi, based on the IEEE 802.11 standards, allows devices to communicate over radio frequencies without physical cables. Successive generations have brought higher speeds and improved performance in crowded environments: Wi-Fi 6 supports speeds up to 9.6 Gbps. Cellular networks – 3G, 4G LTE, and now 5G – provide mobile connectivity over wider areas. 5G promises not only faster speeds but also very low latency, making it suitable for applications such as autonomous vehicles and remote surgery.

The World Wide Web (WWW) is a system of interlinked documents and resources accessible over the internet using the HTTP and HTTPS protocols. Although the terms 'internet' and 'web' are often used interchangeably, they are distinct: the internet is the

underlying network infrastructure, while the web is a service that runs on top of it. Web pages are written in HTML, styled with CSS, and made interactive with JavaScript. Web servers deliver these resources in response to HTTP requests from browsers.

HTTPS (HyperText Transfer Protocol Secure) adds a layer of encryption using TLS (Transport Layer Security) to protect the confidentiality and integrity of data in transit. It ensures that communication between a browser and a web server cannot be intercepted or tampered with by third parties. The presence of HTTPS is indicated by a padlock icon in the browser address bar. Today, virtually all reputable websites use HTTPS, and browsers warn users when they visit HTTP-only sites.

Network security is a continuous challenge. Distributed Denial of Service (DDoS) attacks attempt to overwhelm a server or network with excessive traffic, rendering it unavailable to legitimate users. Man-in-the-middle attacks intercept communications between two parties without their knowledge. Intrusion detection systems (IDS) monitor network traffic for suspicious patterns and alert administrators to potential threats. Network segmentation – dividing a network into separate zones – limits the damage an attacker can cause if they gain access to one segment.

The transition from IPv4 to IPv6 is an ongoing process. The exhaustion of IPv4 addresses has been partially addressed through

Network Address Translation (NAT), which allows multiple devices on a private network to share a single public IP address. However, NAT introduces complexity and limitations. IPv6's adoption eliminates these workarounds and also includes built-in support for IPsec (Internet Protocol Security), providing improved security at the network layer.

7. Define sentences as True (T) or False (F). Correct the false sentences:

1. The internet and the World Wide Web are the same thing.
2. HTTPS uses TLS encryption to protect data in transit.
3. A DDoS attack attempts to make a service unavailable by flooding it with traffic.
4. IPv4 addresses are 128 bits long and provide a vast address space.
5. NAT allows multiple devices on a private network to share one public IP address.
6. 5G cellular networks offer very low latency compared to previous generations.

8. Match words with similar meaning:

1) to transmit	a) to translate / to look up
2) to intercept	b) to send / to transfer

3) to segment	c) to flood / to overload
4) to overwhelm	d) to divide / to separate
5) to comply	e) to capture / to block midway
6) to resolve (a name)	f) to conform / to follow rules

9. Give the English equivalents to the following words and word combinations:

бездротова мережа; радіочастота; пропускна здатність Wi-Fi; мобільний зв'язок; мала затримка; автономний транспортний засіб; Всесвітня павутина; мова розмітки гіпертексту; таблиця стилів; шифрування TLS; перехоплення даних; захист від третіх сторін; замок в адресному рядку; атака відмови в обслуговуванні; система виявлення вторгнень; сегментація мережі; трансляція мережевих адрес; протокол безпеки; перехід на IPv6; архітектура нульової довіри.

10. Answer the questions:

- What is the difference between the internet and the World Wide Web?
- Why is HTTPS preferred over HTTP for all modern websites?
- What is a DDoS attack and how can it be mitigated?

- What is Network Address Translation and why was it introduced?
- What new capabilities does 5G offer compared to 4G?
- What is network segmentation and why is it used as a security measure?

LESSON 5

DATABASES AND DATA MANAGEMENT

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
database management system (DBMS)	система управління базами даних
relational database	реляційна база даних
SQL (Structured Query Language)	мова структурованих запитів
table / record / field	таблиця / запис / поле
primary key	первинний ключ
foreign key	зовнішній ключ
query	запит
transaction	транзакція
ACID properties	властивості ACID
NoSQL database	нереляційна база даних
data normalisation	нормалізація даних
index	індекс
data warehouse	сховище даних

English term / word combination	Ukrainian equivalent
big data	великі дані
data pipeline	конвеєр обробки даних

2. Read and translate the text.

Relational Databases and SQL

A database is an organised collection of structured data stored and accessed electronically. Databases are managed by software systems called database management systems (DBMS), which provide tools for storing, retrieving, updating, and deleting data while ensuring consistency and integrity. The relational model, introduced by Edgar Codd at IBM in 1970, remains the dominant paradigm for storing structured data.

In a relational database, data is organised into tables, each representing a specific entity type – such as customers, products, or orders. Each table has columns (fields) defining the data it stores and rows (records) containing actual data values. A primary key uniquely identifies each record in a table. A foreign key links a record in one table to a record in another, establishing relationships between entities. These relationships eliminate data redundancy through a process called normalisation.

SQL (Structured Query Language) is the standard language for interacting with relational databases. It supports four main operations, often called CRUD: Create (INSERT), Read (SELECT), Update (UPDATE), and Delete (DELETE). A SELECT statement retrieves data from one or more tables, optionally filtering rows with a WHERE clause, sorting with ORDER BY, and aggregating with GROUP BY. JOIN operations combine data from multiple tables based on matching key values, enabling complex queries that span several related entities.

Database transactions ensure data integrity in multi-step operations. A transaction groups one or more SQL statements that must all succeed or all fail together. For example, transferring money between bank accounts requires debiting one account and crediting another: if either operation fails, neither should be applied. Transactions adhere to the ACID properties: Atomicity (all-or-nothing), Consistency (the database remains valid), Isolation (concurrent transactions do not interfere), and Durability (committed changes survive failures).

Database indexes improve query performance. An index is a data structure that allows the DBMS to locate rows quickly without scanning the entire table. Indexes speed up SELECT queries but add overhead to INSERT, UPDATE, and DELETE operations. Database administrators must balance query performance against write

performance when deciding which columns to index. Caching frequently accessed query results is another common technique for improving database performance.

3. Give the Ukrainian equivalents to the following words and word combinations:

organised collection; to retrieve data; to ensure integrity; dominant paradigm; column and row; to uniquely identify; to establish relationships; data redundancy; normalisation; CRUD operations; to aggregate data; JOIN operation; multi-step operation; to debit/credit an account; concurrent transactions; committed changes; database administrator; to balance performance; to scan a table; caching technique.

4. Fill in the gaps with the words from the list:

primary key, foreign key, query, transaction, index, normalisation, ACID, NoSQL, data warehouse, pipeline

1. A _____ uniquely identifies each record in a database table.
2. A _____ links a record in one table to a related record in another.

3. A _____ is a request sent to the database to retrieve or modify data.
4. A database _____ groups SQL operations that must all succeed or all fail together.
5. A database _____ speeds up row retrieval without scanning entire tables.
6. Data _____ is the process of organising tables to minimise redundancy.
7. The _____ properties (Atomicity, Consistency, Isolation, Durability) guarantee transaction reliability.
8. _____ databases do not use fixed table schemas and prioritise flexibility and scalability.
9. A _____ is a centralised repository optimised for analytical queries.
10. A data _____ is an automated sequence of steps that moves and transforms data from source to destination.

5. Match the SQL command with its function:

#	Term		Definition
1	SELECT	a	removes rows from a table
2	INSERT	b	retrieves data from one or more tables
3	UPDATE	c	adds new rows to a table
4	DELETE	d	modifies existing data in a table
5	JOIN	e	creates a new table or database object
6	CREATE	f	combines data from two or more tables

6. Read and translate the text.

NoSQL, Big Data, and Modern Data Engineering

The rise of web-scale applications in the 2000s exposed limitations of relational databases for certain workloads. Social networks and e-commerce platforms needed to store vast quantities of data with highly variable structure – data that did not fit neatly into rows and columns. This drove the development of NoSQL databases, a broad category that does not require a fixed schema and prioritises flexibility and horizontal scalability over strict relational consistency.

NoSQL databases come in several main types. Document databases such as MongoDB store data as flexible JSON-like documents, making them well-suited for content management and user profiles. Key-value stores such as Redis are extremely fast and simple, making them ideal for caching and session management. Column-family databases such as Apache Cassandra excel at reading and writing large volumes of data across distributed clusters. Graph databases such as Neo4j represent data as nodes and edges, making them ideal for social networks and recommendation systems.

Big data refers to datasets too large and complex for conventional tools to process efficiently. The three Vs of big data are Volume (enormous amounts), Velocity (high-speed generation), and

Variety (many formats and sources). Processing big data requires distributed computing frameworks such as Apache Spark, which distributes computation across clusters of machines. Stream processing – handling data continuously as it arrives – is used for real-time analytics, fraud detection, and monitoring.

ETL (Extract, Transform, Load) pipelines move data from source systems, transform it into a suitable format, and load it into a data warehouse or other destination. Modern organisations increasingly use ELT (Extract, Load, Transform), loading raw data first and transforming it later using the processing power of the destination system. Data lakes store raw data in any format, offering greater flexibility than traditional warehouses but requiring robust governance to remain useful.

Data governance encompasses the policies, standards, and practices that ensure data quality, security, and regulatory compliance. As organisations accumulate more data, managing accuracy, accessibility, and privacy becomes critical. Regulations such as the GDPR in Europe impose specific obligations on how personal data is collected, stored, processed, and deleted. Data lineage – tracking the origins and transformations of data through a pipeline – is a key component of transparency and auditability.

7. Define sentences as True (T) or False (F). Correct the false sentences:

1. NoSQL databases require a fixed, predefined schema for all stored data.
2. Redis is a key-value store commonly used for caching and session management.
3. The three Vs of big data are Volume, Velocity, and Variety.
4. In ETL pipelines, data is transformed before being loaded into the destination system.
5. Data lakes require rigid schemas and are less flexible than traditional data warehouses.
6. GDPR imposes obligations on organisations in Europe regarding personal data handling.

8. Match words with similar meaning:

1) to retrieve	a) reliable / uniform
2) to aggregate	b) to fetch / to access
3) to distribute	c) the ability to grow
4) consistent	d) unnecessary duplication
5) redundancy	e) to spread / to divide across
6) scalability	f) to combine / to summarise

9. Give the English equivalents to the following words and word combinations:

система управління базами даних; реляційна модель; таблиця і рядок; унікальний ідентифікатор; зовнішній ключ; нормалізація бази даних; мова запитів; CRUD-операції; об'єднання таблиць; транзакція бази даних; властивості ACID; документно-орієнтована база даних; розподілений кластер; великі дані; потокова обробка; конвеєр ETL; сховище даних; озеро даних; управління даними; відстеження походження даних.

10. Answer the questions:

- What is the relational model and how does it organise data?
- What do the ACID properties guarantee and why are they important?
- In what situations would you choose a NoSQL database over a relational database?
- What are the three Vs of big data and what does each represent?
- What is the difference between a data warehouse and a data lake?
- What is data governance and why is it important for organisations?

LESSON 6

CYBERSECURITY AND DATA PROTECTION

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
cybersecurity	кібербезпека
threat	загроза
vulnerability	вразливість
malware	шкідливе програмне забезпечення
phishing	фішинг
ransomware	програма-вимагач
encryption	шифрування
authentication	автентифікація
data breach	витік даних
zero-day exploit	атака нульового дня
social engineering	соціальна інженерія
multi-factor authentication (MFA)	багатофакторна автентифікація
intrusion detection system (IDS)	система виявлення вторгнень

English term / word combination	Ukrainian equivalent
patch	патч / виправлення безпеки
CIA triad	тріада ЦРД (конфіденційність, цілісність, доступність)

2. Read and translate the text.

Cyber Threats and Defensive Mechanisms

Cybersecurity is the practice of protecting computers, networks, software, and data from unauthorised access, damage, or disruption. As digital infrastructure becomes increasingly central to every sector of society – from banking and healthcare to critical national systems – securing that infrastructure has become one of the most pressing challenges of our time. The cost of cybercrime globally is estimated to reach trillions of dollars annually in direct financial losses, recovery costs, and reputational damage.

Cyber threats take many forms. Malware is any software designed to damage or gain unauthorised access to a system; it includes viruses, worms, trojans, and spyware. Ransomware, a particularly destructive subtype, encrypts the victim's files and demands payment – typically in cryptocurrency – before restoring access. In 2021, the Colonial Pipeline attack in the United States

demonstrated that ransomware could disrupt critical national infrastructure. Phishing attacks use deceptive emails or websites to trick users into revealing passwords, credit card numbers, or other sensitive information.

A vulnerability is a weakness in a system that can be exploited. Zero-day exploits target vulnerabilities that have not yet been discovered by the software vendor – and for which, therefore, no patch exists. Encryption transforms readable data into an unreadable form using a cryptographic algorithm; only parties possessing the correct key can decrypt it. Modern standards such as AES-256 are computationally infeasible to break with current technology. Authentication verifies the identity of a user or system; multi-factor authentication (MFA) requires two or more independent verification factors, significantly reducing account compromise risk.

The CIA triad – Confidentiality, Integrity, and Availability – defines the three core objectives of information security. Confidentiality ensures data is accessible only to authorised parties. Integrity means data is accurate and has not been tampered with. Availability ensures systems remain accessible to authorised users when needed. Attackers may target any of these properties: theft attacks violate confidentiality, tampering attacks violate integrity, and denial-of-service attacks violate availability.

Social engineering attacks exploit human psychology rather than technical vulnerabilities. An attacker might impersonate a senior executive, IT support representative, or government official to manipulate an employee into revealing credentials or transferring funds. Security awareness training – educating employees to recognise and resist these tactics – is one of the most cost-effective defensive measures available. Penetration testing simulates real attacks to identify weaknesses before genuine attackers can exploit them.

3. Give the Ukrainian equivalents to the following words and word combinations:

to protect from unauthorised access; pressing challenge; reputational damage; to gain access; to encrypt files; cryptocurrency; to disrupt infrastructure; deceptive email; to reveal passwords; sensitive information; software vendor; cryptographic algorithm; to possess a key; to reduce risk; account compromise; to be accessible to authorised parties; data integrity; denial-of-service attack; to impersonate an executive; to transfer funds.

4. Fill in the gaps with the words from the list:

vulnerability, phishing, ransomware, firewall, patch, MFA, data breach, encryption, malware, social engineering

1. _____ is the conversion of data into a coded form to prevent unauthorised reading.
2. A _____ controls network traffic, blocking communications that do not comply with security rules.
3. _____ refers to any software intentionally designed to harm or gain access to a system.
4. A _____ attack sends deceptive messages that appear to come from trusted sources.
5. When a company's database is illegally accessed and user data is exposed, this is called a _____.
6. A software _____ is a fix that addresses a known security weakness.
7. A system _____ is a weakness that could be exploited to gain unauthorised access.
8. _____ manipulates people psychologically to divulge confidential information.
9. _____ is a type of malware that locks files and demands payment for their release.
10. _____ requires more than one verification factor, making account takeover significantly harder.

5. Choose the most suitable answer:

1. A zero-day exploit is particularly dangerous because ____.
a) it only works at night when systems are unmonitored b)
no patch exists for the targeted vulnerability at the time of

- attack c) it requires physical access to the target machine
d) it only affects mobile devices
2. The CIA triad stands for _____. a) Cybercrime, Integration, Analysis b) Confidentiality, Integrity, Availability c) Certificates, Identification, Authentication d) Compliance, Investigation, Auditing
 3. Ransomware typically demands payment in cryptocurrency because _____. a) cryptocurrency is the only accepted currency on the dark web b) it enables attackers to receive payment with a degree of anonymity c) victims are required by law to use cryptocurrency d) bank transfers are too slow for ransom payments
 4. Multi-factor authentication improves security by _____. a) using longer and more complex passwords b) requiring two or more independent verification methods c) encrypting all outgoing network traffic d) blocking all logins from unknown IP addresses

6. Read and translate the text.

Cybersecurity in Organisations: Policies, Practices, and Regulation

For modern organisations, cybersecurity is not merely a technical concern but a strategic business priority. A significant data breach can cause severe financial consequences – regulatory fines, legal costs, customer compensation, and lasting reputational damage. High-profile incidents involving major corporations, hospitals, and government agencies have demonstrated that even well-resourced

organisations can fall victim to sophisticated attacks. As a result, many companies have elevated cybersecurity to board-level responsibility, with dedicated Chief Information Security Officers (CISOs).

A layered security approach – often called defence in depth – combines multiple defensive mechanisms so that a failure in any single layer does not compromise the entire system. Perimeter defences such as firewalls and intrusion detection systems form the outer layer. Internal controls such as access management, data loss prevention tools, and network segmentation limit the damage an attacker can cause after breaching the perimeter. Endpoint detection and response (EDR) tools monitor individual devices for signs of compromise.

Identity and access management (IAM) governs who can access which resources and under what conditions. The principle of least privilege dictates that each user and system component should have only the minimum access required to perform its function, reducing the potential impact of a compromised account. Role-based access control (RBAC) assigns permissions based on organisational roles rather than individual identity. Regular access reviews ensure that permissions are revoked when no longer needed.

Incident response is the organised approach to addressing and managing a security breach. An incident response plan defines the

steps to be taken when a security event is detected: contain the threat, eradicate the attacker's presence, recover affected systems, and conduct a post-incident review to learn from the event. Regular tabletop exercises test the plan without disrupting operations. Cyber insurance has become a common tool for managing the financial risk of incidents that cannot be prevented.

Regulatory frameworks increasingly impose specific cybersecurity obligations on organisations. The GDPR requires organisations operating in Europe to implement appropriate technical and organisational security measures and to notify authorities of certain data breaches within 72 hours. The NIS2 Directive extends cybersecurity requirements to a wider range of critical sectors. In the United States, frameworks such as NIST CSF provide voluntary guidance, while sector-specific regulations apply to healthcare (HIPAA) and financial services.

7. Define sentences as True (T) or False (F). Correct the false sentences:

1. Defence in depth means relying on a single strong security mechanism.
2. The principle of least privilege states that users should have only the minimum access needed for their role.
3. An incident response plan is only activated after a security breach has caused significant damage.

4. GDPR requires organisations to notify authorities of certain data breaches within 72 hours.
5. Penetration testing uses simulated attacks to find vulnerabilities before real attackers do.
6. Cyber insurance eliminates the financial risk of cybersecurity incidents.

8. Match the terms with their definitions:

#	Term		Definition
1	defence in depth	a	assigns permissions based on organisational roles
2	RBAC	b	the minimum access principle for users and systems
3	least privilege	c	a layered security approach with multiple defensive mechanisms
4	IAM	d	a tool that monitors devices for signs of compromise
5	EDR	e	governance of who can access which resources and under what conditions
6	incident response	f	the organised approach to managing a security breach

9. Give the English equivalents to the following words and word combinations:

стратегічний пріоритет бізнесу; регуляторний штраф; директор з інформаційної безпеки; ешелонований захист; система виявлення вторгнень; управління доступом; сегментація мережі; принцип найменших привілеїв; контроль доступу на основі ролей; перегляд прав доступу; план реагування на інциденти; усунення загрози; відновлення систем; навчальні навчання; кіберстрахування; регуляторна база; захист персональних даних; сповіщення про витік даних; галузеве регулювання; аудит безпеки.

10. Answer the questions:

- Why has cybersecurity become a board-level priority for organisations?
- What is defence in depth and how does it differ from relying on a single security mechanism?
- What is the principle of least privilege and why is it important?
- What are the main stages of an incident response plan?
- What obligations does GDPR impose on organisations regarding data breaches?
- Why is security awareness training considered one of the most cost-effective defensive measures?

LESSON 7

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
artificial intelligence (AI)	штучний інтелект
machine learning (ML)	машинне навчання
neural network	нейронна мережа
deep learning	глибинне навчання
training data	навчальні дані
supervised / unsupervised learning	навчання з учителем / без учителя
natural language processing (NLP)	обробка природної мови
computer vision	комп'ютерний зір
large language model (LLM)	велика мовна модель
generative AI	генеративний штучний інтелект
reinforcement learning	навчання з підкріпленням
bias (in AI)	упередженість / зміщення
inference	висновок / логічний вивід

English term / word combination	Ukrainian equivalent
overfitting	перенавчання моделі
autonomous system	автономна система

2. Read and translate the text.

Fundamentals of Artificial Intelligence and Machine Learning

Artificial intelligence (AI) refers to the ability of machines to perform tasks that typically require human intelligence, such as recognising speech, understanding text, identifying objects in images, or making decisions. The field of AI dates back to the 1950s, when researchers first explored whether machines could simulate human thought. Today, AI is no longer confined to research laboratories – it is embedded in the products and services that billions of people use daily.

Machine learning (ML), a central subfield of AI, is a method by which systems learn from data rather than following explicitly programmed rules. A machine learning model is exposed to large quantities of labelled examples during a training phase; it adjusts its internal parameters until it can generalise from those examples to new inputs. Supervised learning trains on labelled data – pairs of inputs and their correct outputs. Unsupervised learning finds structure in

unlabelled data, identifying clusters or patterns without being told what to look for.

Deep learning, a subfield of machine learning, uses artificial neural networks with many layers. These networks are loosely inspired by the structure of the human brain: they consist of interconnected nodes (neurons) organised in layers, and information flows from the input layer through multiple hidden layers to the output layer. During training, the network learns to adjust the weights of connections between neurons to minimise prediction errors. Deep learning has produced remarkable results in image recognition, speech recognition, and natural language processing.

Natural language processing (NLP) is the branch of AI concerned with enabling machines to understand and generate human language. Modern NLP systems are based on transformer models – a neural network architecture introduced in 2017 that processes entire sequences of words simultaneously rather than one at a time. Large language models (LLMs) such as GPT-4 are transformer-based models trained on enormous text corpora and can generate coherent, contextually appropriate text on virtually any topic. Generative AI refers to systems capable of creating new content – text, images, audio, or code – rather than merely classifying or predicting.

Reinforcement learning trains an agent to take actions in an environment by rewarding desirable outcomes and penalising

undesirable ones. The agent learns through trial and error, gradually discovering strategies that maximise cumulative reward. This approach has produced impressive results in game playing (AlphaGo, AlphaZero) and is being applied to robotics, resource management, and autonomous systems. Despite rapid progress, AI systems face important challenges including bias (models may reflect and amplify biases present in training data), lack of interpretability, and concerns about safety and alignment with human values.

3. Give the Ukrainian equivalents to the following words and word combinations:

to perform tasks requiring intelligence; to recognise speech; labelled examples; training phase; internal parameters; to generalise; unlabelled data; hidden layers; to minimise prediction errors; to process sequences simultaneously; enormous text corpora; coherent text; desirable outcomes; trial and error; cumulative reward; to amplify biases; lack of interpretability; alignment with human values; autonomous system; remarkable results.

4. Fill in the gaps with the words from the list:

neural network, training data, bias, inference, deep learning, overfitting, LLM, reinforcement, generative, supervised

1. _____ learning trains models on labelled input-output pairs.
2. The quality and representativeness of _____ directly affects model performance.
3. _____ uses many-layered artificial neural networks to learn complex patterns.
4. A _____ is a system of interconnected nodes that processes information in layers.
5. When applying a trained model to new data to make predictions, this is called _____.
6. _____ in an AI model occurs when training data reflects social or historical inequalities.
7. _____ AI generates new content such as text, images, or music.
8. In _____ learning, an agent learns by receiving rewards and penalties.
9. A large _____ is a transformer-based model trained on vast text data.
10. _____ happens when a model learns training data too precisely and performs poorly on new data.

5. Define sentences as True (T) or False (F). Correct the false sentences:

1. Machine learning systems require a programmer to write explicit rules for every situation.
2. Unsupervised learning finds patterns in data without being given labelled examples.

3. Deep learning neural networks have only two layers: input and output.
4. Large language models are based on the transformer architecture.
5. Reinforcement learning agents learn by minimising a loss function on labelled datasets.
6. AI bias can arise when training data reflects existing social inequalities.

6. Read and translate the text.

AI Applications, Ethics, and the Future

Artificial intelligence is transforming virtually every sector of the economy. In healthcare, ML models can analyse medical images to detect diseases such as diabetic retinopathy and certain cancers at accuracy levels comparable to specialist physicians. Drug discovery is being accelerated by AI systems that predict how molecules interact. In finance, AI is used for fraud detection, credit scoring, and algorithmic trading. Recommendation systems on streaming platforms and e-commerce sites use ML to personalise content for hundreds of millions of users simultaneously.

Computer vision allows machines to interpret visual information. It underpins facial recognition systems used in security and smartphone unlocking, quality inspection systems in

manufacturing, and the perception systems of autonomous vehicles. Self-driving cars must combine computer vision with sensor fusion, real-time path planning, and decision-making under uncertainty – combining several AI domains in a safety-critical application. Progress has been significant, but fully autonomous driving in complex environments remains an unsolved challenge.

The rapid advancement of generative AI has raised profound questions about intellectual property, misinformation, and the future of creative and knowledge work. AI systems can now generate photorealistic images, convincing text, and functional code. Deepfakes – AI-generated videos that realistically depict people saying or doing things they did not – pose serious risks to trust and public discourse. Detecting AI-generated content is an active research area. Regulatory frameworks for AI are under development in many jurisdictions, including the EU AI Act.

AI alignment refers to the challenge of ensuring that AI systems pursue goals consistent with human values and intentions. As AI systems become more capable, misalignment – where a system optimises for a proxy objective rather than the intended goal – becomes more concerning. The field of AI safety research studies how to build systems that remain robust, reliable, and aligned under a wide range of conditions, including adversarial inputs and unexpected deployment environments.

The demand for AI talent is growing rapidly. Roles in machine learning engineering, data science, and AI research are among the most sought-after in technology. However, AI skills are not sufficient in isolation: professionals who understand both the technical and ethical dimensions of AI – including fairness, privacy, accountability, and transparency – are increasingly valued. Computing students who develop this broader perspective are well-positioned for careers that will be defined by the responsible deployment of intelligent systems.

7. Define sentences as True (T) or False (F). Correct the false sentences:

1. AI models in healthcare have demonstrated diagnostic accuracy comparable to specialist physicians in some domains.
2. Fully autonomous driving in all complex environments is a fully solved technical problem.
3. Deepfakes are AI-generated videos that can realistically depict false events.
4. AI alignment is the challenge of ensuring AI systems consistently optimise for human well-being.
5. The EU AI Act is the first regulatory framework addressing risks from artificial intelligence systems.
6. Professionals who understand only the technical aspects of AI are considered the most valuable in the industry.

8. Match the terms with their definitions:

#	Term		Definition
1	computer vision	a	generating new content such as text, images, or code
2	generative AI	b	ensuring AI systems pursue goals consistent with human values
3	deepfake	c	AI-generated video depicting false events or statements
4	overfitting	d	the ability of machines to interpret visual information
5	AI alignment	e	training a model on labelled input-output pairs
6	supervised learning	f	a model that performs poorly on new data due to excessive memorisation

9. Give the English equivalents to the following words and word combinations:

перетворення галузі охорони здоров'я; розпізнавання медичних зображень; прискорення розробки ліків; виявлення шахрайства; кредитний скоринг; алгоритмічна торгівля; рекомендаційна система; персоналізація контенту; комп'ютерний зір; розпізнавання облич; автономний транспортний засіб; обробка зображень; генеративний контент; дипфейк; інтелектуальна власність; регуляторна база;

вирівнювання ІІІ; безпека ІІІ; попит на спеціалістів; відповідальне впровадження.

10. Answer the questions:

- In what areas of healthcare is AI currently showing the most significant results?
- What challenges remain in the development of fully autonomous vehicles?
- What risks does the rapid advancement of generative AI pose to society?
- What is AI alignment and why is it considered an important research problem?
- What combination of skills makes AI professionals particularly valuable?
- How do you think AI will change the profession of a software developer in the next decade?

LESSON 8

CLOUD COMPUTING AND EMERGING TECHNOLOGIES

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
cloud computing	хмарні обчислення
infrastructure as a service (IaaS)	інфраструктура як послуга
platform as a service (PaaS)	платформа як послуга
software as a service (SaaS)	програмне забезпечення як послуга
virtualisation	віртуалізація
containerisation	контейнеризація
scalability	масштабованість
microservices architecture	мікросервісна архітектура
Internet of Things (IoT)	Інтернет речей
blockchain	блокчейн
distributed ledger	розподілена книга обліку
quantum computing	квантові обчислення
qubit	кубіт

English term / word combination	Ukrainian equivalent
edge computing	крайові обчислення
DevOps	ДевОпс

2. Read and translate the text.

Cloud Computing: Principles, Models, and Benefits

Cloud computing is the delivery of computing resources – including servers, storage, databases, networking, software, and analytics – over the internet on a pay-as-you-go basis. Rather than purchasing and maintaining physical hardware, organisations rent access to resources from cloud providers such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud. This model converts large capital expenditures into predictable operational costs, reduces time-to-market, and enables organisations to scale resources up or down in response to demand.

Cloud services are categorised into three main models. Infrastructure as a Service (IaaS) provides virtualised computing resources – virtual machines, storage, and networking – while the customer manages the operating system and applications. Platform as a Service (PaaS) adds a managed runtime environment, databases, and development tools, allowing developers to focus on writing

applications without managing infrastructure. Software as a Service (SaaS) delivers fully functional applications over the internet; users access them through a browser without managing anything below the application layer.

Virtualisation is the foundational technology of cloud computing. It allows a single physical server to run multiple virtual machines (VMs) simultaneously, each isolated from the others and appearing as a fully independent computer. Containerisation, exemplified by Docker and orchestrated by Kubernetes, goes further by packaging applications with all their dependencies into lightweight, portable containers that start and stop rapidly. Containers are smaller and faster than VMs and are well-suited for microservices architectures, where applications are decomposed into small, independently deployable services.

Cloud deployments may be public (infrastructure shared among multiple customers, owned by the provider), private (dedicated infrastructure for a single organisation), or hybrid (combining both to balance control, cost, and compliance). Redundancy and geographic distribution allow cloud platforms to offer high availability and disaster recovery: data is replicated across multiple regions, so a hardware failure or natural disaster in one location does not result in service interruption. The recovery time objective can be reduced from hours to seconds.

DevOps is a cultural and technical movement that integrates software development and IT operations to shorten delivery cycles and improve software quality. CI/CD (Continuous Integration / Continuous Delivery) pipelines automate the building, testing, and deployment of code, allowing teams to release changes frequently and reliably. Infrastructure as Code (IaC) manages server configuration through version-controlled scripts, making infrastructure reproducible and auditable. Cloud platforms are natural enablers of DevOps practices, providing the automation and programmable infrastructure these practices require.

3. Give the Ukrainian equivalents to the following words and word combinations:

to deliver computing resources; pay-as-you-go model; capital expenditure; to rent access; cloud provider; virtual machine; lightweight container; to orchestrate; independently deployable service; shared infrastructure; geographic distribution; high availability; disaster recovery; data replication; service interruption; delivery cycle; to automate deployment; version-controlled script; reproducible infrastructure; programmable infrastructure.

4. Fill in the gaps with the words from the list:

IaaS, SaaS, PaaS, virtualisation, containerisation, scalability, microservices, DevOps, edge computing, IoT

1. _____ is the cloud model that delivers fully functional applications via a web browser.
2. _____ is the ability of a system to handle growing workloads by adding resources.
3. _____ allows one physical server to run multiple isolated virtual machines.
4. _____ packages applications with their dependencies into lightweight, portable units.
5. In the _____ model, the provider manages hardware and the customer manages OS and applications.
6. _____ architecture structures applications as collections of small, independent services.
7. _____ integrates development and operations to enable faster, more reliable software delivery.
8. _____ refers to the network of physical objects embedded with sensors and connectivity.
9. _____ places computing resources closer to data sources to reduce latency.
10. In the _____ model, developers build on a managed platform without managing the underlying infrastructure.

5. Match the terms with their definitions:

#	Term		Definition
1	qubit	a	a digital, decentralised record of transactions
2	blockchain	b	a technology placing computation close to where data is generated
3	edge computing	c	the basic unit of quantum information
4	smart contract	d	a self-executing agreement coded on a blockchain
5	redundancy	e	duplicating components to prevent single points of failure
6	CI/CD	f	automated pipeline for building, testing, and deploying code

6. Read and translate the text.

Emerging Technologies: IoT, Blockchain, and Quantum Computing

The Internet of Things (IoT) refers to the growing network of physical devices – from home appliances and wearable sensors to industrial machinery and city infrastructure – that are embedded with sensors, software, and connectivity. IoT devices collect and transmit data continuously, enabling applications such as smart home automation, precision agriculture, connected healthcare, and industrial monitoring. The number of connected IoT devices is

projected to exceed 30 billion by 2030, generating vast quantities of data that cloud and edge computing platforms must process and analyse.

Security is a central challenge in IoT. Many devices are manufactured with minimal security features – default passwords, outdated firmware, and no mechanism for remote updates. Compromised IoT devices have been weaponised in large-scale DDoS attacks. Ensuring the security of billions of small, low-power devices requires new security protocols, standardised update mechanisms, and regulation. Interoperability – enabling devices from different manufacturers to work together – remains an ongoing challenge addressed by industry alliances and emerging standards.

Blockchain is a distributed ledger technology that stores records in a chain of cryptographically linked blocks. Each block contains transactions, a timestamp, and a hash of the previous block. Altering any block changes its hash, invalidating all subsequent blocks – making the chain tamper-resistant. Blockchain's decentralised, trustless design has applications beyond cryptocurrency: supply chain transparency, digital identity, voting systems, and smart contracts – self-executing agreements coded directly on the chain.

Quantum computing exploits quantum mechanical phenomena – superposition and entanglement – to process

information in fundamentally different ways from classical computers. A classical bit is either 0 or 1; a qubit can be in a superposition of both simultaneously. Entangled qubits share correlated states, enabling massively parallel computation for certain problem types. Quantum computers could solve problems – such as factoring large numbers or simulating molecular interactions – that are intractable for classical machines. However, building stable, error-corrected quantum computers at useful scale remains a major engineering challenge.

The implications of quantum computing for cybersecurity are significant. Shor's algorithm could break widely used public-key cryptography (RSA, ECC) if run on a sufficiently powerful quantum computer. This has prompted investment in post-quantum cryptography – new algorithms resistant to quantum attacks, currently being standardised by NIST. While large-scale quantum computers capable of breaking current encryption are not yet available, organisations handling sensitive data with long confidentiality requirements are beginning to plan transitions to quantum-resistant cryptographic standards.

7. Define sentences as True (T) or False (F). Correct the false sentences:

1. IoT devices transmit data to enable applications such as smart home automation and industrial monitoring.
2. Blockchain's tamper-resistance comes from the cryptographic linking of blocks.
3. A qubit can only represent either 0 or 1, just like a classical bit.
4. Shor's algorithm could be used to break current public-key encryption on a sufficiently powerful quantum computer.
5. Large-scale quantum computers capable of breaking RSA encryption are widely available today.
6. Post-quantum cryptography develops new algorithms resistant to attacks from quantum computers.

8. Match words with similar meaning:

1) to deploy	a) distributed / not controlled by one party
2) to replicate	b) to launch / to release
3) to orchestrate	c) can grow to meet demand
4) tamper-resistant	d) to coordinate / to manage
5) decentralised	e) to duplicate / to copy
6) scalable	f) secure against modification

9. Give the English equivalents to the following words and word combinations:

хмарний провайдер; віртуальна машина; контейнеризація додатків; мікросервісна архітектура; безперервна інтеграція та доставка; інфраструктура як код; Інтернет речей; вбудований сенсор; промисловий моніторинг; безпека пристроїв IoT; блокчейн; криптографічно пов'язані блоки; розумний контракт; квантові обчислення; квантова суперпозиція; квантове заплутування; паралельні обчислення; постквантова криптографія; крайові обчислення; розподілена книга обліку.

10. Answer the questions:

- What are the three main cloud service models and what does each provide?
- What is the difference between virtualisation and containerisation?
- What security challenges does widespread IoT adoption present?
- How does blockchain achieve tamper-resistance?
- What is the difference between a classical bit and a qubit?
- Why are organisations beginning to prepare for post-quantum cryptography now, even though large-scale quantum computers do not yet exist?

LESSON 9

SOFTWARE DEVELOPMENT METHODOLOGIES AND VERSION CONTROL

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
software development lifecycle (SDLC)	життєвий цикл розробки програмного забезпечення
agile methodology	гнучка методологія розробки
sprint	спринт (ітераційний цикл)
Scrum	Скрам (фреймворк управління проектами)
product backlog	беклог продукту
version control system (VCS)	система контролю версій
repository	репозиторій
commit	коміт (фіксація змін у репозиторії)
branch	гілка коду
merge / pull request	злиття гілок / запит на злиття
code review	перегляд коду

English term / word combination	Ukrainian equivalent
continuous integration / delivery (CI/CD)	безперервна інтеграція / доставка
technical debt	технічний борг
refactoring	рефакторинг
unit testing	модульне тестування

2. Read and translate the text.

From Waterfall to Agile: How Software Projects Are Managed

Software development is a complex, collaborative process that benefits from structured approaches to planning, execution, and quality control. The software development lifecycle (SDLC) describes the stages a project passes through from initial conception to final retirement. Early formalised approaches were linear and sequential. The Waterfall model, for example, divides development into distinct phases – requirements, design, implementation, testing, deployment, and maintenance – each of which must be completed before the next begins. While this works well when requirements are stable and fully understood from the outset, it struggles to accommodate the change and uncertainty that characterise most real-world software projects.

The Agile Manifesto, published in 2001 by seventeen experienced software practitioners, articulated a different philosophy. It prioritised individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a fixed plan. These values reflect hard-won experience: many large software projects had failed or delivered poor results under rigid, plan-driven approaches, wasting enormous resources. Agile methodologies organise work into short, iterative cycles called sprints, typically one to four weeks long. At the end of each sprint, a working and potentially shippable increment of the product is delivered, reviewed by stakeholders, and used to inform the plan for the next sprint.

Scrum is the most widely adopted Agile framework. It defines three core roles: the Product Owner, who represents stakeholder interests and prioritises the product backlog – an ordered list of all desired features and requirements; the Scrum Master, who facilitates the process and removes obstacles; and the Development Team, which builds the product. Scrum structures each sprint with four ceremonies: sprint planning at the start, a brief daily stand-up meeting to synchronise progress, a review with stakeholders to inspect the increment at the end, and a retrospective in which the team reflects on its way of working and identifies improvements. Kanban is a

complementary method that visualises work as cards moving across a board whose columns represent workflow stages, helping teams identify bottlenecks and manage work in progress limits.

Test-driven development (TDD) is a disciplined practice in which developers write an automated failing test that precisely describes the desired behaviour of a unit of code before writing the implementation. The red-green-refactor cycle repeats: write a failing test (red), write the minimum code to make it pass (green), then improve the code's design without changing its behaviour (refactor). This approach encourages developers to think clearly about requirements before implementation, leads to more modular and better-designed code, and produces a comprehensive regression test suite that provides confidence when making future changes. Behaviour-driven development (BDD) extends TDD by expressing test scenarios in natural language readable by non-technical stakeholders, promoting shared understanding of requirements.

Technical debt is the accumulated cost of design shortcuts and suboptimal decisions taken during development – choosing the quick solution over the better one because of time pressure or incomplete understanding. Like financial debt, technical debt accrues interest: the longer it goes unaddressed, the harder the codebase becomes to understand, extend, and maintain. Symptoms include long, tangled functions, duplicated code, missing tests, and poor naming.

Refactoring – the disciplined restructuring of code to improve its internal quality without changing its observable behaviour – is the primary tool for reducing technical debt. Regular refactoring, supported by a comprehensive automated test suite, keeps the codebase healthy and reduces the risk that future changes introduce new defects.

3. Give the Ukrainian equivalents to the following words and word combinations:

structured approach; distinct phases; stable requirements; hard-won experience; iterative cycle; shippable increment; to represent stakeholder interests; to remove obstacles; daily stand-up; sprint retrospective; to identify bottlenecks; work in progress limit; automated failing test; desired behaviour; red-green-refactor cycle; regression test suite; time pressure; incomplete understanding; tangled function; duplicated code.

4. Fill in the gaps with the words from the list:

sprint, backlog, refactoring, unit testing, technical debt, CI/CD, Scrum Master, retrospective, TDD, Waterfall

1. The _____ model organises development into sequential phases that must each be completed before the next begins.
2. In Scrum, a _____ is a short fixed-length iteration, typically one to four weeks.
3. The product _____ is the ordered list of all features and tasks the team needs to complete.
4. The _____ facilitates the Scrum process and helps the team overcome obstacles.
5. In _____, a failing test is written before the implementation code.
6. _____ verifies the behaviour of individual functions or methods in isolation from the rest of the system.
7. _____ is the accumulated cost of shortcuts and suboptimal design decisions made during development.
8. _____ improves the internal quality of code without changing its observable behaviour.
9. _____ pipelines automatically build, test, and deploy code when changes are pushed to the repository.
10. A sprint _____ is a meeting where the team reflects on its process and agrees on improvements.

5. Match the Agile Manifesto values (column A) with their explanations (column B):

A	B
1) individuals and interactions	a) adjusting plans when understanding or circumstances evolve

A	B
2) working software	b) delivering a functional product matters more than exhaustive paperwork
3) customer collaboration	c) people and direct communication are more valuable than rigid procedures
4) responding to change	d) close, ongoing cooperation between developers and the client throughout the project

6. Rewrite the sentences in the Passive Voice:

1. The team holds a sprint review at the end of every iteration.
2. Developers write automated tests before implementing new features.
3. The Product Owner prioritises the items in the product backlog.
4. Engineers use refactoring to improve the internal structure of the code.
5. The CI/CD pipeline automatically builds and tests all code changes.
6. The team identifies sources of technical debt during the sprint retrospective.

7. Read and translate the text.

Version Control: Git, Collaboration, and Code Quality

Version control systems (VCS) are among the most essential tools in modern software development. They track every change made to a codebase over time, record who made each change, when it was made, and why, and allow the entire team to revert to earlier states if a problem is introduced. Without version control, coordinating work across a team would be chaotic: changes made by different developers would overwrite each other, the history of modifications would be lost, and diagnosing when and how a defect was introduced would be extremely difficult. A complete, navigable history of changes is also indispensable for auditing, compliance, and onboarding new team members.

Git, created by Linus Torvalds in 2005 to manage the Linux kernel source code, has become by far the dominant version control system worldwide. Git is distributed: every developer holds a complete copy of the repository – including its full history – on their own machine. This eliminates dependency on a central server for daily work, allows development to continue offline, and provides natural redundancy against server failures. Platforms such as GitHub, GitLab, and Bitbucket host Git repositories and extend them with web-based interfaces, issue trackers, project management boards, and integration with CI/CD pipelines.

Branching is one of Git's most powerful features. A branch is an independent line of development that diverges from a shared base. Developers create feature branches to implement new functionality in isolation, and bug-fix branches to address defects without disturbing ongoing development. When the work on a branch is complete and reviewed, it is merged back into the main branch. A pull request (or merge request on GitLab) is the mechanism through which this process is formalised: the developer requests that their changes be reviewed by colleagues before merging. Reviewers examine the code for correctness, performance, security, adherence to style guidelines, and maintainability – code review is one of the most effective quality assurance practices available, catching defects before they reach production and spreading knowledge across the team.

Continuous integration (CI) is the practice of automatically building and running the full automated test suite whenever a developer pushes changes to the shared repository. By integrating frequently – ideally multiple times per day – teams detect integration conflicts and test failures early, when they are small and easy to fix, rather than allowing them to accumulate into large, complex problems before a scheduled release. Continuous delivery (CD) extends CI by automatically deploying successfully tested code to a staging environment and, with appropriate approval gates, to production. Together, CI/CD pipelines enable teams to release software reliably

and frequently, reducing deployment risk and shortening the time between writing code and delivering value to users.

Code quality encompasses readability, maintainability, testability, security, and the absence of known defects. Beyond automated testing, code quality is supported by static analysis tools that scan source code for potential bugs, security vulnerabilities, and deviations from coding standards without executing it. Code coverage metrics measure what proportion of the codebase is exercised by automated tests; while high coverage increases confidence in the test suite, it does not by itself guarantee correctness. Meaningful variable names, well-structured functions of limited scope, clear comments, and up-to-date documentation collectively reduce the cognitive burden of working with a codebase and make it more resilient to the inevitable changes it will undergo over its lifetime.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. Git is a centralised version control system that requires developers to be connected to a server at all times.
2. Branching allows developers to work on features or fixes in isolation without disturbing the main codebase.
3. A pull request is a mechanism for requesting that code changes be reviewed before being merged.

4. Continuous integration runs tests automatically whenever code is pushed to the shared repository.
5. High code coverage is a sufficient guarantee that software is free of all bugs.
6. Static analysis tools execute the code in order to find potential security vulnerabilities.

9. Choose the most suitable answer:

1. Git is described as a distributed version control system because _____.
a) it stores the repository on a single central server
b) every developer has a full copy of the repository on their machine
c) it automatically resolves all merge conflicts
d) it requires an internet connection to commit changes
2. The main purpose of a pull request is to _____.
a) download the latest changes from the remote repository
b) automatically deploy code to production
c) request that colleagues review and approve code changes before merging
d) create a new branch from the main codebase
3. Continuous integration improves software quality by _____.
a) automatically writing tests for new code
b) detecting integration conflicts and test failures early and frequently
c) preventing developers from pushing code more than once per day
d) replacing manual code review with automated tools
4. Technical debt is best managed by _____.
a) avoiding the use of automated tests
b) delaying all design improvements until after the product is released
c) regular refactoring supported by a comprehensive automated test suite
d) rewriting the entire codebase from scratch periodically

10. Give the English equivalents to the following words and word combinations:

система контролю версій; відстежувати зміни; повернутися до попереднього стану; розподілена система; повна копія репозиторію; відстеження задач; гілка для нової функції; виправлення дефекту; злиття гілки; перегляд коду; затвердити зміни; безперервна інтеграція; автоматичне розгортання; стадійне середовище; частота випусків; технічний борг; рефакторинг; статичний аналіз коду; покриття тестами; читабельність коду.

11. Arrange the steps of a typical Git-based workflow in the correct order (1 = first):

- a) The developer creates a pull request asking colleagues to review the changes.
- b) Reviewers approve the pull request and the branch is merged into the main branch.
- c) A developer creates a new feature branch from the main branch.
- d) The CI/CD pipeline builds and tests the merged code automatically.
- e) The developer writes code, commits changes with descriptive messages, and pushes to the remote repository.

12. Answer the questions and discuss in a group:

- What are the main advantages of Agile over the Waterfall model for most real-world projects?
- What are the three roles in Scrum and what is each responsible for?
- What is the red-green-refactor cycle in TDD and what are its benefits?
- Why is technical debt described as a form of debt that accrues interest?
- What advantages does a distributed VCS like Git have over a centralised system?
- Why is code review considered one of the most effective quality assurance practices?
- In your opinion, is it always possible to maintain zero technical debt in a commercial project? Justify your answer.

LESSON 10

HUMAN-COMPUTER INTERACTION AND UX DESIGN

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
human-computer interaction (HCI)	взаємодія людини з комп'ютером
user experience (UX)	досвід користувача
user interface (UI)	інтерфейс користувача
usability	зручність використання
accessibility	доступність
wireframe	каркасний макет / вайрфрейм
prototype	прототип
user research	дослідження користувачів
usability testing	тестування зручності використання
cognitive load	когнітивне навантаження
affordance	зрозумілість елемента інтерфейсу
responsive design	адаптивний дизайн

English term / word combination	Ukrainian equivalent
information architecture (IA)	інформаційна архітектура
A/B testing	А/В тестування
persona	персонаж / образ типового користувача

2. Read and translate the text.

Principles of Human-Computer Interaction and Usability

Human-computer interaction (HCI) is the academic and professional field concerned with the design, evaluation, and implementation of interactive computing systems for human use. As computing has become embedded in virtually every aspect of daily life – from banking and healthcare to education and entertainment – ensuring that these systems are usable, accessible, and enjoyable to interact with has become critically important. Poor interface design is not merely inconvenient: in high-stakes environments such as aviation, nuclear power plants, and medical devices, poorly designed interfaces have contributed to serious accidents with human consequences.

Usability is one of the central concepts in HCI. Jakob Nielsen, a leading figure in the discipline, defined it as the degree to which a

product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. Nielsen identified ten usability heuristics that remain widely referenced by practitioners: visibility of system status (the system should always keep users informed about what is happening through timely feedback); error prevention (the design should minimise the opportunity for users to make errors rather than relying solely on error messages); recognition rather than recall (users should not be required to remember information from one part of the interface to use another); flexibility and efficiency of use (shortcuts should be available for experienced users without impeding novices); and aesthetic and minimalist design (interfaces should not contain irrelevant information, since each additional element competes for the user's attention).

Cognitive load refers to the mental effort required to use a system. Interfaces that present too many simultaneous choices, use unfamiliar or inconsistent terminology, or lack clear visual hierarchy impose unnecessary cognitive load and make systems harder to learn and operate efficiently. Effective interface design reduces unnecessary cognitive load by grouping related elements, applying consistent visual conventions, prioritising the most common actions, and providing clear, immediate feedback for user actions. The concept of affordance – derived from the work of psychologist James

Gibson and introduced to design by Donald Norman – holds that the physical and visual properties of interface elements should communicate how they are to be used: a button should look pressable, a text field should appear editable, a scrollbar should suggest that content extends beyond the visible area.

Information architecture (IA) is the practice of organising, structuring, and labelling content to support effective navigation and task completion. In the context of websites and applications, good IA means designing navigation menus, content hierarchies, search functions, and labelling systems that align with the mental models and goals of users. A mismatch between the system's organisation and the user's mental model – sometimes called the 'designer's curse', where familiarity with the system makes its structure seem obvious – is one of the most common causes of usability problems. Card sorting is a standard IA research technique in which participants organise content labels into groups that make sense to them, revealing their mental categorisation of the content.

Accessibility is the practice of designing products so that they can be used by people with the widest possible range of abilities, including those with visual, auditory, motor, and cognitive impairments. The Web Content Accessibility Guidelines (WCAG) provide an internationally recognised framework, covering requirements such as providing text alternatives for all non-text

content, ensuring sufficient contrast between foreground and background colours, making all interactive functionality operable through a keyboard (not just a mouse), and providing captions for audio and video content. Accessibility improvements frequently benefit all users: subtitles help users in noisy environments; clear navigation helps users with limited time or attention; and high colour contrast improves legibility on mobile screens in bright sunlight.

3. Give the Ukrainian equivalents to the following words and word combinations:

high-stakes environment; interface failure; to keep users informed; timely feedback; to prevent errors; to remember information; to impede novices; irrelevant information; unfamiliar terminology; visual hierarchy; to impose cognitive load; visual convention; to prioritise actions; interface element; mental model; designer's curse; card sorting; text alternative; colour contrast; keyboard operability.

4. Fill in the gaps with the words from the list:

usability, wireframe, prototype, user research, A/B testing, accessibility, affordance, cognitive load, responsive design, persona

1. _____ is the degree to which a product allows users to achieve their goals effectively and efficiently.
2. A _____ is a low-fidelity structural diagram showing content placement and layout without visual design details.
3. A _____ is an interactive model that simulates the product experience and can be tested with real users.
4. A user _____ is a composite profile built from research data representing a key type of user.
5. _____ involves gathering data about users' needs, behaviours, and goals to inform design decisions.
6. _____ is the mental effort required to process and understand information while using an interface.
7. The _____ of a button is the visual property that communicates it should be pressed.
8. _____ ensures an interface works correctly across screens of varying sizes and input methods.
9. _____ compares two design variants with real users to determine which performs better on a given metric.
10. _____ design ensures that products can be used by people with a wide range of physical and cognitive abilities.

5. Match the terms with their definitions:

#	Term		Definition
1	usability heuristic	a	the practice of organising and labelling content to support navigation
2	information architecture	b	a guideline for evaluating interface design quality

#	Term		Definition
3	cognitive load	c	observing users in their natural working environment
4	contextual inquiry	d	the mental effort required to use a system
5	WCAG	e	standards for making web content accessible to people with disabilities
6	card sorting	f	a technique to understand how users mentally categorise content

6. Insert the correct preposition (in, for, with, on, of, to, by, from, between, about):

1. HCI is concerned _____ the design and evaluation of interactive computing systems.
2. Usability is defined as the degree _____ which a product can be used effectively.
3. The affordance _____ a button communicates that it should be pressed.
4. Information architecture helps users navigate _____ large amounts of structured content.
5. Responsive design ensures that interfaces adapt _____ screens of different sizes.
6. WCAG provides standards _____ making web content accessible _____ people with disabilities.

7. Read and translate the text.

The UX Design Process and Emerging Interaction Paradigms

User experience (UX) design is the process of creating products that provide meaningful, efficient, and satisfying experiences. It extends far beyond the visual layer to encompass the entire journey a user has with a product – from first encounter through long-term mastery. Good UX design is rooted in empathy: understanding users' goals, mental models, frustrations, and contexts deeply enough to make design decisions that serve them rather than the product team's assumptions. The UX design process is iterative: research informs design, design is tested with real users, insights from testing drive refinement, and the cycle continues throughout the product's development and operation.

User research is the foundation of effective UX. Qualitative methods include in-depth interviews that explore users' motivations, values, and mental models; contextual inquiry, where researchers observe users working in their natural environment to reveal the gap between what people say they do and what they actually do; and think-aloud protocols, where participants narrate their thoughts and reactions as they complete tasks with a prototype or existing product. Quantitative methods include surveys for measuring attitudes across large samples, analytics data to reveal actual usage patterns, and eye-tracking studies to map where users focus their visual attention. User

personas synthesise research findings into composite profiles representing key user groups, helping design teams maintain focus on real user needs rather than hypothetical ones.

The design phase progresses through increasing levels of fidelity. Wireframes are low-fidelity structural diagrams that define content placement and interaction flow without committing to visual design decisions. Mockups add colour, typography, and imagery, giving stakeholders a realistic visual impression of the intended product. Interactive prototypes allow users to click, tap, and navigate through a simulated product before development begins, enabling early discovery of usability problems when changes are inexpensive. The principle of 'fail early, fail cheap' underlies iterative design: identifying fundamental problems in a ten-screen prototype costs far less than discovering them after months of engineering work. Usability testing – observing real users attempting to complete realistic tasks – is more effective at revealing genuine usability problems than expert heuristic evaluation alone.

Emerging interaction paradigms are expanding the range of ways in which people interact with digital systems. Voice interfaces, driven by advances in natural language processing, allow hands-free interaction through speech – particularly valuable in contexts where visual attention or manual dexterity is occupied. Touchscreens enable direct, intuitive manipulation of digital content through gestures such

as tap, swipe, pinch, and zoom. Augmented reality (AR) overlays contextually relevant digital information onto the user's view of the real world through a smartphone camera or specialised glasses, with applications in navigation, retail, maintenance, surgery, and education. Virtual reality (VR) places users inside fully immersive simulated environments, used for professional training simulations, architectural walkthroughs, and therapeutic applications including exposure therapy for phobias.

As AI systems become more sophisticated, new forms of adaptive and personalised interfaces are emerging. Systems can now observe user behaviour and preferences over time and adjust their content, recommendations, layout, and even functionality to better suit individual users. This personalisation can significantly improve usability and efficiency – but it also raises important ethical concerns. Users should be able to understand when and how a system is adapting to them, inspect the preferences and profile the system has built, and adjust or override its decisions. Transparency, user control, and the prevention of harmful filter bubbles are core principles for the responsible design of AI-driven interfaces. The future of HCI will be shaped by the challenge of making intelligent, adaptive systems genuinely usable, trustworthy, and respectful of user autonomy.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. UX design is primarily about choosing attractive colour schemes and typography.
2. Contextual inquiry involves observing users working in their natural environment.
3. Wireframes are high-fidelity designs that show the final visual appearance of a product.
4. Interactive prototypes allow usability problems to be identified before development begins.
5. Augmented reality completely replaces the user's view of the real world with a virtual environment.
6. AI-driven personalised interfaces should be transparent and give users control over adaptations.

9. Choose the most suitable answer:

1. The principle of 'recognition rather than recall' means ____.
a) the system should prompt users with available options rather than expecting them to remember commands b) the system should automatically complete all user tasks without input c) users should memorise the complete set of keyboard shortcuts d) error messages should be written in technical language for precision
2. Card sorting is used primarily to ____.
a) evaluate the loading speed of a website b) compare two different visual designs c) understand how users naturally categorise and group content d) measure the contrast ratio of text elements

3. A/B testing is most useful for ____.
a) checking whether an application meets security standards
b) determining which of two design variations produces better user outcomes
c) verifying that the application works on all browsers
d) testing the scalability of the backend infrastructure
4. Good information architecture reduces usability problems by ____.
a) increasing the number of navigation options available
b) aligning the system's organisation with users' mental models
c) reducing the number of pages in the application
d) ensuring all pages load within two seconds

10. Give the English equivalents to the following words and word combinations:

глибинне інтерв'ю з користувачем; ментальна модель; метод «думай вголос»; розрив між словами та діями; хмарна аналітика даних; дослідження за допомогою відстеження погляду; профіль користувача; каркасний макет; адаптивний прототип; тестування зручності використання; принцип ранньої відмови; рівень деталізації; голосовий інтерфейс; доповнена реальність; іммерсивне середовище; пряме маніпулювання; персоналізований інтерфейс; фільтрувальний міхур; автономія користувача; відповідальний дизайн.

11. Arrange the stages of the UX design process in a logical order (1 = first):

- a) Test the prototype with real users and collect feedback.
- b) Refine the design based on testing insights; repeat the cycle.
- c) Conduct user research: interviews, contextual inquiry, surveys.
- d) Build wireframes and define the information architecture.
- e) Create an interactive prototype and add visual design detail.

12. Answer the questions and discuss in a group:

- What is the difference between UX design and UI design?
- Why do researchers say there is often a gap between what users say they do and what they actually do?
- What is the difference between a wireframe, a mockup, and an interactive prototype?
- What does 'fail early, fail cheap' mean in the context of UX design?
- What are the main emerging interaction paradigms and what use cases is each suited to?
- What ethical responsibilities do designers have when building AI-driven personalised interfaces?
- Think of a digital product you use regularly. Which of Nielsen's heuristics does it follow well, and which does it violate?

LESSON 11

COMPUTER GRAPHICS, MULTIMEDIA, AND SIGNAL PROCESSING

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
raster graphics	растрова графіка
vector graphics	векторна графіка
pixel / resolution	піксель / роздільна здатність
rendering	рендеринг / візуалізація
3D modelling	тривимірне моделювання
texture mapping	текстурування (накладання текстур)
shading / lighting	затінення / освітлення
compression: lossy / lossless	стиснення: з втратами / без втрат
codec	кодек
frame rate (fps)	частота кадрів (кадрів на секунду)
digital signal processing (DSP)	цифрова обробка сигналів
sampling rate	частота дискретизації

English term / word combination	Ukrainian equivalent
Fourier transform	перетворення Фур'є
augmented reality (AR)	доповнена реальність
virtual reality (VR)	віртуальна реальність

2. Read and translate the text.

Computer Graphics: Raster, Vector, and Three-Dimensional Rendering

Computer graphics is the discipline concerned with generating, manipulating, and displaying visual content using computers. It underpins a vast range of applications: from the text and images rendered in a word processor or web browser to the photorealistic imagery produced for feature films, the real-time three-dimensional worlds of video games, scientific visualisation of complex datasets, and medical imaging systems that help clinicians diagnose disease. Graphics processing is computationally intensive: contemporary video games render tens of millions of polygons per frame at sixty or more frames per second, a workload that has driven the development of highly parallel graphics processing units (GPUs) with thousands of processing cores.

Digital images are represented in two fundamentally different ways. Raster (or bitmap) graphics represent an image as a rectangular grid of pixels, each storing a colour value typically encoded as red, green, and blue (RGB) components. The resolution of a raster image – usually expressed as width \times height in pixels or as pixels per inch (PPI) – determines how detailed it appears at a given display size. Common raster formats include JPEG, PNG, and TIFF. A key limitation of raster images is that they scale poorly: enlarging them beyond their native resolution reveals individual pixels as visible squares, producing pixelation. Common raster file formats handle this constraint differently: PNG uses lossless compression and is preferred for graphics with sharp edges and flat areas of colour; JPEG uses lossy compression and is preferred for photographs where some quality loss is acceptable in exchange for much smaller file sizes.

Vector graphics represent images not as pixel grids but as mathematical descriptions of geometric shapes – lines, curves, and filled regions defined by coordinates, bezier curves, and equations. Because vector images are defined mathematically rather than by a fixed grid, they can be rendered at any size without quality loss, making them ideal for logos, icons, typography, and illustrations. SVG (Scalable Vector Graphics) is the standard format for vector graphics on the web. Applications such as Adobe Illustrator and

Inkscape are designed specifically for vector image creation and editing.

Three-dimensional computer graphics involves creating virtual representations of objects in 3D space and rendering them as 2D images from a chosen viewpoint. A 3D model is typically a polygon mesh – a collection of vertices, edges, and polygonal faces that define the surface geometry of an object. The surface of a 3D model is given visual detail through texture mapping: a 2D image (texture) is projected onto the mesh surface to add colour, surface imperfections, and fine detail without increasing the geometric complexity of the model. Shading and lighting algorithms determine how simulated light sources interact with surfaces to produce highlights, shadows, and the visual cues from which viewers perceive depth and material properties.

Image compression reduces the storage space and transmission bandwidth required for digital images. Lossless compression algorithms, such as those used in PNG and GIF formats, reconstruct the original image data exactly from the compressed form – every pixel is preserved perfectly. Lossy compression, exemplified by JPEG, discards data that is judged least perceptible to the human visual system, achieving considerably higher compression ratios at the cost of some irreversible quality loss. Video compression standards such as H.264 and H.265 (HEVC) exploit both spatial

redundancy within each frame and temporal redundancy across successive frames – encoding only the differences between frames rather than each complete frame – to achieve very high compression ratios that make streaming high-definition video practical over typical internet connections.

3. Give the Ukrainian equivalents to the following words and word combinations:

to generate visual content; polygon mesh; pixel grid; colour value; visible pixelation; mathematical description; bezier curve; quality loss; surface geometry; to project a texture; material property; lossless compression; to reconstruct exactly; irreversible quality loss; compression ratio; spatial redundancy; temporal redundancy; successive frames; streaming video; typical internet connection.

4. Fill in the gaps with the words from the list:

raster, vector, resolution, rendering, texture mapping, shading, frame rate, codec, compression, sampling rate

1. _____ graphics represent images as grids of pixels with a fixed number of colour values.
2. _____ graphics use mathematical descriptions of shapes and can scale to any size without quality loss.

3. The _____ of a raster image determines how much detail it contains per unit of display area.
4. _____ is the process of generating a 2D image from a 3D scene description.
5. _____ applies 2D images to the surfaces of 3D models to add visual detail and colour.
6. _____ algorithms simulate how light interacts with surfaces to produce highlights and shadows.
7. _____ is the number of individual frames displayed or rendered per second.
8. A _____ is a software component that encodes and decodes digital audio or video streams.
9. _____ reduces the storage and bandwidth requirements of digital images and video.
10. In digital audio, the _____ is how many times per second the audio waveform amplitude is measured.

5. Match the graphics and compression formats with their descriptions:

#	Term		Definition
1	JPEG	a	a lossless format commonly used for graphics with sharp edges and transparency
2	PNG	b	the standard scalable vector format for the web
3	SVG	c	a lossy compressed format widely used for digital photographs

#	Term		Definition
4	H.264	d	a compression standard that encodes differences between video frames
5	GIF	e	a lossless format supporting animation, limited to 256 colours
6	TIFF	f	a high-quality lossless format used in professional printing and archiving

6. Combine the sentences using a relative clause (who / which / that / where / whose):

1. JPEG is a lossy compression format. It is widely used for digital photographs.
2. A GPU is a specialised processor. It handles large numbers of parallel calculations.
3. Texture mapping is a technique. It applies 2D images to 3D model surfaces.
4. A polygon mesh consists of vertices and faces. They define the surface geometry of a 3D object.
5. Physically-based rendering is a method. Its goal is to simulate real-world light behaviour accurately.
6. A codec is a software component. It encodes and decodes audio or video data.

7. Read and translate the text.

Digital Signal Processing, Audio, and Immersive Technologies

Digital signal processing (DSP) is the mathematical manipulation of digital signals – sequences of numbers representing sampled measurements of physical quantities such as sound, light, temperature, or radio waves – in order to filter, transform, analyse, or extract information from them. DSP is fundamental to an enormous range of technologies that are used daily: audio processing (equalisation, compression, noise cancellation), telecommunications (radio, mobile phones, satellite communications), medical imaging (MRI, ultrasound, CT reconstruction), radar and sonar systems, speech recognition, and image enhancement in cameras and displays.

Sound is a continuous (analogue) signal – a variation in air pressure over time. To store, transmit, or process it digitally, it must be converted to a numerical representation through a process called analogue-to-digital conversion (ADC). Sampling captures the amplitude of the audio waveform at regular time intervals; the number of samples taken per second is the sampling rate. The Nyquist–Shannon sampling theorem states that, to accurately reconstruct a signal, the sampling rate must be at least twice the highest frequency present in the signal. Standard audio CDs use a sampling rate of 44,100 Hz, comfortably exceeding the approximately 20,000 Hz upper limit of human hearing. Bit depth – the number of bits used to encode each sample – determines the dynamic range: the ratio between the quietest and loudest representable sounds. CD audio uses

16-bit samples, yielding a theoretical dynamic range of 96 dB, sufficient for most listening contexts.

The Fourier transform is a mathematical tool that decomposes a signal into its constituent sinusoidal frequency components, revealing which frequencies are present and at what amplitudes. In signal processing, understanding a signal's frequency content is essential for designing effective filters: a low-pass filter attenuates frequencies above a threshold while passing lower frequencies, used for example in audio equalisation to reduce high-pitched noise; a high-pass filter does the reverse; a band-pass filter passes only a specific range of frequencies. The Fast Fourier Transform (FFT) is a computationally efficient algorithm for computing the Discrete Fourier Transform, and is one of the most important algorithms in computational science – used in audio processing, image compression, wireless communications, and scientific computing.

Augmented reality (AR) integrates computer-generated content into the user's view of the real world in real time, typically through a smartphone camera or transparent wearable display. AR applications overlay contextual information on physical objects: a navigation app can display turn-by-turn directions on a live street view; a maintenance technician can see step-by-step instructions and sensor readings overlaid on a piece of machinery; a surgeon can view real-time imaging data positioned anatomically on a patient. The

precision of AR registration – the alignment of virtual content with the real world – depends on accurate, low-latency tracking of the device's position and orientation. Virtual reality (VR) replaces the real world with an immersive simulated environment presented through a head-mounted display, surrounding the user with stereoscopic imagery and spatial audio. Applications include professional training simulations for surgeons, pilots, and military personnel; architectural and engineering visualisations; and therapeutic use in the treatment of post-traumatic stress and phobias.

Producing high-quality AR and VR experiences imposes demanding technical requirements. For VR, frame rates below approximately 90 fps are associated with motion sickness – a form of disorientation caused by a mismatch between visual motion and the vestibular system's perception of physical movement. End-to-end latency (the delay from physical head movement to corresponding display update) must be kept below 20 milliseconds to avoid a perceptible lag. Spatial audio processing computes how sounds travel through a simulated space to the user's ears, accounting for distance, reflections, and the directional filtering of the outer ear (HRTF – Head-Related Transfer Function), creating the impression that sounds come from specific locations in three-dimensional space and greatly enhancing the sense of presence. Achieving all these requirements

simultaneously while rendering complex 3D environments is a significant engineering challenge.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. The Nyquist theorem states that the sampling rate must be at least twice the highest frequency in the signal.
2. CD audio uses a sampling rate of 44,100 Hz, which is sufficient to represent all frequencies audible to humans.
3. The Fourier transform decomposes a signal into its constituent frequency components.
4. Augmented reality completely replaces the user's view of the real world with a simulated environment.
5. VR frame rates should ideally exceed 90 fps to avoid motion sickness.
6. Spatial audio processing creates the impression that sounds originate from specific locations in 3D space.

9. Choose the most suitable answer:

1. Lossless compression differs from lossy compression in that _____.
a) it achieves higher compression ratios at the cost of quality
b) the original data can be reconstructed exactly from the compressed form
c) it is only suitable for video, not still images
d) it requires specialised hardware to decode
2. The Fourier transform is important in signal processing because it _____.
a) compresses audio files without any

- quality loss b) converts analogue signals to digital form
 c) reveals which frequency components are present in a signal d) increases the sampling rate of a digital recording
3. Vector graphics are preferred over raster graphics for logos because _____. a) they have a much larger file size b) they can be scaled to any size without visible quality loss c) they support a wider range of colours d) they load faster in web browsers
 4. Motion sickness in VR is primarily caused by _____. a) excessive brightness of the display b) frame rates that are too high c) a mismatch between visual motion and the body's physical perception of movement d) insufficient resolution of the head-mounted display

10. Give the English equivalents to the following words and word combinations:

цифрова обробка сигналів; аналоговий сигнал; аналого-цифрове перетворення; частота дискретизації; теорема Найквіста–Шеннона; бітова глибина; динамічний діапазон; перетворення Фур'є; частотна складова; фільтр нижніх частот; алгоритм FFT; скасування шуму; доповнена реальність; відстеження пристрою; наголовний дисплей; просторовий звук; передаточна функція голови; час наскрізної затримки; вестибулярна система; рівень занурення.

11. Arrange the steps of the 3D real-time rendering pipeline in the correct order (1 = first):

- a) Fragment processing: computing the final colour of each pixel based on lighting, textures, and special effects.
- b) Rasterisation: converting the projected 2D geometry into a grid of screen pixels.
- c) Output: displaying the completed 2D image on screen.
- d) Scene setup: placing 3D objects, light sources, and a virtual camera in the scene.
- e) Geometry processing: applying transformations and projecting 3D vertex coordinates onto the 2D screen.

12. Answer the questions and discuss in a group:

- What is the difference between raster and vector graphics and when is each more appropriate?
- What is the difference between lossy and lossless compression and in what situations is each preferred?
- Why must the sampling rate be at least twice the highest frequency in the signal being recorded?
- What is the Fourier transform used for in audio processing?
- What are the main technical challenges in producing high-quality VR experiences?
- What is the difference between augmented reality and virtual reality? Give a real-world application for each.
- Can you name three everyday technologies that rely on digital signal processing?

LESSON 12

ETHICS, SOCIETY, AND THE FUTURE OF COMPUTING

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
digital ethics	цифрова етика
privacy	конфіденційність / право на приватність
data protection	захист персональних даних
surveillance	стеження / наглядова діяльність
algorithmic bias	алгоритмічна упередженість
digital divide	цифровий розрив
intellectual property (IP)	інтелектуальна власність
copyright / open source	авторське право / відкрите ПЗ
misinformation / disinformation	дезінформація / навмисна дезінформація
automation and job displacement	автоматизація та зміщення робочих місць
sustainable computing	екологічно відповідальні обчислення

English term / word combination	Ukrainian equivalent
digital wellbeing	цифровий добробут
AI governance	управління штучним інтелектом
responsible innovation	відповідальні інновації
informed consent	усвідомлена згода

2. Read and translate the text.

Digital Ethics: Privacy, Surveillance, and Algorithmic Fairness

Computing technology is not ethically neutral. Every system that software engineers design and deploy embodies choices about who it serves, whose interests it prioritises, what data it collects, and what risks it creates or redistributes. Digital ethics is the study of the moral questions raised by digital technologies and the obligations of those who design, deploy, and regulate them. For computing professionals, ethical awareness is not a supplementary concern – it is a core professional responsibility, as recognised by codes of ethics from bodies such as the Association for Computing Machinery (ACM) and the IEEE.

Privacy is the right of individuals to control information about themselves – what is collected, how it is used, with whom it is shared, and for how long it is retained. Digital technologies have dramatically

expanded the capacity to collect, aggregate, and analyse personal data. Social media platforms, search engines, mobile applications, smart speakers, fitness trackers, and payment systems continuously gather fine-grained data about users' behaviour, preferences, locations, health, and social relationships. Much of this collection happens in ways that users neither fully understand nor genuinely consent to, despite lengthy terms-of-service documents that nominally request their agreement. The GDPR in Europe, the CCPA in California, and similar legislation in other jurisdictions attempt to restore meaningful control to individuals by establishing rights to access, correct, delete, and port personal data, and by requiring genuine informed consent for data collection beyond strict necessity.

Surveillance – the systematic, large-scale observation and monitoring of individuals – has been transformed by digital technology. CCTV networks enhanced with real-time facial recognition can track individuals across entire cities without their knowledge or consent. Mobile phones continuously broadcast location and network identity data that can reveal detailed information about an individual's movements, associations, and daily routines. Social media monitoring tools enable automated tracking of expressed opinions and social connections at scale. Disclosure of mass surveillance programmes by intelligence agencies has revealed that enormous volumes of communications metadata – and in some

cases content – are collected and retained by governments, raising profound questions about the compatibility of such practices with democratic values, rule of law, and the right to free expression.

Algorithmic bias occurs when automated decision-making systems produce systematically unfair or discriminatory outcomes for certain groups of people. This typically arises because the data used to train a model reflects historical patterns of discrimination – for example, biased hiring records, racially unequal criminal sentencing, or gender disparities in medical treatment – and the model learns to reproduce those patterns at scale. Studies have shown that facial recognition systems have significantly higher error rates for women and for darker-skinned individuals than for white men. AI systems used in high-stakes decisions – loan approvals, medical diagnoses, job screening, parole decisions, content moderation – can perpetuate or amplify existing social inequalities if they are not carefully designed, audited, and subject to meaningful human oversight. Fairness, accountability, and transparency are essential properties of any AI system that affects significant decisions in people's lives.

The digital divide refers to inequalities in access to digital technologies and the skills to use them effectively between different groups – whether defined by income, geography, age, disability, or education level. While global internet penetration has grown substantially, significant disparities persist: in many parts of the

world, reliable broadband access remains unaffordable or technically unavailable; older adults and people with lower educational attainment are less likely to have the digital literacy skills needed to navigate increasingly digitised public services; and people with disabilities may encounter inaccessible systems and interfaces. As more aspects of economic participation, healthcare, education, and civic engagement are mediated through digital systems, exclusion from those systems increasingly means exclusion from society more broadly.

3. Give the Ukrainian equivalents to the following words and word combinations:

to embody choices; to prioritise interests; professional responsibility; code of ethics; to aggregate data; genuine informed consent; terms-of-service document; to restore meaningful control; systematic observation; to reveal detailed information; mass surveillance; communications metadata; historical discrimination; racially unequal sentencing; high-stakes decision; human oversight; to amplify inequality; internet penetration; digital literacy; civic engagement.

4. Fill in the gaps with the words from the list:

privacy, surveillance, algorithmic bias, digital divide, GDPR, misinformation, copyright, automation, digital wellbeing, AI governance

1. _____ is the right of individuals to control information about themselves.
2. The _____ requires organisations in Europe to obtain genuine consent before collecting personal data.
3. Digital _____ refers to the large-scale observation and monitoring of individuals using technology.
4. _____ occurs when automated systems produce systematically unfair outcomes for certain groups.
5. The _____ refers to inequalities in access to digital technologies between different populations.
6. _____ is the spread of false or inaccurate information, whether intentionally or not.
7. _____ protects creators' legal rights over their original works.
8. _____ replaces human labour with machines or algorithms, which may displace workers in certain sectors.
9. _____ refers to the development of policies and frameworks to ensure AI is deployed safely and fairly.
10. _____ concerns how people's use of technology affects their mental health and quality of life.

5. Match the terms with their definitions:

#	Term		Definition
1	informed consent	a	the spread of intentionally false information to deceive
2	disinformation	b	policies and frameworks for responsible AI development and deployment
3	AI governance	c	meaningful agreement based on genuine understanding of what is being agreed to
4	digital divide	d	inequalities in access to and ability to use digital technologies
5	algorithmic fairness	e	the protection of creators' rights over their original works
6	copyright	f	the principle that automated systems should not produce discriminatory outcomes

6. Complete the sentences with a suitable modal verb (must, should, can, may, might, need not):

1. Companies _____ obtain genuine informed consent before collecting personal data from users.
2. AI systems used in hiring _____ be regularly audited to detect and correct bias.
3. A developer _____ redistribute an open-source library as long as the licence conditions are met.
4. Automated systems _____ perpetuate historical discrimination if trained on biased historical data.

5. Citizens _____ be excluded from vital public services if the digital divide is not addressed.
6. Organisations _____ collect more personal data than is strictly necessary for their stated purpose.

7. Read and translate the text.

Automation, Sustainability, Digital Wellbeing, and the Future of the Profession

Automation has always reshaped the nature of work, and computing technology is accelerating this transformation at an unprecedented pace. Routine cognitive tasks that were once the exclusive domain of white-collar workers – data entry, document processing, basic analysis, structured customer interactions – are increasingly automated by software systems. More recently, advances in generative AI have begun to affect tasks that many assumed were uniquely human: writing, translation, legal research, medical diagnosis, financial analysis, and software development itself. The net economic effect of this wave of automation is contested: optimists point to historical precedent showing that technological progress ultimately creates more jobs than it displaces, while pessimists argue that the current pace and breadth of AI-driven automation is qualitatively different from previous technological transitions.

Regardless of one's view on the aggregate employment effects, it is clear that automation will substantially change the composition of skills that are economically valuable. Tasks characterised by routine and predictability are most vulnerable; tasks requiring creativity, complex judgement, emotional intelligence, ethical reasoning, and adaptive problem-solving in novel situations are most resilient. For computing professionals, this suggests that technical competence – though necessary – will not be sufficient in isolation. The ability to communicate technical ideas clearly to non-technical audiences, to collaborate in multidisciplinary teams, to navigate ethical trade-offs, and to learn continuously as technologies evolve will be at least as important as programming skill. The concept of lifelong learning has shifted from a professional aspiration to a practical necessity.

Sustainable computing addresses the environmental costs of the digital economy. Data centres already consume approximately 200–250 terawatt-hours of electricity annually – roughly 1–2% of global electricity consumption – and this figure is growing as demand for computation increases, driven by cloud services, video streaming, and AI training. Training large-scale AI models is particularly energy-intensive: researchers have estimated that training a single large language model can produce carbon emissions comparable to the lifetime emissions of several average cars. The manufacture of

computing devices consumes significant quantities of rare earth metals and other minerals, many of which are extracted under environmentally damaging or ethically problematic conditions. Electronic waste (e-waste) is the fastest-growing category of solid waste globally, with only a small fraction formally recycled. Computing professionals and organisations increasingly recognise a responsibility to consider energy efficiency, hardware longevity, and end-of-life disposal as integral parts of system design.

Digital wellbeing concerns the effects of technology use on mental health, relationships, cognitive development, and social life. Research has established associations between heavy social media use and elevated rates of anxiety, depression, and loneliness, particularly among adolescents. Persuasive design patterns – infinite scroll, variable-reward notification systems, algorithmically curated feeds optimised for maximum engagement – are deliberately engineered to capture and hold attention, often in ways that serve the platform's commercial interests at the expense of users' wellbeing and autonomy. Legislators and regulators in multiple jurisdictions have introduced or are considering measures to restrict these design practices, particularly in products aimed at children and young people. The concept of 'time well spent' – measuring the value of technology use by the quality of experience it enables rather than the

volume of time consumed – is gaining traction among designers and policymakers.

Looking ahead, computing professionals will work in a landscape shaped by several converging technological trajectories: increasingly capable AI systems integrated into professional and everyday life; quantum computing requiring a fundamental rethinking of cryptographic infrastructure; the proliferation of connected devices generating unprecedented data volumes; and augmented and virtual reality expanding the boundary between physical and digital experience. Navigating this landscape successfully will require professionals who understand not only how these technologies work but also the societal systems into which they are deployed – the economic incentives, regulatory frameworks, cultural values, and human needs that shape how technologies are adopted, adapted, and contested. Responsible innovation means designing with foresight: anticipating consequences, engaging affected stakeholders, and building in safeguards before problems emerge rather than responding to harm after the fact.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. Automation only affects manual labour and has no impact on knowledge-intensive white-collar tasks.

2. Tasks requiring creativity, ethical reasoning, and adaptive judgement are among the most resilient to automation.
3. Data centres currently consume approximately 50% of global electricity.
4. Training large AI models can produce carbon emissions comparable to the lifetime emissions of several cars.
5. Persuasive design patterns like infinite scroll are intended to maximise platform engagement time.
6. Responsible innovation involves anticipating consequences and building in safeguards before harm occurs.

9. Choose the most suitable answer:

1. Which types of tasks are most vulnerable to automation by AI systems? a) Tasks requiring novel creative judgement and ethical reasoning b) Tasks involving emotional intelligence and interpersonal relationships c) Routine, predictable, and well-defined cognitive tasks d) Tasks requiring adaptation to rapidly changing circumstances
2. The concept of 'time well spent' in digital wellbeing refers to _____. a) maximising the total number of hours users spend on a platform b) measuring technology use by the quality of experience it enables c) encouraging users to spend money on premium features d) reducing screen time to zero for all age groups
3. Responsible innovation in computing means _____. a) developing technology as fast as possible to maximise commercial advantage b) leaving ethical and societal analysis to regulators after deployment c) anticipating consequences, engaging stakeholders, and designing in safeguards proactively d) publishing all source code under an open-source licence

4. E-waste is considered a growing problem primarily because ____.
- a) it contains valuable components that cannot be recycled
 - b) it is the fastest-growing category of solid waste and only a small fraction is formally recycled
 - c) it is entirely composed of hazardous materials that cannot be safely disposed of
 - d) it is generated only in developed countries with high computing consumption

10. Give the English equivalents to the following words and word combinations:

рутинні когнітивні завдання; обробка документів; безпрецедентний темп; сукупний вплив на зайнятість; адаптивне вирішення проблем; безперервне навчання; стала обчислювальна техніка; щорічне споживання електроенергії; вуглецеві викиди; рідкісноземельні метали; електронні відходи; офіційна переробка; цифровий добробут; нескінченне прокручування; система сповіщень зі змінною винагородою; алгоритмічно підібрана стрічка; комерційні інтереси; цінність часу в цифровому середовищі; відповідальні інновації; попередження наслідків.

11. Read the following statements and decide whether you agree or disagree. Justify your position:

1. "Social media platforms have a moral obligation to prioritise users' wellbeing over engagement metrics."

2. "Governments should regulate AI systems used in hiring and criminal justice as strictly as they regulate pharmaceuticals."
3. "The environmental cost of computing is a shared responsibility of users, companies, and governments alike."
4. "The digital divide is ultimately an economic problem, not a technological one."

12. Answer the questions and discuss in a group:

- Which categories of work are most and least vulnerable to automation by AI systems?
- Why will broad human skills such as communication and ethical reasoning remain valuable in an automated economy?
- What is the environmental footprint of modern computing infrastructure and what can be done to reduce it?
- What is e-waste and why is it considered a significant environmental and social problem?
- What persuasive design patterns do social media platforms use and what concerns do they raise?
- What does 'responsible innovation' mean for a software developer in their day-to-day work?
- In your opinion, what is the single most important ethical challenge facing computing professionals in the next decade? Justify your answer.

LESSON 13

EMBEDDED SYSTEMS AND THE INTERNET OF THINGS

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
embedded system	вбудована система
microcontroller	мікроконтролер
real-time operating system (RTOS)	операційна система реального часу
firmware	мікропрограма / прошивка
sensor / actuator	датчик / виконавчий пристрій
Internet of Things (IoT)	Інтернет речей
smart device	розумний пристрій
wireless protocol (Zigbee, Z-Wave, MQTT)	бездротовий протокол
edge computing	крайові обчислення
power consumption	енергоспоживання
interrupt	переривання
watchdog timer	сторожовий таймер
over-the-air update (OTA)	оновлення прошивки через ефір

English term / word combination	Ukrainian equivalent
interoperability	сумісність / взаємодія пристроїв
digital twin	цифровий двійник

2. Read and translate the text.

Embedded Systems: Architecture and Constraints

An embedded system is a computing system designed to perform a dedicated function within a larger device, as opposed to a general-purpose computer that can run any software. Embedded systems are found in virtually every category of manufactured product: smartphones, automobiles, industrial machinery, medical devices, household appliances, aviation systems, and consumer electronics all rely on embedded controllers to manage their operation. The global market for embedded systems is enormous – estimates suggest that over 98% of all microprocessors produced are used in embedded applications, with only a small fraction ending up in general-purpose computers.

The hardware at the heart of most embedded systems is the microcontroller (MCU) – a single integrated circuit combining a CPU, memory (both flash for program storage and RAM for working data), and a range of peripheral interfaces such as timers, analogue-

to-digital converters, serial communication ports, and general-purpose input/output (GPIO) pins. Compared with desktop processors, microcontrollers are slower and have much less memory, but they are far smaller, cheaper, and more energy-efficient. A typical microcontroller might run at 16–200 MHz and have 256 KB of flash memory and 32 KB of RAM – sufficient for well-written, resource-conscious firmware.

Embedded software – known as firmware – differs from conventional application software in several important respects. It runs directly on hardware, often without an operating system. It must respond to physical events (sensor readings, incoming signals, timer expirations) within strict time constraints. Reliability is paramount: a failure in automotive brake software or a pacemaker controller could have fatal consequences. For systems with real-time requirements, a real-time operating system (RTOS) provides a scheduler that guarantees tasks will execute within defined time bounds, along with mechanisms for inter-task communication and synchronisation.

Interrupts are a fundamental mechanism in embedded programming. Rather than continuously polling peripherals to check whether an event has occurred – a wasteful approach – the processor can be instructed to suspend its current activity and immediately execute an interrupt service routine (ISR) when a hardware event occurs, such as the arrival of new sensor data or the expiration of a

timer. Efficient interrupt handling is essential for responsive, low-power embedded systems. Watchdog timers provide a safety net: if the main program fails to reset the watchdog within a specified interval – indicating that it has crashed or become stuck in an infinite loop – the watchdog automatically resets the system.

Power consumption is a critical design constraint for battery-powered embedded devices such as wearable health monitors, wireless sensor nodes, and remote environmental sensors. Designers use techniques such as putting the processor into sleep mode between tasks, reducing the clock speed to the minimum necessary, duty-cycling radio transmitters, and selecting low-power hardware components. A well-designed IoT sensor node might run for years on a pair of AA batteries, transmitting brief data packets at infrequent intervals.

3. Give the Ukrainian equivalents to the following words and word combinations:

dedicated function; integrated circuit; peripheral interface; analogue-to-digital converter; general-purpose input/output; resource-conscious firmware; physical event; time constraint; real-time requirement; task scheduler; inter-task communication; interrupt service routine; to poll peripherals; watchdog timer; infinite loop;

battery-powered device; sleep mode; duty-cycling; clock speed; data packet.

4. Fill in the gaps with the words from the list:

microcontroller, RTOS, firmware, sensor, actuator, interrupt, watchdog, OTA, edge computing, interoperability

1. A _____ is a single chip combining a CPU, memory, and peripheral interfaces for embedded applications.
2. A _____ guarantees that tasks execute within defined time bounds in real-time embedded systems.
3. The software stored permanently in an embedded device and controlling its basic operation is called _____.
4. A _____ measures a physical quantity and converts it into an electrical or digital signal.
5. An _____ is a mechanism that activates in response to a physical output command rather than measuring.
6. An _____ is a signal that causes the processor to suspend its current task and execute a handler routine.
7. A _____ timer resets the system automatically if the program fails to respond within a set interval.
8. _____ updates allow firmware to be upgraded remotely without physical access to the device.
9. _____ places processing resources close to IoT devices to reduce latency and bandwidth usage.
10. _____ is the ability of devices from different manufacturers to work together seamlessly.

5. Match the terms in column A with their definitions in column B:

#	Term		Definition
1	digital twin	a	a virtual replica of a physical object updated with real-time sensor data
2	MQTT	b	putting the processor into a low-power state between tasks
3	sleep mode	c	a lightweight messaging protocol designed for IoT devices
4	GPIO	d	configurable pins on a microcontroller for input and output
5	RTOS	e	an OS providing schedulers that guarantee task execution deadlines
6	firmware	f	software embedded permanently in hardware to control its basic functions

6. Rewrite the sentences in the Passive Voice:

1. Designers put the processor into sleep mode between tasks to save power.
2. The watchdog timer resets the system if the program fails to respond.
3. Engineers flash new firmware to the device using a JTAG interface.
4. The RTOS schedules all tasks according to their priority and deadlines.
5. Manufacturers embed microcontrollers in automobiles, appliances, and medical devices.

6. The sensor converts the physical measurement into a digital signal.

7. Read and translate the text.

The Internet of Things: Applications, Protocols, and Challenges

The Internet of Things (IoT) refers to the expanding network of physical objects embedded with sensors, software, and wireless connectivity that enables them to collect and exchange data. The term was coined by Kevin Ashton in 1999, but the concept has become a practical reality only in recent years, driven by dramatic reductions in the cost and size of sensors, wireless modules, and microcontrollers, combined with the ubiquitous availability of cloud computing for data storage and processing. The number of connected IoT devices is projected to exceed 30 billion by 2030.

IoT applications span an extraordinary range of domains. In smart homes, connected thermostats, lighting systems, security cameras, and voice assistants form an integrated ecosystem that can be monitored and controlled remotely. In industry, IoT sensors monitor equipment condition continuously, enabling predictive maintenance – identifying signs of wear or impending failure before a breakdown occurs, reducing costly downtime. In precision agriculture, soil sensors, drone imagery, weather stations, and satellite

data are integrated to optimise irrigation, fertilisation, and harvesting, reducing waste and environmental impact. Connected healthcare devices – wearable heart rate monitors, smart insulin pumps, implantable cardiac monitors – enable continuous remote patient monitoring.

The communication protocols used by IoT devices reflect the constraints of low-power, low-bandwidth hardware. MQTT (Message Queuing Telemetry Transport) is a lightweight publish-subscribe messaging protocol designed for constrained devices and unreliable networks, widely used in IoT applications. CoAP (Constrained Application Protocol) is a RESTful protocol optimised for machine-to-machine communication. Zigbee and Z-Wave are low-power mesh networking protocols used in smart home devices. LoRaWAN provides long-range, low-power wide-area connectivity suitable for sensors deployed over large geographical areas such as farmland or city infrastructure.

Security is among the most critical challenges facing IoT at scale. Many IoT devices are shipped with weak default credentials, run outdated software, and lack mechanisms for remote security updates. A 2016 Mirai botnet attack demonstrated the consequences: hundreds of thousands of compromised IoT devices were enlisted to launch a massive distributed denial-of-service attack that disrupted major internet services including Twitter, Netflix, and Spotify.

Securing IoT requires device authentication, encrypted communications, secure boot, regular over-the-air (OTA) firmware updates, and – increasingly – regulatory mandates requiring minimum security standards for connected devices.

A digital twin is a continuously updated virtual model of a physical asset, system, or process, fed by real-time sensor data from its physical counterpart. Digital twins allow engineers to monitor asset condition, simulate scenarios, predict failures, and optimise operations without risking the physical system. They are used in manufacturing (simulating production lines), energy (managing power grids), construction (monitoring structural health of buildings and bridges), and healthcare (modelling patient physiology). As IoT sensor coverage increases and cloud computing power grows, digital twins are becoming increasingly detailed and valuable.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. IoT devices are deployed only in consumer electronics and have no industrial applications.
2. MQTT is a lightweight messaging protocol designed for constrained devices and unreliable networks.
3. The Mirai botnet attack demonstrated that compromised IoT devices can be used to launch DDoS attacks.

4. Over-the-air updates allow firmware to be upgraded remotely without physical access to the device.
5. A digital twin is a physical backup copy of a device stored in a data centre.
6. LoRaWAN provides long-range, low-power connectivity suitable for geographically dispersed IoT sensors.

9. Give the English equivalents to the following words and word combinations:

вбудована система; мікроконтролер; операційна система реального часу; мікропрограма; датчик і виконавчий пристрій; переривання; сторожовий таймер; оновлення прошивки через ефір; Інтернет речей; розумний будинок; промислове технічне обслуговування за прогнозом; протокол зв'язку; обмежений пристрій; мережа з топологією «меш»; безпека IoT; шифрування зв'язку; безпечне завантаження; цифровий двійник; хмарна обробка даних; регуляторні вимоги.

10. Answer the questions:

- What distinguishes an embedded system from a general-purpose computer?
- What is a microcontroller and how does it differ from a desktop CPU?

- What is a real-time operating system and when is it necessary?
- Why are interrupts preferred over polling in embedded systems?
- What security challenges does widespread IoT adoption present?
- What is a digital twin and what advantages does it offer?

LESSON 14

BLOCKCHAIN AND DISTRIBUTED LEDGER TECHNOLOGY

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
blockchain	блокчейн
distributed ledger	розподілена книга обліку
block	блок (у блокчейні)
hash / cryptographic hash	хеш / криптографічний хеш
consensus mechanism	механізм консенсусу
Proof of Work (PoW)	доказ виконання роботи
Proof of Stake (PoS)	доказ частки
smart contract	розумний контракт
node	вузол (мережі)
wallet / private key	гаманець / приватний ключ
cryptocurrency	криптовалюта
decentralised application (DApp)	децентралізований застосунок
Non-Fungible Token (NFT)	невзаємозамінний токен
51% attack	атака 51%

English term / word combination	Ukrainian equivalent
tokenisation	токенізація

2. Read and translate the text.

How Blockchain Works: Structure and Consensus

Blockchain is a form of distributed ledger technology that stores records in a chain of cryptographically linked blocks, maintained simultaneously across a peer-to-peer network of computers rather than by a central authority. The core insight is simple yet profound: by combining cryptographic hashing, distributed storage, and a consensus mechanism, it is possible to create a record of transactions that is practically impossible to alter retroactively without the agreement of the majority of network participants – even though there is no trusted central party responsible for maintaining the record.

Each block in the chain contains a set of transaction records, a timestamp, a reference to its own content in the form of a cryptographic hash, and crucially, the hash of the immediately preceding block. A cryptographic hash function takes an input of any size and produces a fixed-length output (the hash) that changes completely and unpredictably if even a single bit of the input is

altered. This chaining means that modifying any block would change its hash, which would invalidate the next block's reference to it, requiring that block to be recomputed, which would invalidate the following block, and so on – making retroactive tampering computationally infeasible for any block that has been followed by a significant number of subsequent blocks.

For new blocks to be added to the chain, network participants must reach agreement through a consensus mechanism. Bitcoin uses Proof of Work (PoW): participants called miners compete to solve a computationally intensive puzzle that requires finding a number (nonce) such that the hash of the block header falls below a target value. This process requires enormous amounts of computation and therefore energy, making it costly to attack the network. The winning miner broadcasts the solved block, other nodes verify it, and the chain is extended. In return, the miner receives newly created cryptocurrency – the block reward – plus any transaction fees included in the block.

Proof of Stake (PoS), adopted by the Ethereum network in 2022 as a replacement for PoW, selects validators based on the quantity of cryptocurrency they have 'staked' – locked up as collateral – rather than on computational work. Validators are chosen pseudo-randomly in proportion to their stake, verify transactions, and propose new blocks. If they attempt to include fraudulent transactions, their

stake can be 'slashed' – destroyed as a penalty. PoS dramatically reduces the energy consumption of consensus compared with PoW while maintaining security guarantees, though it introduces different trade-offs regarding decentralisation and capital requirements.

The 51% attack represents the primary theoretical vulnerability of public blockchains: if a single entity controls more than half of the network's hashing power (in PoW) or staked capital (in PoS), they could potentially rewrite recent transaction history by producing an alternative chain longer than the honest chain. In practice, mounting such an attack against large, well-established blockchains such as Bitcoin or Ethereum would require resources so enormous as to be economically irrational – the attack would likely destroy the value of the cryptocurrency the attacker holds. Smaller, newer blockchains with less total network participation are more vulnerable.

3. Give the Ukrainian equivalents to the following words and word combinations:

cryptographically linked; peer-to-peer network; central authority; to alter retroactively; fixed-length output; to invalidate a block; computationally infeasible; to reach agreement; computationally intensive puzzle; nonce; block header; block reward;

transaction fee; to lock up as collateral; fraudulent transaction; to slash a stake; energy consumption; trade-off; alternative chain; economically irrational.

4. Fill in the gaps with the words from the list:

blockchain, hash, consensus, PoW, PoS, smart contract, node, wallet, NFT, tokenisation

1. A _____ is a chain of cryptographically linked blocks storing transaction records across a distributed network.
2. A cryptographic _____ is a fixed-length output produced from input data that changes completely if the input changes.
3. A _____ mechanism enables all network participants to agree on the current state of the ledger.
4. In _____, miners compete to solve computationally intensive puzzles to add new blocks.
5. In _____, validators are chosen based on the quantity of cryptocurrency they have locked up.
6. A _____ is a self-executing program stored on a blockchain that enforces agreement terms automatically.
7. Every computer that participates in a blockchain network and maintains a copy of the ledger is called a _____.
8. A cryptocurrency _____ stores a user's private key, which proves ownership of funds.
9. An _____ is a unique digital asset recorded on a blockchain, representing ownership of a specific item.

10. _____ converts rights to a real-world asset into digital tokens that can be traded on a blockchain.

5. Match the terms in column A with their definitions in column B:

#	Term		Definition
1	DApp	a	a decentralised application running on a blockchain rather than a centralised server
2	51% attack	b	representing ownership of real-world assets as digital tokens
3	tokenisation	c	the fixed-length output of a cryptographic hash function
4	hash	d	an attack where one entity controls the majority of network power
5	block reward	e	cryptocurrency given to the miner or validator who adds a new block
6	staking	f	locking cryptocurrency as collateral to participate in PoS consensus

6. Complete the sentences using the correct form of the verb in brackets (Active or Passive):

1. Each block _____ (contain) a cryptographic hash of the previous block.
2. If any block _____ (modify), all subsequent hashes _____ (invalidate).

3. Miners _____ (compete) to solve a computationally intensive puzzle in Proof of Work.
4. The Ethereum network _____ (switch) from Proof of Work to Proof of Stake in 2022.
5. Smart contracts _____ (execute) automatically when predefined conditions _____ (meet).
6. A validator's stake _____ (slash) if they _____ (attempt) to include fraudulent transactions.

7. Read and translate the text.

Blockchain Applications Beyond Cryptocurrency

Although blockchain technology first gained public attention through Bitcoin, its potential applications extend far beyond peer-to-peer digital cash. The fundamental properties of blockchain – tamper-resistant record keeping, transparent and auditable transaction history, decentralised control, and programmable logic through smart contracts – make it relevant wherever multiple parties need to share a trusted record of transactions or events without relying on a centralised intermediary.

Smart contracts, self-executing programs stored on a blockchain whose terms are written directly in code, are the foundation of a wide range of decentralised applications. In decentralised finance (DeFi), smart contracts replace traditional

financial intermediaries: users can lend, borrow, trade, and earn interest on digital assets through protocols that execute automatically without requiring a bank or exchange. Supply chain management uses blockchain to create an immutable audit trail: each step in the journey of a product – from raw material sourcing to manufacturing, shipping, and retail – can be recorded on-chain, enabling consumers and regulators to verify provenance and authenticity.

Digital identity management is another promising application. Traditional identity systems require users to store sensitive personal information with multiple institutions, each of which becomes a target for data breaches. Self-sovereign identity (SSI) systems use blockchain to allow individuals to control their own identity credentials, selectively disclosing only the information required for a specific interaction – proving age without revealing a birthdate, for example – without a centralised identity provider holding a master record.

Non-Fungible Tokens (NFTs) represent unique ownership of a digital or physical asset on a blockchain. Unlike cryptocurrencies, which are fungible (each unit is identical and interchangeable), each NFT is unique. NFTs gained prominence in digital art, where they allow creators to establish verified scarcity and provenance for digital works and to receive royalties automatically through smart contracts when their work is resold. Applications extend to music rights

management, gaming items, real estate tokenisation, and event ticketing.

The governance of blockchain networks is itself an evolving challenge. Public blockchains require mechanisms for the community to agree on protocol upgrades without central authority. Disagreements can lead to hard forks – permanent divergences in the chain where two incompatible versions continue in parallel, as occurred with Bitcoin Cash splitting from Bitcoin in 2017. Permissioned blockchains, used by enterprises and consortia, allow a known set of participants to maintain the ledger while restricting write access, trading the openness of public blockchains for privacy and performance.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. Blockchain technology was developed specifically for cryptocurrency and has no other practical applications.
2. Smart contracts execute automatically when predefined conditions recorded in their code are met.
3. DeFi uses smart contracts to provide financial services without traditional intermediaries such as banks.
4. NFTs are fungible tokens, meaning each unit is identical and interchangeable.
5. A hard fork occurs when a blockchain network splits into two incompatible versions.

6. Permissioned blockchains restrict write access to a known set of participants.

9. Give the English equivalents to the following words and word combinations:

розподілена книга обліку; криптографічно пов'язаний ланцюг; механізм консенсусу; додавання нового блоку; доказ виконання роботи; доказ частки; розумний контракт; самовиконуваний код; децентралізовані фінанси; управління ланцюгом постачань; незмінний аудиторський слід; цифрова ідентичність; самосуверенна ідентичність; вибіркоче розкриття даних; невзаємозамінний токен; верифікована рідкість; управління блокчейном; жорстке розгалуження; дозволений блокчейн; токенизація активів.

10. Answer the questions:

- What makes blockchain records resistant to retroactive tampering?
- What is the difference between Proof of Work and Proof of Stake?
- What is a smart contract and what advantages does it offer over traditional contracts?
- What are three real-world applications of blockchain beyond cryptocurrency?

- What is an NFT and how does it differ from a cryptocurrency?
- What is a hard fork and why does it occur?

LESSON 15

QUANTUM COMPUTING

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
quantum computing	квантові обчислення
qubit	кубіт
quantum superposition	квантова суперпозиція
quantum entanglement	квантове заплутування
quantum gate	квантовий вентиль
quantum circuit	квантова схема
decoherence	декогеренція
quantum error correction	квантове виправлення помилок
quantum supremacy / advantage	квантова перевага
Shor's algorithm	алгоритм Шора
Grover's algorithm	алгоритм Гровера
post-quantum cryptography	постквантова криптографія
NISQ device	пристрій NISQ (шумний квантовий комп'ютер середнього масштабу)

English term / word combination	Ukrainian equivalent
quantum annealing	квантовий відпал
quantum key distribution (QKD)	квантовий розподіл ключів

2. Read and translate the text.

The Principles of Quantum Computing

Quantum computing is a fundamentally different model of computation that exploits the principles of quantum mechanics to process information in ways that are impossible for classical machines. The key difference lies in the basic unit of information: where a classical bit can represent only 0 or 1 at any given moment, a quantum bit (qubit) can exist in a superposition of both 0 and 1 simultaneously. This is not simply a matter of the qubit being in an unknown state – in a true quantum superposition, the qubit is genuinely in both states at once, with each having an associated probability amplitude. Measurement collapses the superposition to a definite 0 or 1, with probabilities determined by the amplitudes.

Quantum entanglement is a correlation between the quantum states of two or more qubits such that the state of each cannot be described independently – measuring one instantly determines the correlated aspect of the other, regardless of the physical distance

between them. Entanglement allows quantum computers to perform coordinated operations on multiple qubits simultaneously in ways that have no classical analogue. A quantum computer with n entangled qubits can represent 2^n states simultaneously, enabling exponential parallelism for certain computational problems.

Quantum gates are reversible operations applied to qubits, analogous to the logic gates (AND, OR, NOT) of classical computers. A quantum circuit is a sequence of quantum gates arranged to perform a computation. Building a useful quantum circuit requires maintaining the fragile quantum states of qubits long enough to complete the computation – a challenge, because qubits are extraordinarily sensitive to environmental disturbances. Decoherence, the loss of quantum coherence due to interaction with the environment, introduces errors that corrupt computations. Current quantum hardware is described as NISQ (Noisy Intermediate-Scale Quantum) – devices with 50–1000 qubits that are not yet error-corrected.

Quantum error correction is the technique of encoding one logical qubit redundantly across many physical qubits so that errors can be detected and corrected without destroying the quantum information. While classical error correction is straightforward, quantum error correction faces the additional constraint that quantum states cannot be copied (the no-cloning theorem) and that

measurement itself disturbs quantum states. Theoretical proposals such as the surface code and the toric code show that fault-tolerant quantum computing is possible in principle, but the overhead in physical qubits required is enormous – estimates suggest thousands of physical qubits per logical qubit.

Quantum algorithms provide provable speedups over the best known classical algorithms for specific problem classes. Shor's algorithm can factor large integers in polynomial time, compared with the exponential time required by the best classical factoring algorithms – a capability that would render current public-key cryptography (RSA, ECC) insecure if run on a sufficiently large, error-corrected quantum computer. Grover's algorithm provides a quadratic speedup for unstructured search problems. Quantum simulation algorithms can model molecular and materials systems with exponentially less resources than classical simulation, offering transformative potential for drug discovery and materials science.

3. Give the Ukrainian equivalents to the following words and word combinations:

fundamental difference; basic unit of information; probability amplitude; measurement collapse; quantum correlation; exponential parallelism; classical analogue; logic gate; reversible operation;

environmental disturbance; quantum coherence; to introduce errors; to corrupt computation; error-corrected device; to encode redundantly; no-cloning theorem; fault-tolerant computing; physical qubit overhead; provable speedup; unstructured search.

4. Fill in the gaps with the words from the list:

qubit, superposition, entanglement, decoherence, NISQ, Shor, Grover, post-quantum, QKD, quantum annealing

1. A _____ is the basic unit of quantum information that can exist in a combination of 0 and 1 simultaneously.
2. Quantum _____ allows a qubit to be in multiple states at once until it is measured.
3. Quantum _____ links the states of two qubits so that measuring one instantly determines a correlated property of the other.
4. _____ is the loss of quantum coherence due to interactions with the environment, causing computational errors.
5. A _____ device is a noisy intermediate-scale quantum computer not yet capable of full error correction.
6. _____ algorithm can factor large integers in polynomial time, threatening current public-key cryptography.
7. _____ algorithm provides a quadratic speedup for searching an unstructured database.
8. _____ cryptography develops new algorithms designed to resist attacks from quantum computers.

9. _____ uses quantum mechanical properties to distribute encryption keys with theoretically provable security.
10. _____ is a quantum computing technique used to solve optimisation problems by finding low-energy configurations.

5. Match the terms in column A with their definitions in column B:

#	Term		Definition
1	quantum supremacy	a	a theoretical principle stating that quantum states cannot be perfectly copied
2	no-cloning theorem	b	the demonstration that a quantum computer has solved a problem no classical computer could practically solve
3	surface code	c	a quantum error correction code that requires a 2D lattice of physical qubits
4	logical qubit	d	a qubit that is error-corrected by encoding it across many physical qubits
5	quantum circuit	e	a sequence of quantum gates performing a computation
6	quantum key distribution	f	using quantum mechanics to distribute encryption keys with provable security

6. Rewrite each sentence using a conditional structure (If ... , ...):

1. Large-scale quantum computers exist – they can break RSA encryption.
2. A qubit is measured – superposition collapses to a definite value.
3. Decoherence is not controlled – quantum computation produces incorrect results.
4. Quantum error correction is achieved – fault-tolerant quantum computing becomes possible.
5. Shor's algorithm runs on a powerful quantum computer – current public-key cryptography becomes insecure.
6. Post-quantum cryptographic standards are adopted early – organisations protect themselves against future quantum attacks.

7. Read and translate the text.

Quantum Computing: Applications, Timelines, and Societal Implications

While large-scale, fault-tolerant quantum computers do not yet exist, their potential applications are already being studied intensively. The most transformative near-term opportunity is quantum simulation: the ability to model the quantum mechanical behaviour of molecules and materials at a level of accuracy impossible for classical computers. Drug discovery could be revolutionised if researchers could accurately simulate how candidate molecules interact with biological targets, accelerating the

identification of new medicines and reducing the cost of clinical trials. Materials science applications include the design of more efficient solar cells, better batteries for electric vehicles, and room-temperature superconductors.

Optimisation problems – finding the best solution among a vast number of possibilities – are ubiquitous in logistics, finance, scheduling, and engineering. Classical heuristic algorithms handle these problems well enough in practice, but quantum approaches may find provably better solutions in less time for certain problem structures. Quantum annealing, implemented in D-Wave systems, is a quantum-inspired approach to optimisation already being tested for applications such as traffic routing, portfolio optimisation, and drug compound screening, though its advantage over classical heuristics remains a subject of active research.

The cryptographic implications of quantum computing are the most urgent near-term concern. Shor's algorithm could break RSA and elliptic-curve cryptography – the algorithms protecting the vast majority of encrypted internet traffic, from HTTPS web browsing to financial transactions and government communications – if run on a sufficiently large, error-corrected quantum computer. The threat is not merely future: adversaries may already be collecting encrypted data today with the intention of decrypting it once quantum computers are available – a strategy known as 'harvest now, decrypt later'. This

has prompted NIST to standardise new post-quantum cryptographic algorithms, announced in 2024, and organisations with long-term data security requirements to begin migration planning.

Quantum key distribution (QKD) uses quantum mechanical properties to distribute encryption keys between two parties with theoretically provable security: any eavesdropper necessarily disturbs the quantum state of the transmitted photons, making interception detectable. Commercial QKD systems are already deployed in some high-security networks. However, QKD requires dedicated optical infrastructure (fibre or line-of-sight laser links) and solves only the key distribution problem, not the broader challenge of authentication and post-quantum algorithm migration.

Predicting a precise timeline for quantum computing milestones is notoriously difficult. Industry and government projections vary widely, and progress depends on solving formidable engineering challenges in qubit fabrication, connectivity, coherence time, and error correction. A consensus estimate suggests that quantum computers capable of breaking 2048-bit RSA may be a decade or more away. However, the time required to migrate large-scale cryptographic infrastructure is measured in years to decades, making early preparation prudent regardless of the precise timeline. The quantum computing landscape is developing rapidly, and

computing professionals will benefit from maintaining awareness of advances in this area.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. Fault-tolerant quantum computers are already widely deployed in commercial data centres.
2. Quantum simulation may transform drug discovery by modelling molecular interactions accurately.
3. The 'harvest now, decrypt later' strategy involves collecting encrypted data to decrypt it when quantum computers become available.
4. NIST has not yet standardised any post-quantum cryptographic algorithms.
5. Quantum key distribution makes eavesdropping detectable because it disturbs the quantum state of transmitted photons.
6. The time required to migrate cryptographic infrastructure makes early preparation important even if quantum threats are distant.

9. Give the English equivalents to the following words and word combinations:

квантова механіка; квантовий біт; суперпозиція станів; вимірювання кубіта; квантове заплутування; квантовий вентиль; квантова схема; декогеренція; виправлення квантових помилок; відмовостійкі обчислення; квантова перевага; алгоритм

факторизації; квантове моделювання молекул; постквантова криптографія; стандарти NIST; збір зашифрованих даних; квантовий розподіл ключів; оптична інфраструктура; прогноз розвитку технологій; своєчасна підготовка.

10. Answer the questions:

- What is the key difference between a classical bit and a qubit?
- What is decoherence and why is it a major challenge for quantum computing?
- What makes Shor's algorithm a threat to current public-key cryptography?
- What is post-quantum cryptography and why is it being developed now?
- What is quantum key distribution and what are its limitations?
- Why is early preparation for post-quantum cryptography recommended even though large-scale quantum computers do not yet exist?

LESSON 16

PARALLEL AND DISTRIBUTED COMPUTING

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
parallel computing	паралельні обчислення
distributed computing	розподілені обчислення
thread / process	потік / процес
race condition	стан гонки
deadlock	взаємне блокування
mutex / semaphore	м'ютекс / семафор
Amdahl's law	закон Амдала
MapReduce	MapReduce (модель розподіленої обробки)
cluster / grid computing	кластерні / ґрід-обчислення
MPI (Message Passing Interface)	інтерфейс передачі повідомлень
GPU computing	GPU-обчислення
SIMD / MIMD	SIMD / MIMD (класифікація паралельних архітектур)
load balancing	балансування навантаження

English term / word combination	Ukrainian equivalent
fault tolerance	ВІДМОВОСТІЙКІСТЬ
consistency model	МОДЕЛЬ УЗГОДЖЕНОСТІ ДАНИХ

2. Read and translate the text.

Parallel Computing: Architecture and Challenges

Parallel computing is the simultaneous execution of multiple computations, exploiting the fact that many computational problems can be divided into sub-tasks that are solved concurrently. As clock speed improvements in single-core processors approached physical limits in the mid-2000s, the industry shifted to multi-core architectures, and parallel programming became a necessary skill for extracting performance from modern hardware. Today's high-end consumer CPUs contain 16–64 cores; server processors may have hundreds; and graphics processing units (GPUs) contain thousands of smaller cores optimised for the massively parallel numerical workloads involved in rendering, scientific simulation, and AI training.

Parallel computing architectures are classified by Flynn's taxonomy. SIMD (Single Instruction, Multiple Data) processors apply the same operation to multiple data elements simultaneously –

the approach used by GPU shader units and CPU vector extensions such as AVX. MIMD (Multiple Instruction, Multiple Data) systems have multiple processing units, each executing a different instruction stream on different data – the architecture of multi-core CPUs and distributed computing clusters. Shared-memory systems have multiple processors accessing a common memory space; distributed-memory systems have each processor accessing only its own memory and communicating with others through explicit message passing.

Programming shared-memory parallel systems requires careful management of concurrent access to shared data. A race condition occurs when two threads read and write the same memory location concurrently and the outcome depends on the relative timing of their operations – producing non-deterministic, hard-to-reproduce bugs. Mutual exclusion mechanisms – mutexes and semaphores – allow threads to coordinate access to shared resources by ensuring that only one thread holds a lock at a time. A deadlock occurs when two or more threads are each waiting for a resource held by the other, causing all of them to wait indefinitely.

Amdahl's law quantifies the theoretical maximum speedup achievable by parallelising a computation. If a fraction p of a program can be parallelised and the rest $(1-p)$ must execute sequentially, the maximum speedup with n processors is $1 / ((1-p) + p/n)$. As n approaches infinity, the speedup approaches $1/(1-p)$. This reveals that

even a small sequential bottleneck severely limits the benefit of adding more processors: if 10% of a program is inherently sequential, the maximum possible speedup regardless of the number of processors is only $10\times$. Writing highly parallel code therefore requires minimising sequential bottlenecks.

GPU computing exploits the massive parallelism of graphics processors for general-purpose numerical workloads. CUDA (Compute Unified Device Architecture), developed by NVIDIA, and OpenCL are programming frameworks that allow developers to write code that executes on thousands of GPU cores in parallel. GPU computing has become central to training deep learning models: the matrix operations involved in forward and backward propagation through neural networks map naturally to the SIMD architecture of GPUs, providing speedups of $10\text{--}100\times$ over CPUs for these workloads. Cloud providers now offer GPU-accelerated virtual machines as a standard service.

3. Give the Ukrainian equivalents to the following words and word combinations:

simultaneous execution; to divide into sub-tasks; clock speed limitation; multi-core architecture; Flynn's taxonomy; vector extension; instruction stream; shared-memory system; common

memory space; message passing; concurrent access; non-deterministic bug; mutual exclusion; to hold a lock; sequential bottleneck; theoretical maximum speedup; numerical workload; matrix operation; forward propagation; backward propagation.

4. Fill in the gaps with the words from the list:

thread, race condition, deadlock, mutex, Amdahl, MapReduce, MPI, GPU, load balancing, fault tolerance

1. A _____ is a lightweight unit of execution within a process that shares memory with other threads.
2. A _____ occurs when the outcome of concurrent operations depends on unpredictable thread timing.
3. A _____ occurs when two threads wait indefinitely for resources held by each other.
4. A _____ is a synchronisation primitive that prevents multiple threads from accessing a shared resource simultaneously.
5. _____ law states that the sequential portion of a program limits the maximum achievable speedup.
6. _____ is a programming model for processing large datasets by distributing map and reduce operations across a cluster.
7. _____ is a standard library for writing distributed-memory parallel programs that communicate via message passing.

8. _____ computing uses graphics processors with thousands of cores for massively parallel numerical workloads.
9. _____ distributes work evenly across processors to prevent some from being idle while others are overloaded.
10. _____ is the ability of a distributed system to continue operating correctly despite the failure of some components.

5. Match the terms in column A with their definitions in column B:

#	Term		Definition
1	SIMD	a	one processor, multiple independent instruction streams and data sets
2	MIMD	b	same operation applied simultaneously to multiple data elements
3	semaphore	c	a counting synchronisation primitive allowing limited concurrent access
4	cluster computing	d	a group of networked computers working together as a single system
5	consistency model	e	rules defining how distributed processes see the order of memory updates
6	grid computing	f	geographically dispersed computing resources shared across organisations

6. Insert the correct preposition (in, of, on, for, with, between, across, through, without, from):

1. Parallel computing divides a problem _____ sub-tasks that can be solved concurrently.
2. A race condition arises _____ two threads access shared memory _____ proper synchronisation.
3. Amdahl's law reveals that even a small sequential bottleneck limits the overall speedup _____ adding processors.
4. MapReduce distributes computation _____ a cluster _____ many machines.
5. GPU computing provides speedups _____ 10 to 100 times _____ CPUs for deep learning workloads.
6. Fault tolerance ensures the system continues operating _____ the failure _____ individual nodes.

7. Read and translate the text.

Distributed Systems: Design Principles and the CAP Theorem

A distributed system is a collection of independent computers that appear to users as a single coherent system. Distributed systems are necessary when the scale of a problem exceeds the capacity of a single machine, when geographic distribution of computation or data is required for performance or reliability, or when a service must remain available despite the failure of individual nodes. The internet, cloud computing platforms, content delivery networks, peer-to-peer file sharing networks, and distributed databases are all examples of distributed systems.

Designing distributed systems involves fundamental trade-offs that have no perfect solution. The CAP theorem, proved by Eric Brewer in 2000, states that a distributed data store can guarantee at most two of the following three properties simultaneously: Consistency (every read returns the most recent write or an error), Availability (every request receives a non-error response, though it may not be the most recent data), and Partition tolerance (the system continues operating correctly despite network partitions that prevent some nodes from communicating with others). Since network partitions are an unavoidable reality in any distributed system, designers must choose between prioritising consistency or availability when partitions occur.

The BASE properties – Basically Available, Soft state, Eventually consistent – describe the consistency model adopted by many large-scale distributed systems as an alternative to ACID transactions. In an eventually consistent system, replicas of the same data may temporarily diverge due to network delays or partitions, but are guaranteed to converge to the same value given sufficient time without further updates. This relaxed consistency model enables higher availability and better performance at scale than strict consistency, at the cost of increased complexity in application logic that must handle temporary inconsistencies.

MapReduce, introduced by Google in 2004 and popularised through the open-source Apache Hadoop framework, is a programming model for processing and generating large datasets on clusters. The programmer expresses computation as two functions: Map, which processes each input record independently and emits key-value pairs, and Reduce, which aggregates all values associated with each key. The framework handles parallelisation, data distribution, fault recovery, and load balancing automatically. Apache Spark improved on MapReduce by supporting in-memory processing of iterative workloads, making it far more efficient for machine learning and graph processing.

Fault tolerance in distributed systems is achieved through replication – maintaining multiple copies of data across nodes – and redundancy – running multiple instances of services so that the failure of one does not interrupt the overall system. Consensus algorithms such as Paxos and Raft allow distributed systems to agree on a single value even when some nodes fail or messages are lost, forming the foundation of replicated state machines and distributed databases. Container orchestration platforms such as Kubernetes automate the deployment, scaling, and recovery of distributed applications, restarting failed containers, replacing unhealthy nodes, and redistributing load automatically.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. The CAP theorem states that a distributed system can simultaneously guarantee all three of Consistency, Availability, and Partition tolerance.
2. BASE stands for Basically Available, Soft state, Eventually consistent.
3. In an eventually consistent system, all nodes always return the most recent data immediately after a write.
4. MapReduce handles parallelisation, fault recovery, and load balancing automatically.
5. Apache Spark improved on MapReduce by supporting in-memory processing of iterative workloads.
6. Kubernetes automates deployment, scaling, and recovery of distributed applications.

9. Give the English equivalents to the following words and word combinations:

незалежні комп'ютери; єдина когерентна система; географічний розподіл; відмова вузла; фундаментальний компроміс; теорема CAP; розділення мережі; узгодженість даних; доступність сервісу; модель BASE; кінцева узгодженість; тимчасове розходження реплік; модель MapReduce; пара ключ-значення; відновлення після збою; реплікація даних; алгоритм консенсусу; реплікований автомат станів; оркестровка контейнерів; автоматичне масштабування.

10. Answer the questions:

- What is the difference between parallel computing and distributed computing?
- What is a race condition and how can it be prevented?
- What does Amdahl's law reveal about the limits of parallelisation?
- What are the three properties in the CAP theorem and what trade-off does it impose?
- What is eventual consistency and when is it an acceptable trade-off?
- How does Kubernetes contribute to fault tolerance in distributed systems?

LESSON 17

MOBILE APPLICATION DEVELOPMENT

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
native application	нативний застосунок
cross-platform framework	кросплатформений фреймворк
React Native / Flutter	React Native / Flutter (фреймворки)
Android SDK	Android SDK (набір інструментів розробки)
iOS / Swift / Kotlin	iOS / Swift / Kotlin (мова і платформа)
progressive web app (PWA)	прогресивний вебзастосунок
app store / distribution	магазин застосунків / дистрибуція
push notification	push-сповіщення
offline support	підтримка офлайн-режиму
background processing	фонова обробка
battery and data optimisation	оптимізація акумулятора і трафіку

English term / word combination	Ukrainian equivalent
deep link	глибоке посилання
mobile UI patterns (gestures, navigation)	мобільні патерни UI
App Store / Play Store review	перевірка в App Store / Play Store
analytics and crash reporting	аналітика і звітність про збої

2. Read and translate the text.

Approaches to Mobile Development: Native, Cross-Platform, and PWA

Mobile application development presents a unique set of challenges and decisions not encountered in traditional desktop or web development. The global smartphone market is dominated by two platforms – Apple's iOS and Google's Android – each with its own programming language, development tools, UI paradigms, and distribution channel. Developers must choose an approach that balances performance, user experience quality, development cost, and the ability to reach users on both platforms.

Native development means building separate applications for each target platform using that platform's official tools and languages. iOS apps are written in Swift (or the older Objective-C) using Xcode

and Apple's SDK; Android apps are written in Kotlin (or Java) using Android Studio. Native apps have direct access to all platform APIs, can use the latest hardware features, and deliver the best possible performance and the most platform-consistent user experience. The drawback is cost: maintaining two separate codebases requires approximately twice the development effort, and keeping features in parity across platforms can be challenging.

Cross-platform frameworks address this duplication by allowing developers to write a single codebase that compiles to native code for multiple platforms. React Native, developed by Facebook, uses JavaScript and React to build UIs from native components, translating the JavaScript logic into native calls at runtime. Flutter, developed by Google, uses the Dart programming language and a custom rendering engine to draw its own UI components, providing pixel-perfect visual consistency across platforms independent of the host OS. Both approaches involve trade-offs: cross-platform frameworks typically have some performance overhead, may lag behind native APIs when new platform features are introduced, and can make debugging more complex.

Progressive Web Apps (PWAs) represent a third approach: web applications enhanced with modern browser APIs to provide app-like experiences including offline support (through service workers that cache content), push notifications, home screen

installation, and background synchronisation. PWAs are distributed through URLs rather than app stores, eliminating the friction and review process of native distribution. They are particularly valuable in markets with lower device capability or limited storage, since they require no download before use. However, PWAs have more limited access to device hardware and platform features than native or cross-platform apps, and iOS has historically imposed restrictions on certain PWA capabilities.

Regardless of the chosen approach, mobile development imposes specific constraints. Small screens demand careful information hierarchy and touch-friendly UI patterns – tap targets must be large enough for fingers, navigation must accommodate one-handed use, and content must reflow gracefully across different screen sizes. Battery and data consumption are critical concerns: background processes, network requests, location updates, and sensor polling all drain the battery and consume mobile data, requiring careful optimisation. Push notifications, when used thoughtfully, maintain user engagement; when used excessively, they prompt users to disable them or uninstall the app.

3. Give the Ukrainian equivalents to the following words and word combinations:

unique set of challenges; programming language; development tool; UI paradigm; distribution channel; to balance performance; direct access to APIs; hardware feature; platform-consistent experience; separate codebase; development effort; feature parity; custom rendering engine; pixel-perfect consistency; service worker; to cache content; home screen installation; limited storage; touch-friendly pattern; tap target.

4. Fill in the gaps with the words from the list:

native, cross-platform, PWA, Swift, Kotlin, React Native, Flutter, push notification, offline support, deep link

1. A _____ application is built using the official tools and languages of a specific mobile platform.
2. A _____ framework allows a single codebase to compile to apps for multiple mobile operating systems.
3. A _____ is a web application enhanced with browser APIs to provide app-like features.
4. _____ is Apple's modern programming language for iOS app development.
5. _____ is Google's preferred programming language for Android development.
6. _____ uses JavaScript and React to build UIs from native components on iOS and Android.
7. _____ uses the Dart language and a custom rendering engine for cross-platform mobile development.

8. A _____ delivers a message to a user's device even when the app is not actively open.
9. _____ allows an application to function without an internet connection by caching content locally.
10. A _____ is a URL that opens directly to a specific screen or content within a mobile app.

5. Match the terms in column A with their definitions in column B:

#	Term		Definition
1	Android SDK	a	Apple's development environment for iOS and macOS applications
2	Xcode	b	Google's tools and libraries for building Android applications
3	service worker	c	a background script enabling PWA offline support and push notifications
4	app store review	d	the process Apple or Google uses to approve apps before publication
5	crash reporting	e	collecting data about application failures to support debugging
6	analytics	f	measuring user behaviour within an application to guide decisions

6. Complete the sentences using the correct comparative or superlative form of the adjective in brackets:

1. Native apps generally offer _____ (high) performance than cross-platform apps.
2. Flutter provides _____ (consistent) visual appearance across platforms than React Native.
3. PWAs are _____ (easy) to distribute than native apps because they require no app store submission.
4. The _____ (common) approach for enterprise mobile development is cross-platform frameworks.
5. Battery consumption is one of _____ (critical) concerns in mobile application design.
6. Touch targets must be _____ (large) than 44×44 points to be comfortably tappable.

7. Read and translate the text.

Designing and Publishing Successful Mobile Applications

Designing a successful mobile application requires a deep understanding of both the platform's design language and the specific needs and context of its users. Apple's Human Interface Guidelines and Google's Material Design are comprehensive design systems that define the visual style, interaction patterns, navigation paradigms, and accessibility requirements expected of apps on each platform. Following these guidelines is not merely an aesthetic choice – apps that deviate significantly from platform conventions create friction for users who must unlearn familiar patterns, and may be rejected during app store review for violating platform standards.

Mobile navigation patterns differ significantly from desktop paradigms. Tab bars (iOS) and bottom navigation (Android) are the preferred primary navigation components for apps with a small number of top-level destinations, providing always-visible navigation without requiring the user to return to a home screen. Gesture navigation – swiping left to go back, pulling down to refresh, swiping up to access the home screen – has become standard on both platforms. Forms and text input on mobile require special attention: virtual keyboards occlude significant portions of the screen, scrolling behaviour must be carefully managed to keep active inputs visible, and input validation should provide immediate feedback to prevent frustrating submit-and-fix cycles.

App store distribution imposes a gatekeeping function that affects both quality and discoverability. Apple's App Store and Google Play both require apps to pass a review process before being listed. Apple's review is known for being more rigorous and can take days; Google's review is generally faster but still enforces policies on content, privacy, and technical quality. Apps must comply with privacy requirements including declaring all data types collected and providing a clear privacy policy. App Store Optimisation (ASO) – optimising the app's name, description, screenshots, keywords, and ratings – is the mobile equivalent of search engine optimisation and is important for organic discoverability.

Performance monitoring, crash reporting, and analytics are essential tools for maintaining a production mobile application. Crash reporting services such as Firebase Crashlytics collect detailed information about application failures – the stack trace, device model, OS version, and recent user actions – that allows developers to reproduce and fix bugs efficiently. Analytics tools such as Firebase Analytics and Mixpanel track user behaviour: which features are used, where users drop off in key flows, and how long sessions last. This data drives product decisions and prioritises engineering effort toward improvements that have the greatest impact on users.

The mobile development landscape evolves rapidly. New iOS and Android versions are released annually, introducing new APIs, deprecating old ones, and changing platform behaviours. Successful mobile teams invest in automated testing – unit tests, integration tests, and UI automation tests – to ensure that each new OS version does not break existing functionality. Continuous integration pipelines for mobile development automatically build and test the app on a range of device configurations, catching regressions before they reach users. Subscription-based business models have become dominant in the App Store and Google Play, and monetisation strategy – free with advertising, freemium, paid, or subscription – must be decided early as it shapes many aspects of the application's design.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. Following Apple's Human Interface Guidelines and Google's Material Design is purely an aesthetic choice with no practical consequences.
2. Tab bars and bottom navigation are preferred for primary navigation in apps with a small number of top-level destinations.
3. Apple's App Store review process is generally faster and less rigorous than Google Play's.
4. App Store Optimisation involves improving an app's discoverability through its name, description, keywords, and ratings.
5. Crash reporting tools collect stack traces, device information, and recent user actions to help developers reproduce bugs.
6. Automated testing in mobile development helps ensure that new OS versions do not break existing functionality.

9. Give the English equivalents to the following words and word combinations:

нативний застосунок; кросплатформений фреймворк; прогресивний вебзастосунок; дизайн-система; людський інтерфейс; навігація жестами; нижня навігаційна панель; віртуальна клавіатура; підтвердження форми; розповсюдження через магазин застосунків; процес перевірки; вимоги до конфіденційності; оптимізація в магазині застосунків; звітність

про збої; трасування стека; аналітика поведінки; відсів користувачів; автоматизоване тестування інтерфейсу; безперервна інтеграція; стратегія монетизації.

10. Answer the questions:

- What are the main advantages and disadvantages of native mobile development?
- How do cross-platform frameworks like React Native and Flutter differ in their approach?
- What is a Progressive Web App and when is it a suitable alternative to a native app?
- What mobile-specific design constraints must developers consider?
- What does App Store Optimisation involve and why is it important?
- Why is automated testing particularly important in mobile development?

LESSON 18

INFORMATION RETRIEVAL AND SEARCH ENGINES

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
information retrieval (IR)	інформаційний пошук
search engine	пошукова система
web crawler / spider	вебкраулер / павук
index	пошуковий індекс
inverted index	інвертований індекс
query	пошуковий запит
relevance ranking	ранжування за релевантністю
TF-IDF	TF-IDF (частота термін– обернена частота документа)
PageRank	PageRank (алгоритм ранжування)
precision and recall	точність і повнота
natural language query	запит природною мовою
stemming / lemmatisation	стемінг / лематизація
semantic search	семантичний пошук

English term / word combination	Ukrainian equivalent
knowledge graph	граф знань
search engine optimisation (SEO)	оптимізація для пошукових систем

2. Read and translate the text.

How Search Engines Work: Crawling, Indexing, and Ranking

Information retrieval (IR) is the field of computer science concerned with finding relevant information within large collections of documents in response to a user's query. Web search engines are the most visible application of IR technology, but the same principles underlie enterprise search systems, academic database search, legal document discovery, and the search functionality built into most applications. As the volume of digital information continues to grow exponentially, effective retrieval has become one of the most important and challenging problems in computing.

A web search engine operates in three broad phases: crawling, indexing, and ranking. Web crawling is the process of systematically discovering and downloading web pages. A crawler (also called a spider or bot) starts from a set of seed URLs, downloads the HTML of each page, extracts all hyperlinks it contains, and adds newly

discovered URLs to a queue for future crawling. Well-designed crawlers respect the robots.txt protocol, which allows website owners to specify which parts of their site should not be crawled, and manage crawl rate to avoid overloading servers.

The raw HTML collected by crawlers must be processed and stored in a form that supports fast querying. The central data structure of any search engine is the inverted index: a mapping from each unique term in the corpus to the list of documents containing that term, along with information about the frequency and positions of its occurrences. Constructing and maintaining an inverted index for billions of web pages requires distributed storage systems and continuous incremental updating as new pages are discovered and existing ones change. Text preprocessing steps – tokenisation (splitting text into words), case normalisation, stop-word removal, and stemming or lemmatisation (reducing words to their base form) – reduce the vocabulary size and improve matching across morphological variants.

Once the index is built, the ranking algorithm determines which documents, among potentially millions containing the query terms, should be presented to the user and in what order. TF-IDF (Term Frequency–Inverse Document Frequency) is a classical relevance scoring formula: it weights terms that appear frequently in a document (TF) but rarely in the overall corpus (IDF), identifying

terms that are distinctive to that document. PageRank, developed by Larry Page and Sergey Brin as the foundation of Google, augments content-based ranking with a measure of a page's authority derived from the structure of incoming hyperlinks: pages linked to by many authoritative pages receive higher PageRank scores, reflecting the web's collective judgement of their importance.

Modern search engines combine hundreds of ranking signals beyond TF-IDF and PageRank: query-document semantic similarity computed by neural language models, user engagement signals (click-through rates, dwell time, bounce rate), freshness of content, mobile-friendliness, page load speed, and personalisation based on the user's location, search history, and browsing behaviour. The ranking function is typically learned from data using machine learning methods, balancing relevance against commercial factors such as advertising. The continuous arms race between search engine algorithms and SEO practitioners – who attempt to manipulate rankings through techniques ranging from legitimate content optimisation to deceptive link schemes – makes web search ranking one of the most competitive and rapidly evolving problems in applied computer science.

3. Give the Ukrainian equivalents to the following words and word combinations:

systematic discovery; hyperlink extraction; seed URL; to respect a protocol; raw HTML; inverted index; document corpus; frequency of occurrences; distributed storage; incremental updating; text preprocessing; stop-word removal; morphological variant; ranking algorithm; relevance scoring; content-based ranking; incoming hyperlink; collective judgement; query-document similarity; engagement signal.

4. Fill in the gaps with the words from the list:

crawler, inverted index, TF-IDF, PageRank, stemming, semantic search, knowledge graph, precision, recall, SEO

1. A web _____ systematically visits pages, extracts links, and adds new URLs to a queue for downloading.
2. An _____ maps each term in a corpus to the list of documents containing it.
3. _____ weights terms that are frequent in a document but rare across the corpus.
4. _____ ranks pages higher if they are linked to by many authoritative pages.
5. _____ reduces words to their base form to improve matching across morphological variants.
6. _____ uses the meaning of words and context rather than just keyword matching.

7. A _____ represents entities and their relationships to support structured queries.
8. _____ is the proportion of retrieved documents that are actually relevant to the query.
9. _____ is the proportion of all relevant documents that are actually retrieved.
10. _____ involves optimising a website to achieve higher rankings in search engine results.

5. Match the terms in column A with their definitions in column B:

#	Term		Definition
1	robots.txt	a	a file specifying which parts of a site should not be crawled
2	tokenisation	b	splitting text into individual words or tokens
3	dwell time	c	the time a user spends on a page after clicking a search result
4	bounce rate	d	the proportion of users who leave a site after viewing only one page
5	lemmatisation	e	reducing words to their dictionary base form using morphological analysis
6	knowledge graph	f	a structured representation of entities and their relationships

6. Choose the correct word from the pair in brackets to complete the sentence:

1. A search engine crawler (discovers / exploits) new pages by following hyperlinks.
2. TF-IDF gives higher weight to terms that are (common / distinctive) within a document relative to the rest of the corpus.
3. PageRank (measures / ignores) the authority of a page based on the quality and quantity of incoming links.
4. Stemming (expands / reduces) words to their base form to improve matching across different word forms.
5. High (precision / recall) means that most retrieved documents are relevant, even if some relevant documents are missed.
6. SEO practitioners aim to (increase / decrease) a website's ranking in organic search results.

7. Read and translate the text.

Modern Search: Neural Methods, Semantic Understanding, and Challenges

Classical information retrieval methods based on term frequency statistics and link analysis proved highly effective for a generation of web search, but they have inherent limitations. Keyword-based matching fails when a user's intent is expressed in words different from those in the relevant documents – a search for 'heart attack' may miss highly relevant pages that use only the term 'myocardial infarction'. Ranking based on links can be gamed by

coordinated link schemes and is slow to update as new content appears. These limitations have driven the adoption of neural and semantic approaches to search.

Dense retrieval uses neural language models to encode both queries and documents as dense vector embeddings in a high-dimensional semantic space. Semantically similar queries and documents are placed near each other in this space regardless of the exact words used. Retrieval becomes a nearest-neighbour search in the embedding space rather than a term-matching lookup in an inverted index. This approach, exemplified by models such as BERT (Bidirectional Encoder Representations from Transformers) fine-tuned for retrieval tasks, significantly improves recall for queries where intent cannot be expressed by exact keyword matching.

Knowledge graphs are structured representations of entities and their relationships, used by major search engines to provide direct answers to factual queries rather than merely listing links. When a user searches for 'capital of Germany', a knowledge graph allows the search engine to immediately display 'Berlin' as a featured snippet, drawn from a structured entity record rather than a ranked document. Google's Knowledge Graph, Bing's Satori, and the Wikidata open knowledge base are prominent examples. Knowledge graphs also power entity disambiguation (distinguishing between different

people, places, or organisations sharing the same name) and enable structured queries using languages such as SPARQL.

Evaluating search engine quality requires precise measurement. The standard IR metrics are precision (the fraction of retrieved documents that are relevant) and recall (the fraction of relevant documents that are retrieved). These metrics trade off against each other: a system that retrieves everything has perfect recall but potentially very low precision; one that retrieves only the single most relevant document has high precision but low recall. Mean Average Precision (MAP) and Normalised Discounted Cumulative Gain (NDCG) are composite metrics that account for both the relevance and the ranking position of retrieved results.

Search engines face persistent challenges beyond technical quality. Misinformation and low-quality content are prevalent on the web and can rank highly if they accumulate links or engagement signals through coordinated amplification campaigns. Search personalisation, while improving relevance for individual users, can create filter bubbles – information environments in which users primarily encounter content that confirms their existing beliefs. Adversarial SEO, through which bad actors attempt to manipulate rankings to promote scam, phishing, or disinformation content, is a constant operational challenge. Maintaining high-quality, trustworthy

search results at web scale is one of the most complex ongoing challenges in applied computer science.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. Classical keyword-based search effectively handles queries where user intent is expressed differently from the words in relevant documents.
2. Dense retrieval encodes queries and documents as vector embeddings and uses nearest-neighbour search for retrieval.
3. Knowledge graphs allow search engines to provide direct factual answers rather than only ranked document lists.
4. High precision means that most relevant documents are retrieved, while some irrelevant ones are also returned.
5. Filter bubbles in search personalisation expose users primarily to content that challenges their existing beliefs.
6. Adversarial SEO involves attempts to manipulate search rankings to promote low-quality or deceptive content.

9. Give the English equivalents to the following words and word combinations:

класична система інформаційного пошуку; частотна статистика; ключове слово; намір користувача; векторне вкладення; семантичний простір; пошук найближчого сусіда; двонаправлена модель трансформера; точне збігання ключових

слів; граф знань; пряма відповідь на запит; витягнутий фрагмент; усунення неоднозначності сутностей; структурований запит; середня точність; нормалізований кумулятивний виграш; перевірена якість пошуку; фільтрувальний міхур; маніпуляція результатами пошуку; довіра до інформації.

10. Answer the questions:

- What are the three main phases of web search engine operation?
- What is an inverted index and why is it central to information retrieval?
- What does TF-IDF measure and why is it useful for ranking?
- What advantage does dense retrieval have over classical keyword matching?
- What is a knowledge graph and how does it improve search?
- What is a filter bubble and what problem does it present for search personalisation?

LESSON 19

COMPILER DESIGN AND FORMAL LANGUAGES

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
compiler	компілятор
interpreter	інтерпретатор
lexical analysis / lexer	лексичний аналіз / лексичний аналізатор
parsing / parser	синтаксичний розбір / парсер
abstract syntax tree (AST)	абстрактне синтаксичне дерево
semantic analysis	семантичний аналіз
intermediate representation (IR)	проміжне представлення
code optimisation	оптимізація коду
code generation	генерація коду
formal grammar	формальна граматика
regular expression	регулярний вираз
finite automaton	скінченний автомат
context-free grammar (CFG)	контекстно-вільна граматика
Chomsky hierarchy	ієрархія Хомського

English term / word combination	Ukrainian equivalent
just-in-time compilation (JIT)	КОМПІЛЯЦІЯ «ЯКРАЗ ВЧАСНО»

2. Read and translate the text.

How Compilers Work: Stages of Translation

A compiler is a program that translates source code written in a high-level programming language into a lower-level form – typically machine code or an intermediate bytecode – that can be executed by a computer. Understanding how compilers work provides deep insight into the nature of programming languages, the relationship between source code and machine execution, and the techniques used to improve program performance automatically. Compilers are among the most complex and thoroughly studied software systems in computer science.

The compilation process is conventionally divided into a front end, which analyses the source program and builds an internal representation, and a back end, which transforms the internal representation into output code. The front end comprises three main stages. Lexical analysis (or scanning) reads the raw character stream and groups characters into tokens – the basic syntactic units of the language such as keywords, identifiers, operators, and literals. The

lexer discards whitespace and comments and reports invalid characters as errors. Lexical analysis is typically implemented using finite automata driven by regular expressions.

Parsing (or syntactic analysis) takes the token stream from the lexer and checks whether it conforms to the grammatical rules of the language, building a parse tree or abstract syntax tree (AST) that captures the hierarchical structure of the program. Programming language grammars are typically specified as context-free grammars (CFGs), and parsers are constructed using algorithms such as recursive descent, LL parsing, or LR parsing. The AST serves as the central data structure passed through subsequent compiler stages, representing the program's structure in a form convenient for analysis and transformation.

Semantic analysis checks properties of the program that cannot be expressed in the context-free grammar alone: type checking (verifying that operations are applied to compatible types), scope resolution (ensuring that identifiers are declared before use and used within their scope), and constraint checking (verifying that array indices are within bounds, that function call argument counts match declarations, and similar invariants). Semantic analysis annotates the AST with type information and symbol table entries. Errors detected at this stage – type mismatches, undeclared variables, calling a non-function – are reported as compile-time errors.

The back end transforms the semantically analysed AST, typically via one or more intermediate representations (IR), into output code. Intermediate representations such as LLVM IR or Java bytecode are lower-level than the source language but higher-level than machine code, providing a platform-independent target for front-end code generation and a convenient substrate for optimisation passes. Code optimisation transforms the IR to produce more efficient code while preserving its semantics: common optimisations include constant folding (evaluating constant expressions at compile time), dead code elimination (removing code that can never be reached or whose result is never used), and loop optimisations such as loop unrolling and vectorisation. The final code generation stage translates the optimised IR into the instruction set of the target machine.

3. Give the Ukrainian equivalents to the following words and word combinations:

to translate source code; lower-level form; machine code; intermediate bytecode; compilation process; front end and back end; character stream; basic syntactic unit; to discard whitespace; invalid character; token stream; grammatical rule; parse tree; hierarchical structure; recursive descent; central data structure; type checking; scope resolution; compile-time error; constant folding.

4. Fill in the gaps with the words from the list:

lexer, parser, AST, semantic analysis, IR, optimisation, JIT, regular expression, CFG, Chomsky

1. A _____ reads source characters and groups them into tokens such as keywords and identifiers.
2. A _____ checks whether the token sequence conforms to the language grammar and builds a tree structure.
3. An _____ is a tree representation of the syntactic structure of a source program.
4. _____ checks type compatibility, variable declarations, and other properties beyond grammatical correctness.
5. An _____ is a platform-independent representation used between the front and back ends of a compiler.
6. Code _____ transforms the IR to produce faster or smaller output code while preserving correctness.
7. _____ compilation translates bytecode to native machine code at runtime rather than ahead of time.
8. A _____ is a pattern used to describe and match sets of strings, used in lexical analysis.
9. A _____ describes the syntax of a programming language using production rules.
10. The _____ hierarchy classifies formal grammars by their generative power and the automata that recognise them.

5. Match the terms in column A with their definitions in column B:

#	Term		Definition
1	finite automaton	a	a tree encoding the syntactic structure of source code for further analysis
2	abstract syntax tree	b	a computational model used to recognise patterns described by regular expressions
3	constant folding	c	evaluating constant expressions at compile time rather than runtime
4	dead code elimination	d	removing code that can never execute or whose result is never used
5	loop unrolling	e	replacing a loop with repeated copies of its body to reduce loop overhead
6	symbol table	f	a data structure storing identifier names, types, and scope information

6. Rewrite the sentences using a Participle Clause to replace the underlined subordinate clause:

1. Because it reads the character stream, the lexer groups characters into tokens.
2. After it has built the AST, the compiler passes it to the semantic analyser.
3. While it checks type compatibility, the semantic analyser also resolves variable scopes.
4. Since it uses an intermediate representation, the compiler can target multiple machine architectures.

5. Because they eliminate dead code, optimisation passes reduce the size of the output binary.
6. After they have been optimised, the IR instructions are translated into machine code.

7. Read and translate the text.

Formal Languages, Automata Theory, and Regular Expressions

Formal language theory provides the mathematical foundations of compiler design and programming language theory. A formal language is a set of strings over an alphabet, defined by a grammar – a set of rules for generating strings in the language. Noam Chomsky's hierarchy classifies formal grammars into four types by their generative power. Type 3 (regular grammars) generate regular languages, recognised by finite automata; Type 2 (context-free grammars) generate context-free languages, recognised by pushdown automata; Type 1 (context-sensitive grammars) and Type 0 (unrestricted grammars) describe progressively more powerful language classes.

Regular expressions are a notation for describing regular languages – the simplest class in the Chomsky hierarchy. A regular expression specifies a pattern that a string must match, using operators such as concatenation (ab matches 'ab'), alternation (a|b matches 'a' or 'b'), and Kleene star (a* matches zero or more 'a's).

Regular expressions are implemented by compiling them into finite automata, which can test string membership in $O(n)$ time. They are used throughout computing for lexical analysis, text search and replace, input validation, log parsing, and data extraction.

Context-free grammars (CFGs) are more expressive than regular grammars and describe the syntactic structure of programming languages. A CFG consists of a set of production rules of the form $A \rightarrow \alpha$, where A is a non-terminal symbol and α is a sequence of terminals and non-terminals. The language of a CFG is the set of all terminal strings derivable from the start symbol. Parsing algorithms for CFGs – such as Earley parsing (which handles any CFG in $O(n^3)$), CYK parsing, and the efficient LL and LR family of algorithms used in practice – construct a derivation tree showing how the input string was generated from the grammar.

Just-in-time (JIT) compilation is a hybrid execution strategy that combines the portability of interpretation with the performance of compilation. Rather than compiling source code to machine code ahead of time, a JIT compiler translates bytecode to native code dynamically during execution – typically compiling the most frequently executed ('hot') code paths to take advantage of runtime information unavailable to an ahead-of-time compiler, such as actual data types of dynamically typed variables. JIT compilation is used by the Java Virtual Machine (JVM), the .NET CLR, and JavaScript

engines in web browsers to achieve performance close to that of statically compiled languages.

Domain-specific languages (DSLs) are languages designed for a specific application domain rather than general-purpose programming. Examples include SQL for database queries, HTML/CSS for web markup, LaTeX for scientific document preparation, regular expressions for pattern matching, and Makefile syntax for build automation. DSLs trade generality for expressiveness within their domain: they allow domain experts to express concepts naturally and concisely, often without requiring full programming expertise. Building a DSL typically involves implementing a parser for its syntax and a translator or interpreter for its semantics, tasks for which compiler construction techniques are directly applicable.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. Finite automata are used to recognise context-free languages described by context-free grammars.
2. Regular expressions are compiled into finite automata for efficient string matching.
3. Context-free grammars are more expressive than regular grammars and are used to describe programming language syntax.

4. JIT compilation translates source code to machine code entirely before execution begins.
5. JIT compilation can exploit runtime information unavailable to ahead-of-time compilers.
6. Domain-specific languages are designed for general-purpose programming rather than specific application domains.

9. Give the English equivalents to the following words and word combinations:

формальна мова; алфавіт; породжувальна граматики;
ієрархія Хомського; регулярна мова; скінченний автомат;
контекстно-вільна граматики; бездоганний автомат зі стеком;
правило виведення; нетермінальний символ; рядок виведення;
алгоритм розбору; динамічна компіляція; гаряча ділянка коду;
типізація часу виконання; статично скомпільована мова;
предметно-орієнтована мова; компіляція регулярного виразу;
лексичний аналіз; синтаксичне дерево.

10. Answer the questions:

- What are the main stages of the compilation process?
- What is the purpose of the abstract syntax tree?
- What is the difference between lexical and semantic analysis?

- What are regular expressions and in what contexts are they used?
- What is JIT compilation and what advantage does it have over ahead-of-time compilation?
- What is a domain-specific language and can you give two examples?

LESSON 20

BIOINFORMATICS AND COMPUTATIONAL SCIENCE

1. Read the vocabulary. Pay attention to the pronunciation and meaning of the words and word combinations.

English term / word combination	Ukrainian equivalent
bioinformatics	біоінформатика
genome / genomics	геном / геноміка
DNA sequencing	секвенування ДНК
sequence alignment	вирівнювання послідовностей
BLAST algorithm	алгоритм BLAST
protein structure prediction	передбачення структури білка
AlphaFold	AlphaFold (програма передбачення структури)
phylogenetics	філогенетика
computational biology	обчислювальна біологія
high-performance computing (HPC)	високопродуктивні обчислення
Monte Carlo simulation	метод Монте-Карло
finite element method (FEM)	метод скінченних елементів
numerical analysis	чисельний аналіз

English term / word combination	Ukrainian equivalent
scientific visualisation	наукова візуалізація
reproducible research	відтворювана наука

2. Read and translate the text.

Bioinformatics: Computing at the Intersection of Biology and Computer Science

Bioinformatics is the interdisciplinary field that develops and applies computational methods to analyse biological data, particularly the large-scale molecular datasets generated by modern biology. The field emerged in the 1980s and 1990s as DNA sequencing technologies began producing data faster than it could be manually analysed. Today, bioinformatics encompasses sequence analysis, structural biology, systems biology, population genomics, and the computational aspects of drug discovery and personalised medicine. The explosion of genomic data – a human genome contains approximately 3.2 billion base pairs of DNA – has made computational methods not merely useful but essential.

DNA sequencing technologies have undergone several generations of development. First-generation Sanger sequencing could read sequences of a few hundred base pairs at a time, accurate

but slow and expensive. Next-generation sequencing (NGS) technologies introduced in the 2000s massively paralleled the process, reading millions of short DNA fragments simultaneously and reducing the cost of sequencing a human genome from approximately \$100 million in 2001 to under \$1,000 today. Third-generation technologies such as Oxford Nanopore produce longer reads in real time, enabling rapid sequencing of entire genomes in portable devices. Assembling a complete genome from millions of short, overlapping fragments – the sequence assembly problem – is a challenging computational task, typically addressed using de Bruijn graph algorithms.

Sequence alignment is the process of arranging two or more biological sequences to identify regions of similarity that may indicate functional, structural, or evolutionary relationships. Pairwise alignment aligns two sequences; multiple sequence alignment aligns three or more simultaneously. The Smith-Waterman algorithm finds the optimal local alignment between two sequences by dynamic programming. The BLAST (Basic Local Alignment Search Tool) algorithm provides approximate but extremely fast alignment, allowing researchers to search a sequence against databases containing billions of bases in seconds – a critical tool for identifying genes, finding evolutionary relatives, and annotating newly sequenced genomes.

Protein structure prediction – determining the three-dimensional structure of a protein from its amino acid sequence – was one of the grand challenges of computational biology for fifty years. The structure of a protein determines its function, and knowing the structure is essential for understanding disease mechanisms and designing drugs. AlphaFold, developed by DeepMind and released in 2021, achieved near-experimental accuracy on the protein structure prediction benchmark, solving a problem that had resisted the field for decades. AlphaFold uses deep learning, specifically an attention-based neural network architecture, to predict protein structures from sequence data alone. The release of predicted structures for virtually every known protein in the AlphaFold Protein Structure Database has transformed structural biology.

Systems biology applies computational modelling to understand the behaviour of complex biological networks – gene regulatory networks, metabolic pathways, protein interaction networks, and signalling cascades. Rather than studying individual genes or proteins in isolation, systems biology seeks to understand how networks of molecular interactions give rise to the emergent properties of cells, tissues, and organisms. Ordinary and partial differential equations, stochastic simulation methods (such as the Gillespie algorithm), and graph-theoretic analysis are among the computational tools used. These models can generate testable

predictions about system behaviour under perturbation, accelerating experimental design.

3. Give the Ukrainian equivalents to the following words and word combinations:

interdisciplinary field; large-scale molecular dataset; DNA sequencing technology; base pair; next-generation sequencing; short DNA fragment; sequence assembly; overlapping fragment; de Bruijn graph; sequence alignment; evolutionary relationship; dynamic programming; fast alignment; newly sequenced genome; protein structure prediction; amino acid sequence; three-dimensional structure; disease mechanism; drug design; complex biological network.

4. Fill in the gaps with the words from the list:

genome, sequencing, alignment, BLAST, AlphaFold, phylogenetics, HPC, Monte Carlo, FEM, reproducible

1. A _____ is the complete set of DNA instructions in an organism, containing all its genetic information.
2. DNA _____ determines the order of base pairs in a DNA molecule.

3. Sequence _____ arranges biological sequences to identify regions of similarity.
4. _____ is an algorithm for rapidly searching sequence databases to find similar sequences.
5. _____ is a deep learning system that predicts protein three-dimensional structures from amino acid sequences.
6. _____ uses molecular sequence data to study evolutionary relationships between organisms.
7. _____ computing uses supercomputers or clusters to solve large-scale scientific computational problems.
8. _____ simulation uses random sampling to model systems with many variables or sources of uncertainty.
9. The _____ divides a physical domain into small elements and solves differential equations numerically.
10. _____ research provides data, code, and methods in a form that allows others to verify and repeat the results.

5. Match the terms in column A with their definitions in column B:

#	Term		Definition
1	de Bruijn graph	a	a graph-based approach to assembling genomes from short sequencing reads
2	Smith-Waterman	b	a dynamic programming algorithm for optimal local sequence alignment
3	AlphaFold	c	a deep learning system achieving near-experimental accuracy in protein structure prediction

#	Term		Definition
4	systems biology	d	studying biological behaviour through computational modelling of molecular networks
5	numerical analysis	e	the study of algorithms for computing mathematical problems approximately
6	scientific visualisation	f	representing complex computational results graphically for interpretation

6. Complete the sentences using the correct form of the verb in brackets (Active or Passive, present or past):

- AlphaFold _____ (develop) by DeepMind and _____ (release) publicly in 2021.
- The BLAST algorithm _____ (allow) researchers to search billions of bases in seconds.
- Genome assembly _____ (address) using de Bruijn graph algorithms.
- Systems biology models _____ (generate) testable predictions about network behaviour.
- The cost of sequencing a human genome _____ (reduce) from \$100 million in 2001 to under \$1,000 today.
- Protein structures _____ (determine) by X-ray crystallography before computational methods _____ (become) accurate enough.

7. Read and translate the text.

Computational Science: Simulation, High-Performance Computing, and Reproducibility

Computational science is the discipline that uses computing to model, simulate, and analyse phenomena across all scientific domains. It has become a third pillar of scientific discovery alongside theory and experiment: physical experiments are sometimes too expensive, dangerous, or slow to perform; mathematical theory alone may be insufficient for complex systems; computation can bridge the gap by simulating systems at scales and under conditions inaccessible to experiment. Computational science is central to climate modelling, astrophysics, materials design, fluid dynamics, earthquake simulation, and nuclear energy research.

High-performance computing (HPC) provides the computational resources needed for large-scale scientific simulations. Modern HPC systems – supercomputers – consist of thousands of interconnected processors sharing a high-bandwidth, low-latency network. The TOP500 list ranks the world's most powerful supercomputers semi-annually; as of early 2025, the most powerful systems achieve performance measured in exaflops (10^{18} floating-point operations per second). Programming HPC systems typically uses the MPI standard for inter-node communication and OpenMP for shared-memory parallelism within a node. Scientific libraries optimised for specific hardware architectures – such as BLAS (Basic

Linear Algebra Subprograms) and FFTW (Fastest Fourier Transform in the West) – provide the numerical building blocks for many simulations.

The Monte Carlo method is a broad class of computational algorithms that rely on repeated random sampling to approximate solutions to problems that are analytically intractable or computationally infeasible to solve exactly. Named after the Monaco casino because of its reliance on randomness, Monte Carlo is used to evaluate integrals in high-dimensional spaces, simulate the physical transport of particles through matter (in nuclear and medical physics), model financial risk, and estimate the probability of complex system failures. The accuracy of Monte Carlo estimates improves with the square root of the number of samples – doubling accuracy requires quadrupling the number of samples – making massive parallelism particularly valuable.

The finite element method (FEM) is a numerical technique for solving differential equations that describe physical phenomena such as structural deformation, heat transfer, fluid flow, and electromagnetic fields. FEM divides the physical domain into a mesh of small, simple elements (triangles, tetrahedra, hexahedra), approximates the solution within each element as a simple polynomial function, and assembles a large system of algebraic equations whose solution approximates the solution of the original

differential equation across the entire domain. Commercial FEM software such as ANSYS and COMSOL is used extensively in engineering design and analysis.

Reproducibility is a fundamental requirement of scientific research: results should be obtainable by independent researchers following the same methods with the same data. Computational science faces specific reproducibility challenges: software dependencies change over time, random number seeds may not be fixed, hardware floating-point behaviour can vary subtly across architectures, and insufficient documentation of computational environments makes exact replication difficult. Practices that promote computational reproducibility include publishing all code and data alongside results, using version control for all analysis code, using containerisation (Docker) to capture the full computational environment, and specifying random seeds explicitly. Platforms such as Zenodo and Figshare enable researchers to archive and cite datasets and code with persistent identifiers.

8. Define sentences as True (T) or False (F). Correct the false sentences:

1. Computational science has replaced both theory and experiment as the primary method of scientific discovery.

2. Modern supercomputers can achieve performance measured in exaflops.
3. The Monte Carlo method uses deterministic mathematical formulae to compute exact solutions.
4. The accuracy of Monte Carlo estimates improves with the square root of the number of samples.
5. The finite element method divides the physical domain into small elements and approximates the solution within each.
6. Containerisation using Docker helps ensure computational reproducibility by capturing the full software environment.

9. Give the English equivalents to the following words and word combinations:

обчислювальна наука; моделювання складних систем;
третій стовп наукового відкриття; кліматичне моделювання;
астрофізика; гідродинаміка; суперкомп'ютерна система;
продуктивність в ексафлопах; стандарт MPI; чисельна
бібліотека; метод Монте-Карло; випадкова вибірка; аналітично
нерозв'язна задача; метод скінченних елементів; сітка
скінченних елементів; диференціальне рівняння;
відтворюваність наукових результатів; версійний контроль;
контейнеризація середовища; архів даних і коду.

10. Answer the questions:

- What role does computational science play alongside theory and experiment?
- What is high-performance computing and what are typical applications?
- How does the Monte Carlo method work and in what fields is it used?
- What is the finite element method and what types of problems does it solve?
- What is computational reproducibility and why is it important?
- What practices help ensure that computational research is reproducible?

REFERENCES

Комп'ютерна архітектура та апаратне забезпечення

1. Britannica, The Editors of Encyclopaedia. "Computer." Encyclopedia Britannica. <https://www.britannica.com/technology/computer>
2. Patterson, D. A., & Hennessy, J. L. (2017). Computer Organization and Design: The Hardware/Software Interface (5th ed.). Morgan Kaufmann.
3. Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems (4th ed.). Pearson.

Операційні системи та системне програмне забезпечення

4. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.
5. Love, R. (2013). Linux Kernel Development (3rd ed.). Addison-Wesley.
6. Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems (4th ed.). Pearson.

Мови програмування та алгоритми

7. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press.
8. Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Addison-Wesley.
9. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.

Комп'ютерні мережі та Інтернет

10. Tanenbaum, A. S., & Wetherall, D. J. (2010). Computer Networks (5th ed.). Pearson.
11. Kurose, J. F., & Ross, K. W. (2021). Computer Networking: A Top-Down Approach (8th ed.). Pearson.

12. Stevens, W. R. (1994). TCP/IP Illustrated, Vol. 1: The Protocols. Addison-Wesley.

Бази даних та управління даними

13. Ramakrishnan, R., & Gehrke, J. (2002). Database Management Systems (3rd ed.). McGraw-Hill.
14. Date, C. J. (2003). An Introduction to Database Systems (8th ed.). Addison-Wesley.
15. Kleppmann, M. (2017). Designing Data-Intensive Applications. O'Reilly.

Кібербезпека та захист даних

16. Anderson, R. (2020). Security Engineering: A Guide to Building Dependable Distributed Systems (3rd ed.). Wiley.
17. Stallings, W. (2019). Cryptography and Network Security: Principles and Practice (7th ed.). Pearson.
18. European Parliament. (2016). Regulation (EU) 2016/679 – General Data Protection Regulation (GDPR). Official Journal of the European Union.
19. National Institute of Standards and Technology (NIST). (2018). Framework for Improving Critical Infrastructure Cybersecurity (CSF v1.1). <https://www.nist.gov/cyberframework>

Штучний інтелект та машинне навчання

20. Russell, S., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th ed.). Pearson.
21. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. <https://www.deeplearningbook.org>
22. Vaswani, A., et al. (2017). Attention is All You Need. Advances in Neural Information Processing Systems, 30.
23. European Commission. (2021). Proposal for a Regulation on Artificial Intelligence (EU AI Act). COM(2021) 206 final.

Хмарні та розподілені обчислення

24. Armbrust, M., et al. (2010). A View of Cloud Computing. Communications of the ACM, 53(4), 50–58.
 25. Tanenbaum, A. S., & Van Steen, M. (2016). Distributed Systems: Principles and Paradigms (3rd ed.). Pearson.
 26. Burns, B. (2018). Designing Distributed Systems. O'Reilly.
- Методології розробки програмного забезпечення**
27. Sommerville, I. (2015). Software Engineering (10th ed.). Pearson.
 28. Beck, K., et al. (2001). Manifesto for Agile Software Development. <https://agilemanifesto.org>
 29. Chacon, S., & Straub, B. (2014). Pro Git (2nd ed.). Apress. <https://git-scm.com/book/en/v2>
 30. Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley.
 31. Fowler, M. (2018). Refactoring: Improving the Design of Existing Code (2nd ed.). Addison-Wesley.
 32. IEEE Computer Society. (2014). Guide to the Software Engineering Body of Knowledge (SWEBOK) v3.0.
- Взаємодія людини з комп'ютером та UX-дизайн**
33. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., & Elmqvist, N. (2016). Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th ed.). Pearson.
 34. Nielsen, J. (1994). Usability Engineering. Morgan Kaufmann.
 35. Norman, D. (2013). The Design of Everyday Things (Revised and Expanded Ed.). Basic Books.
 36. World Wide Web Consortium (W3C). (2018). Web Content Accessibility Guidelines (WCAG) 2.1. <https://www.w3.org/TR/WCAG21/>
- Комп'ютерна графіка, мультимедіа та обробка сигналів**

37. Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1995). *Computer Graphics: Principles and Practice* (2nd ed.). Addison-Wesley.
38. Proakis, J. G., & Manolakis, D. G. (2006). *Digital Signal Processing: Principles, Algorithms and Applications* (4th ed.). Pearson.
39. Oppenheim, A. V., & Schaffer, R. W. (2009). *Discrete-Time Signal Processing* (3rd ed.). Pearson.

Цифрова етика та суспільство

40. Floridi, L. (2014). The Ethics of Artificial Intelligence. In N. J. Doorn et al. (Eds.), *Early Engagement and New Technologies*. Springer.
41. Brynjolfsson, E., & McAfee, A. (2014). *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W. W. Norton & Company.
42. Association for Computing Machinery (ACM). (2018). *ACM Code of Ethics and Professional Conduct*. <https://www.acm.org/code-of-ethics>
43. Zuboff, S. (2019). *The Age of Surveillance Capitalism*. PublicAffairs.

Вбудовані системи та Інтернет речей

44. Peckham, M. (2014). *Programming Embedded Systems* (2nd ed.). O'Reilly.
45. Heath, S. (2002). *Embedded Systems Design* (2nd ed.). Newnes.
46. IEEE Standards Association. (2021). *IEEE Standards for IoT Security*. <https://standards.ieee.org/>

Блокчейн і розподілені реєстри

47. Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. <https://bitcoin.org/bitcoin.pdf>

48. Buterin, V. (2013). A Next-Generation Smart Contract and Decentralized Application Platform (Ethereum White Paper). <https://ethereum.org/en/whitepaper/>
49. Antonopoulos, A. M., & Wood, G. (2018). Mastering Ethereum. O'Reilly.

Квантові обчислення

50. Nielsen, M. A., & Chuang, I. L. (2010). Quantum Computation and Quantum Information (10th Anniversary Ed.). Cambridge University Press.
51. Preskill, J. (2018). Quantum Computing in the NISQ Era and Beyond. *Quantum*, 2, 79.
52. National Institute of Standards and Technology (NIST). (2024). Post-Quantum Cryptographic Standards (FIPS 203, 204, 205). <https://csrc.nist.gov/pqcrypto>

Паралельні та розподілені обчислення

53. Grama, A., Gupta, A., Karypis, G., & Kumar, V. (2003). Introduction to Parallel Computing (2nd ed.). Pearson.
54. Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. OSDI'04, 137–150.
55. Brewer, E. (2000). Towards Robust Distributed Systems. PODC Keynote Lecture.

Розробка мобільних застосунків

56. Apple Inc. (2024). Human Interface Guidelines. <https://developer.apple.com/design/human-interface-guidelines>
57. Google LLC. (2024). Material Design 3. <https://m3.material.io>
58. Windmill, E. (2020). Flutter in Action. Manning.

Інформаційний пошук та пошукові системи

59. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press. <https://nlp.stanford.edu/IR-book/>

60. Brin, S., & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1–7), 107–117.

61. Karpukhin, V., et al. (2020). Dense Passage Retrieval for Open-Domain Question Answering. *EMNLP 2020*.

Теорія компіляторів та формальні мови

62. Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). *Compilers: Principles, Techniques, and Tools* (2nd ed.). Pearson.

63. Sipser, M. (2012). *Introduction to the Theory of Computation* (3rd ed.). Cengage Learning.

64. Lattner, C., & Adve, V. (2004). LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation. *CGO 2004*, 75–86.

Біоінформатика та обчислювальна наука

65. Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

66. Jumper, J., et al. (2021). Highly Accurate Protein Structure Prediction with AlphaFold. *Nature*, 596, 583–589.

67. Strang, G. (2007). *Computational Science and Engineering*. Wellesley-Cambridge Press.

68. Wilson, G., et al. (2017). Good Enough Practices in Scientific Computing. *PLOS Computational Biology*, 13(6), e1005510.

Загальна довідкова та методична література

69. Council of Europe. (2020). *Common European Framework of Reference for Languages: Learning, Teaching, Assessment – Companion Volume*. Council of Europe Publishing. <https://www.coe.int/en/web/common-european-framework-reference-languages>

70. Dudley-Evans, T., & St John, M. J. (1998). *Developments in English for Specific Purposes*. Cambridge University Press.

71. Nation, I. S. P. (2001). *Learning Vocabulary in Another Language*. Cambridge University Press.
72. Swales, J. M. (1990). *Genre Analysis: English in Academic and Research Settings*. Cambridge University Press.

Навчальне видання

АНГЛІЙСЬКА МОВА

Навчальний посібник

Укладачі: **Ганніченко** Тетяна Анатоліївна
Жебко Олександр Олегович

Формат 60x84 1/16. Ум. друк. арк. 10,0
Тираж 20 пр. Зам. №__

Надруковано у видавничому відділі
Миколаївського національного аграрного університету
54020, м. Миколаїв, вул. Георгія Гонгадзе, 9

Свідоцтво суб'єкта видавничої справи ДК № 4490 від 20.02. 2013 р.