**UNIT 2**

**I. Make sure that you know the following words:**

Digitally formatted data, to interface the computer, types of input data, to store data, on and off patterns, circuit, numeric value, binary digit, binary code, to convert, to apply, to execute sequentially, to process, time consuming, user-friendly format, computer capabilities, advanced software, accessibility, system software, application software, programming language, operating system, utility software.

**II. Read and translate the text:**

**SOFTWARE DEVELOPMENT**

|  |  |
| --- | --- |
| Although a modern computer is capable of accepting many types of input, it can only operate on digitally formatted data, just as original computers did. Hence software must be created to interface the computer with the various types of input data. Because a computer runs on electricity, data must be stored as a series of on and off patterns.  Computer circuits can be in only one of two states: either on (represented by 1) or off (represented by 0). Each numeric value is known as a binary digit (bit) and unique combinations of those two bits, are what binary code, or machine language, is called. Different patterns in binary code could then be used to represent various input characters. Once data has been converted to binary form, computers would then apply a software program (applications or apps) to the digital input data, sequentially execute the instructions, and successfully process it into information.  Writing software programs in numerical (or digital) format was an immense task for anyone. Therefore, it was necessary to develop software into a more user-friendly format. As programmers developed existing code, new computer capabilities were noticed and the demand for even more advanced software increased. It inspired programmers to develop more software. Thousands of new programs were being written as swiftly as possible and yet the demand continued to increase. Software types typically fall into 2 categories: system software and application software.  System software controls various internal computer activities. Any software that controls such activities will fall into one of three categories: programming language, operating system and utility software.  *Programming languages* are the various methods of writing computer instructions. The instructions adhere to a particular set of protocols for each language. Through the years, more than 200 languages have been developed, some of which are quite specialized. Some of the most popular languages include BASIC, COBOL, Pascal, C, C++, Visual Basic. But regardless of which language a program was written in, a computer can only process binary code. Therefore, each language must eventually be converted back to binary code before any instructions can be followed.  High-level languages were developed for two reasons: one – so programmers could work on different computers without having to learn a new assembly language each time, and software written on one computer could be used by another. A compiler (program translator) was used to help solve these problems by translating a program into machine language and checking the program for syntax errors.  Until 1970, IBM bundled its software with its computers, selling the hardware along with the software needed to run it. IBM began charging a separate fee for its software, thus opening a market for independent software developers to write programs that would run on IBM machines. By the time the first personal computer (PC), called the Altair hit the market in the 1975, there were many well-developed computer languages and competent programmers available to write software for the new industry.  *Operating systems* have become larger and more sophisticated in response to the capabilities of new hardware and other software. CP/M developed by Gary Kildall in 1973 for Intel Corporation, was the first OS that would run on PCs made by different manufacturers, and it had the largest number of programs for data and word processing and calculations. Although it was a powerful operating system, few software developers supported it, referring to write for the growing DOS-based market.  *Utility* software expands the performance of the operating system by adding functions that are not part of the original OS. Utilities perform troubleshooting jobs, inspecting diskettes for damage, file conversion, defragmenting, data compression and file spooling. Some utility programs, such as Symantec’s Norton Utilities, even retrieve data from damaged disks. Utilities can also be used to customize the OS environment. | **digitally formatted data** – данные в цифровом формате  **to interface** – взаимодействовать  **process** – преобразовать  **immense** – огромный  **demand** – спрос  **inspired** – подтолкнуло  **swiftly** – быстро  **typically** – обычно  **utility software** – служебное ПО  **adhere** – придерживаться  **regardless** – независимо  **eventually** – в конечном итоге  **bundle** – поставлять  **along with** – совместно с  **charge a separate fee –** взимать отдельную плату  **hit the market** – выйти в продажу  **in response** – в ответ  **referring** – ссылаясь на необходимость  **troubleshooting** – диагностика  **spooling** – буферизация |

**III. Match the terms in the left-hand column with their definitions in the right-hand column**.

|  |  |
| --- | --- |
| 1. primary | 5.chief |
| 2.definition | 8.precise statement or explanation |
| 3.expand | 1.grow wider or bigger |
| 4.consume | 7.use up |
| 5.restrict | 2.put a limit on |
| 6.eventually | 3.at last, finally |
| 7.compile | 4.make (a book etc) from information that has been collected5 |
| 8.flexible | 10.easily bent |
| 9.enable | 6.make it possible for |
| 10.track | 9.follow |

**IV. Answer the following questions:**

1. Why was software created? What was the reason?

*Software must be created to interface the computer with the various types of input data.*

1. How must data be stored in a computer?

*Data must be stored as a series of on and off patterns*

1. What does the “binary digit” mean?

*Computer circuits can be in only one of two states: either on (represented by 1) or off (represented by 0). Each numeric value is known as a binary digit (bit)*

1. Why does the unceasing demand for more advanced software exist?

*As programmers developed existing code, new computer capabilities were noticed and the demand for even more advanced software increased.*

1. What does the notion “programming language” mean? Can you give any examples?

*Programming languages are the various methods of writing computer instructions. Examples: C, C+, C#, Python*

1. What do you know about high-level languages?

*High-level languages were developed for two reasons: one – so programmers could work on different computers without having to learn a new assembly language each time, and software written on one computer could be used by another.*

1. What is compiler used for?

*A compiler was used to translate a program into machine language and check the program for syntax errors.*

1. What do you know about OS?

*Operating systems have become larger and more sophisticated in response to the capabilities of new hardware and other software.*

1. What kind of OS do you prefer to use? And why?

*I prefer to use Windows 10 because it’s the most widespread*

1. Why do we need utilities?

*Utilities perform troubleshooting jobs, inspecting diskettes for damage, file conversion, defragmenting, data compression and file spooling*

**V. Retell the text briefly using the following expressions:**

Interface the computer with input data, digitally formatted data, binary digit, to process binary code, software on and off patterns, machine language, execute instructions, time consuming, programming languages, operating system, utility software.

**VI. Fill in the gaps with the missing words from the text. Mind that in each item the first letter of the word is used:**

1. At first, numbers were the primary form of **input data**.
2. Software must be created to **interface** the computer with input **data**.
3. Each numeric value is called **binary digit**.
4. System software **controls** internal computer activities.
5. Because of **high-level languages** programmers could work on different computers without having to learn a new **assembly** language each time.
6. The **operating systems** is a group of programs that help computer to interpret commands, **process** the inputs and outputs, and manage data.
7. Utility and application software expand the **performance** of OS.

**VII. Rearrange the words and get the right sentences**:

1. Computer circuit can be in only one of two states.
2. It was necessary to develop software into a user-friendly format.
3. It inspired programmers to develop more software.
4. Software written on one computer could be used on another.
5. Software controls various internal system activities.
6. Programming languages are the various methods of writing computer instructions
7. Each language must be converted back to binary code.
8. Operating systems have become larger and more sophisticated.

**VIII. Supply the missing preposition. Refer to the text if necessary**.

1. Modern computer operates **with** digitally formatted data.
2. A computer runs **by** electricity.
3. Computer circuits can be only **in** two states.
4. The early days of computing were restricted **by** science applications.
5. It increased the demand **of** more advanced software.
6. OS is a group of programs that help … the operation **of** a computer.
7. CP / M was the first OS that could run **the** on PCs made **by** different manufacturers.
8. Utilities inspect diskettes **with** damage.

**IX. There are two words given in each item. You have to explain in what way they are similar and how they differ from each other.**

1. (a) first computers, (b) modern computers;

*It operate on digitally formatted date / Modern computers is capable of accepting many types of input*

1. (a) OS, (b) utility;

*It’s system software / OS is a group of programs for computer commands interpreting, utility expands performance of the OC*

1. (a) utility, (b) apps.

*It’s software / Utility is system software*

**X. Give the opposites of the following words:**

Input - output; quickly - slow; modern - old; fill – consume; notice - ignore; internal - external; specific - common; separate - join; high - low; independent – dependent; competent - incompetent; add - delete; insert - eject; early – late; possible - impossible.

**XI**. **Give the synonyms of the following words:**

Modern - contemproray; accept - agree; realize - understood; increase - extend; restrict - limit; general - main; swiftly - quickly; various - different; particular - specific; error - trouble; memorize - remember; competent - qualified; entire - whole.

**XII.** **Replace the italicized word combinations by appropriate ones given in the list below**. **Some of them might be used more** **than once. Refer to a** **dictionary if necessary.**

*Appear in ; at first; contemporary;**dealing with errors and problems; desire; encourage; examine carefully and select; increase; not cease; payment; quickly; tremendous; understand*.

1. ***At first*** computers used numbers as the primary form of input data.
2. A ***contemporary*** computer is capable of accepting many types of input.
3. Engineers quickly ***understood*** that writing in binary code was extremely difficult.
4. Writing software programs in numerical format was an ***tremendous*** task for anyone.
5. It ***encourage*** programmers to develop more software.
6. IBM began charging a separate ***payment*** for its software.
7. IBM chose DOS as its operating system, and when its open-architecture PC ***appear in*** the market, programmers ***jumped at the chance*** to write application software for it.
8. Utilities perform ***dealing with errors and problems*** jobs, inspecting diskettes for damage.

**XIII. Translate into English.**

1. Хотя современный компьютер способен принимать много типов входных данных, он может работать только с цифровыми данными. Although a modern computer is capable of accepting many types of input, it can only operate on digitally formatted data.
2. Следовательно, программное обеспечение должно быть создано, чтобы обеспечить связь компьютера с разными типами входных данных. Hence software must be created to interface the computer with the various types of input data.
3. В начале своего развития использование компьютеров ограничивалось научными и инженерными применениями. The early days of computing were restricted by science applications
4. Инженеры быстро поняли, что написание программы в двоичном коде было чрезвычайно трудным и утомительным делом. Engineers quickly understood that writing in binary code was extremely difficult and boring.
5. До 1970 года IBM ориентировало программное обеспечение на различные компьютеры и продавало их в комплекте. Until 1970, IBM bundled its software with its computers, selling the hardware along with the software needed to run it.
6. К тому моменту, когда первый персональный компьютер появился на рынке, существовали хорошо развитые компьютерные языки и грамотные программисты способные создать новые программы. By the time the first personal computer (PC) there were many well-developed computer languages and competent programmers available to write software for the new industry.
7. При включении компьютера операционная система загружается автоматически и может активизировать другие программы. When PC is turning on, OS loads automatically and can activate other programs.
8. Утилиты расширяют возможности операционной системы, добавляя ей дополнительные функции. Utilitysoftware expands the performance of the operating system by adding functions.