

Министерство образования Республики Беларусь

Учреждение образования

«Международный государственный экологический  
университет имени А. Д. Сахарова»



КАФЕДРА ИНОСТРАННЫХ ЯЗЫКОВ

PROFESSIONAL ENGLISH

Е. А. Малашенко

# ENGLISH FOR IT STUDENTS

Part I



Минск  
2014

УДК 811.111 (075.8)

ББК 81.2 Анг-923

М87

*Рекомендовано к изданию НМС МГЭУ им. А.Д.Сахарова  
(протокол № от мая 2013 г.)*

**Автор:**

*Е. А.Малашенко*

**Рецензенты:**

*А. Л. Назаренко, доктор филологических наук, профессор, зав. кафедрой лингвистики и информационных технологий, зам. декана факультета иностранных языков и регионоведения МГУ им. М. В. Ломоносова;*

*Т. Г. Ковалева, зав. кафедрой современных языков ГУО КИИ МЧС Республики Беларусь, кандидат филологических наук, доцент*

**Малашенко, Е. А.**

М87 English for IT students = Английский язык для инженеров-программистов/экологов : учеб. пособие для студентов учреждений высшего образования по специальности «Информационные системы и технологии (в экологии/здравоохранении)» / Е. А. Малашенко; под ред. Е. А. Малашенко. – Минск: МГЭУ им. А. Д. Сахарова, 2014. – 124 с.

**ISBN 978-985-551-007-0.**

Учебное пособие по профессионально-ориентированному общению предназначено для развития лексических навыков в области специальной терминологии, совершенствовании умений и навыков устной речи, чтения и перевода текстов, имеющих профессиональную значимость для инженеров-программистов/экологов.

Данное пособие предназначено для студентов факультета мониторинга окружающей среды, а также для разных категорий учащихся, интересующихся данной тематикой.

**УДК 811.111 (075.8)**

**ББК 81.2 Анг-923**

**ISBN 978-985-551-007-0**

© Малашенко Е. А., 2014

© Международный государственный  
экологический университет  
имени А.Д.Сахарова, 2014

## Предисловие

Данное пособие предназначено для студентов технических вузов, получающим специальность инженер-программист/эколог, и направлено на достижение главной цели обучения иностранным языкам – формирование иноязычной коммуникативной компетенции будущего специалиста, позволяющей использовать английский язык как средство профессионального и межличностного общения. Актуальность пособия определяется его направленностью на реализацию задач будущей профессиональной деятельности выпускников экологического вуза. Предполагается, что к началу работы с этим пособием студенты уже освоили курс практической грамматики английского языка и владеют навыками работы с различными словарями.

Идея построения учебного пособия и подбор материала явились результатом работы с профилирующей кафедрой. Пособие состоит из двух частей, охватывающих основные направления данных специальностей. Первая часть содержит тексты для работы в аудитории с последующей самостоятельной проработкой дома, ключевые понятия (на изучение конкретных понятий из рассмотренного материала), грамматические упражнения, направленные на повторение и активизацию грамматических структур с последующим включением в беседу освоенного грамматического материала, лексические упражнения и упражнения на перевод, аудитивные упражнения с использованием видео поддержки (на прогнозирование содержания читаемого, контроль понимания предлагаемого учебного материала с последующим включением в беседу освоенной терминологии (презентация по изученной теме). Во второй части предлагается подобная организация материала с добавлением блока академической письменной речи для подготовки и написания научной работы на иностранном языке и приложение с иллюстративным материалом, представляющим собой необходимый для усвоения материал по пройденным темам. В зависимости от этапа прохождения материала объем текстов варьируется от 1200 печатных знаков для начального этапа и до 1800-1900 печатных знаков для более продвинутого этапа. Планируется выход третьей части (автор-составитель Беляева Т.В.). В третьей части “Vocabulary practice” предлагаются дополнительные упражнения, способствующие расширению активного и пассивного словарного запаса, снятию языковых трудностей; в “Reading practice” приведены дополнительные тексты и упражнения для контроля понимания содержания.

Автор не ставил перед собой задачи проиллюстрировать с помощью предлагаемого материала курс информатики. Целью пособия является обогащение словарного запаса студентов по предлагаемой тематике, развитие лексических навыков в области специальной терминологии, совершенствование навыков письменной и устной монологической речи,

развитие умений понимать содержание прочитанного, формирование навыков перевода и реферирования текстов, имеющих профессиональную значимость для инженеров-программистов/экологов. В пособии представлены актуальные аутентичные тексты, обладающие информационной и познавательной ценностью. Текстовый материал доступен для восприятия, но подразумевает плотную работу со словарем. Учебные тексты сопровождаются комплексом упражнений, направленным на формирование иноязычной коммуникативной компетенции будущего специалиста в сфере его профессиональной деятельности, что позволяет реализовать компетентностный подход, усилить практико-ориентированную составляющую. Представленный грамматический материал пособия имеет отражение в тексте и подразумевает систематизацию пройденного.

В задачу преподавателя входит стимулирование высказываний студентов путем создания соответствующих ситуаций, организаций дискуссий и обсуждений на основе содержания текстов и ключевых понятий. Характер текстового материала и его расположение не обязывает строгому следованию предложенному порядку. Используемая терминология не рассматривается как универсальная, единственно возможная и допускает варьирование. Курсивом выделены слова, предназначенные для выполнения упражнения на снятие лексических трудностей путем определения дефиниций данных слов. Жирным шрифтом выделена лексика, пояснение для которой приводится в разделе KEY CONCEPTS, предназначенном для самостоятельной проработки материала.

Автор выражает благодарность рецензентам пособия - профессору МГУ имени М.В.Ломоносова, заведующей кафедрой лингвистики и информационных технологий, зам. декана факультета иностранных языков и регионоведения, доктору филологических наук Назаренко А. Л. и заведующей кафедрой современных языков ГУО КИИ МЧС Республики Беларусь, кандидату филологических наук, доценту Ковалевой Т.Г.

Автор

## CONTENTS

<b>Part I</b>		
Unit 1	History of computer engineering: Hardware History Overview	5
Unit 2	Information Is A Fundamental Property Of The World Around: What Is Information?	12
Unit 3	Measuring Amount Of Information: How Bits & Bytes Work	19
Unit 4	Microsoft Office: Microsoft Software Suit	25
Unit 5	Electronic Dictionaries & Machine Translation: What Is Machine Translation?	31
Unit 6	Numeral Systems: Different Systems Of Numeration	37
Unit 7	Logical Operations: Boolean Logic	41
Unit 8	Computer Architecture	47
Unit 9	Operating System: The purpose of an operating system	53
Unit 10	Monitor Display Technology	60
Unit 11	Graphics &Image Editor	66
APPENDIX	I	75
APPENDIX	II	76

## UNIT 1

### HISTORY OF COMPUTER ENGINEERING

#### VOCABULARY

#### 1. Match the words with their definitions:

1) cipher (n.) <i>also cypher</i>	['saɪfə]	a) present, appearing, or found everywhere
2) non-volatile (adj.)	[nɒn'vɒlətaɪl]	b) a tube with a nozzle and piston or bulb for sucking in and ejecting liquid in a thin stream
3) inefficient (adj.)	[,ɪnɪ'fɪʃ(ə)nt]	c) taking up much space, typically inconveniently; large and unwieldy
4) bulky (adj.)	['bʌlkɪ]	d) an annual calendar containing important dates and statistical information such as astronomical data and tide tables
5) domain (n.)	[dəu'meɪn]	e) a table or device designed to assist with calculation
6) almanac (n.)	['ɔ:l.mənæk]	f) not achieving maximum productivity; wasting or failing to make the best use of time or resources
7) abacus (n.)	['abəkəs]	g) unreasonably high
8) operand (n.)	['ɒpər(ə)nd]	h) retaining data even if there is a break in the power supply
9) exorbitant (adj.)	[ɪg'zɔ:bɪt(ə)nt]	i) a simple device for calculating, consisting of a frame with rows of wires or grooves along which beads are slid
10) syringe (n.)	[sɪ'rɪndʒ]	j) write-protected
11) reckoner (n.)	['rek(ə)nə]	k) an area of territory owned or controlled by a ruler or government
12) read-only (adj.)	[ri:dəunli]	l) the quantity on which an operation is to be done
13) ubiquitous (adj.)	[ju:'bɪkwɪtəs]	m) a secret or disguised way of writing; a code

## WATCHING

2. Now watch a video 'Computer history in 140 seconds' and say what computer is.

## BEFORE YOU READ

3. Discuss with your partner the following questions.

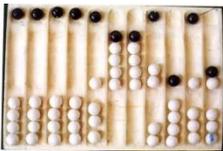
- What do you know about the computers?
- What are the reasons for inventing things?

4. Skim<sup>1</sup> the text to check your ideas.

## READING

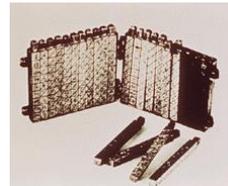
### HARDWARE HISTORY OVERVIEW

The first computers were people. That is, electronic computers (and the earlier mechanical computers) were given this name because they performed the work that had previously been assigned to people. "Computer" was originally a job title: it was used to describe those human beings (predominantly women) whose job it was to perform the repetitive calculations required to compute such things as navigational tables, tide charts, and planetary positions for astronomical almanacs. Imagine you had a job where hour after hour, day after day, you were to do nothing but compute multiplications. Boredom would quickly set in, leading to carelessness, leading to mistakes. And even on your best days you wouldn't be producing answers very fast. Therefore, inventors have been searching for hundreds of years for a way to mechanize (that is, find a mechanism that can perform) this task [John Kop-



plin © 2002].

The abacus was an early aid for mathematical computations. Its only value is that it aids the memory of the human performing the calculation. A skilled abacus operator can work on addition and subtraction problems at the speed of a person equipped with a hand calculator (multiplication and division are slower). The abacus is often wrongly at-

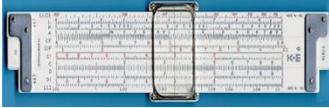


---

<sup>1</sup>Skimming means to get a general idea of what a text is about. Read the first paragraph (the introduction) and the last paragraph (the conclusion). Then read the first and last sentences of the other paragraphs.

tributed to China. In fact, the oldest surviving abacus was used in 300 B.C. by the Babylonians.

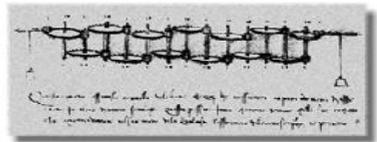
In 1617, an eccentric Scotsman named John Napier invented logarithms, which are a technology that allows multiplication to be performed via addition. The magic ingredient is the logarithm of each operand, which was originally obtained from a printed table. However, Napier also invented an alternative to tables,



where the logarithm values were carved on ivory sticks, which are now called Napier's Bones.

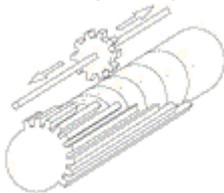
Napier's invention led directly to the slide rule, first built in England in 1632 and still in use in the 1960's by the NASA engineers of the Mercury, Gemini, and Apollo programs which landed men on the moon.

Leonardo da Vinci (1452-1519) made drawings of gear-driven calculating machines but apparently never built any.



The first gear-driven calculating machine to actually be built was probably the calculating clock, so named by its inventor, the German professor Wilhelm Schickard in 1623. This device got little publicity because Schickard died soon afterward in the bubonic plague.

In 1642, Blaise Pascal, at age 19, invented the Pascaline as an aid for his father who was a tax collector. Pascal built 50 of this gear-driven one-function calculator (it could only add) but could not sell many because of their exorbitant cost and because they really weren't that accurate (at that time it was not possible to fabricate gears with the required precision). Up until the present age when car dashboards went digital, the odometer portion of a car's speedometer used the very same mechanism as the Pascaline to increment the next wheel after each full revolution of the prior wheel. Pascal was a child prodigy. At the age of 12, he was discovered doing his version of Euclid's thirty-second proposition on the kitchen floor. Pascal went on to invent probability theory, the hydraulic press, and the syringe.



Just a few years after Pascal, the German Gottfried Wilhelm Leibniz (co-inventor with Newton of calculus) managed to build a four-function (addition, subtraction, multiplication, and division) calculator that he called the stepped reckoner because, instead of gears, it employed fluted drums having ten flutes arranged around their circumference in a stair-step fashion. Although the stepped

reckoner employed the decimal number system (each drum had 10 flutes), Leibniz was the first to advocate use of the binary number system, which is fundamental to the operation of modern computers. Leibniz is considered one of the greatest of the philosophers but he died poor and alone.



[Jacquard and Miskin]

In 1801, the Frenchman Joseph Marie Jacquard invented a power loom that could base its weave (and hence the design on the fabric) upon a pattern automatically read from punched wooden cards, held together in a long row by rope. Descendants of these punched cards have been in use ever since (remember the "hanging chad" from the Florida presidential ballots of the year 2000).

Jacquard's technology was a real boon (benefit) to mill owners, but put many loom operators out of work. Angry mobs smashed Jacquard looms and once attacked Jacquard himself.

History is full of examples of labor unrest following technological innovation yet most studies show that, overall, technology has actually increased the number of jobs.



By 1822, the English mathematician Charles Babbage was proposing a steam driven calculating machine the size of a room, which he called the Difference Engine. This machine would be able to compute tables of numbers, such as logarithm tables. He obtained government funding for this project due to the importance of numeric tables in ocean navigation. By promoting their commercial and military navies, the British government had managed to become the earth's greatest empire. Nevertheless, in

that time frame the British government was publishing a seven volume set of navigation tables which came with a companion volume of corrections which showed that the set had over 1000 numerical errors. It was hoped that Babbage's machine could eliminate errors in these types of tables. However, construction of Babbage's Difference Engine proved exceedingly difficult and the project soon became the most expensive government funded project up to that point in English history. Ten years later the device was still nowhere near complete, acrimony abounded between all involved, and funding dried up. The device was never finished.

Babbage was not deterred, and by then was on to his next brainstorm, which he called the Analytic Engine. This device, large as a house and powered by 6 steam engines, would be more general purpose in nature because it would be programmable, thanks to the punched card technology of Jacquard. But it was Babbage who made an important intellectual leap regarding the punched cards. In the Jacquard loom, the presence or absence of each hole in the card physically allows a colored thread to pass or stops that thread. Babbage saw that the pattern of holes could be used to represent an abstract idea such as a problem statement or the raw data required for that problem's solution. Furthermore, Babbage realized that punched paper could be employed as a storage mechanism, holding computed numbers for future reference. Because of the connection to the Jacquard loom,

Babbage called the two main parts of his Analytic Engine the "Store" and the "Mill", as both terms are used in the weaving industry. The Store was where numbers were held and the Mill was where they were "woven" into new results. In a modern computer these parts are called the memory unit and the CPU. The Analytic Engine also had a key function that distinguishes computers from calculators: the conditional statement.

Babbage befriended Ada Byron, the daughter of the famous poet Lord Byron. Though she was only 19, she was fascinated by Babbage's ideas and through letters and meetings with Babbage she learned enough about the design of the Analytic Engine to begin fashioning programs for the still unbuilt machine. While Babbage refused to publish his knowledge for another 30 years, Ada wrote a series of "Notes" wherein she detailed sequences of instructions she had prepared for the Analytic Engine. The Analytic Engine remained unbuilt but Ada earned her spot in history as the first computer programmer. Ada invented the subroutine and was the first to recognize the importance of looping. Babbage himself went on to invent the modern postal system, cowcatchers on trains, and the ophthalmoscope, which is still used today to treat the eye.



The next breakthrough occurred in America. The U.S. Constitution states that a census should be taken of all U.S. citizens every 10 years in order to determine the representation of the states in Congress. While the very first census of 1790 had only required 9 months, by 1880 the U.S. population had grown so much that the count for the 1880 census took 7.5 years. Automation was clearly needed for the next census. The census bureau offered a prize for an inventor to help with the 1890 census and this prize was won by Herman Hollerith, who proposed and then successfully adopted Jacquard's punched cards for the purpose of computation.

Hollerith's invention, known as the Hollerith desk, consisted of a card reader which sensed the holes in the cards, a gear driven mechanism which could count, and a large wall of dial indicators to display the results of the count.

The patterns on Jacquard's cards were determined when a tapestry was designed and then were not changed. Today, we would call this a read-only form of information storage. Hollerith had the insight to convert punched cards to what is today called a read/write technology. While riding a train, he observed that the conductor didn't merely punch each ticket, but rather punched a particular pattern of holes whose positions indicated the approximate height, weight, eye color, etc. of the ticket owner. This was done to keep anyone else from picking up a discarded ticket and claiming it was his own. Hollerith realized how useful it would be to punch (write) new cards based upon an analysis (reading) of some other set of cards. Complicated analyses, too involved to be accomplished during a single pass through the cards, could be accomplished via multiple passes through the cards using newly printed cards to remember the intermediate results. Unknown to Hollerith, Babbage had proposed this long before.

Hollerith's technique was successful and the 1890 census was completed in only 3 years at a savings of 5 million dollars.

Hollerith built a company, the Tabulating Machine Company which, after a few buyouts, eventually became International Business Machines, known today as IBM. IBM grew rapidly and punched cards became ubiquitous. Your gas bill would arrive each month with a punch card you had to return with your payment. This punch card recorded the particulars of your account: your name, address, gas usage, etc.



Modern computing can probably be traced back to the 'Harvard Mk I' and Colossus. Colossus was an electronic computer built in Britain at the end 1943 and designed to crack the German coding system - Lorenz cipher. The 'Harvard Mk I' was a more general purpose electro-mechanical programmable computer built at Harvard University with backing from IBM.

These computers were among the first of the 'first generation' computers.

First generation computers were normally based around wired circuits containing vacuum valves and used punched cards as the main (non-volatile) storage medium. Another general purpose computer of this era was 'ENIAC' (Electronic Numerical Integrator and Computer) which was completed in 1946. It was typical of first generation computers, it weighed 30 tones contained 18,000 electronic valves and consumed around 25KW of electrical power. It was, however, capable of an amazing 100,000 calculations a second.

The next major step in the history of computing was the invention of the transistor in 1947. This replaced the inefficient valves with a much smaller and more reliable component. Transistorized computers are normally referred to as 'Second Generation' and dominated the late 1950s and early 1960s. Despite using transistors and printed circuits these computers were still bulky and strictly the domain of Universities and governments.

The explosion in the use of computers began with 'Third Generation' computers. These relied Jack St. Claire Kilby's invention - the integrated circuit or microchip; the first integrated circuit was produced in September 1958 but computers using them didn't begin to appear until 1963. While large 'mainframes' such as the I.B.M. 360 increased storage and processing capabilities further, the integrated circuit allowed the development of Minicomputers that began to bring computing into many smaller businesses. Large scale integration of circuits led to the development of very small processing units, an early example of this is the processor used for analysing flight data in the US Navy's F14A 'TomCat' fighter jet. This processor was developed by Steve Geller, Ray Holt and a team from AiResearch and American Microsystems.

On November 15th, 1971, Intel released the world's first commercial microprocessor, the 4004. Fourth generation computers were developed, using a micro-

processor to locate much of the computer's processing abilities on a single (small) chip. Coupled with one of Intel's inventions - the RAM chip (Kilobits of memory on a single chip) - the microprocessor allowed fourth generation computers to be even smaller and faster than ever before. The 4004 was only capable of 60,000 instructions per second, but later processors (such as the 8086 that all of Intel's processors for the IBM PC and compatibles are based) brought ever increasing speed and power to the computers. Supercomputers of the era were immensely powerful, like the Cray-1 which could calculate 150 million floating point operations per second. The microprocessor allowed the development of microcomputers, personal computers that were small and cheap enough to be available to ordinary people. The first such personal computer was the MITS Altair 8800, released at the end of 1974, but it was followed by computers such as the Apple I & II, Commodore PET and eventually the original IBM PC in 1981.

Although processing power and storage capacities have increased beyond all recognition since the 1970s the underlying technology of LSI (large scale integration) or VLSI (very large scale integration) microchips has remained basically the same, so it is widely regarded that most of today's computers still belong to the fourth generation.

*<http://trillian.randomstuff.org.uk>*

## LANGUAGE DEVELOPMENT

### **5. Find and learn Russian equivalents for the following words and expressions:**

1) addition and subtraction problems	a)
2) multiplication and division	b)
3) gear-driven	c)
4) stepped reckoner	d)
5) decimal number system	e)
6) binary number system	f)
7) intellectual leap	g)
8) weaving industry	h)
9) central processing unit	i)
10) integrated circuit	j)

**6. Find and learn English equivalents for the following words and expressions:**

1) схемы с проводным монтажом	a)
2) электрическая лампа накаливания	b)
3) перфорированная карта	c)
4) печатная схема	d)
5) микросхема ОЗУ	e)
6) совместимые устройства	f)
7) требуемая точность	g)
8) инструментальная панель	h)

**7. Create a wordfinder for any 20 computer terms using the following website:**

<http://www.thewordfinder.com/games/wordsearch/fs.wordfinder.php>

**KEY CONCEPTS**

**circuit, circuit board**

Metallic traces printed on a fiberglass panel that carry electrical signals among larger components. Originally, wires ran from and to any component in any electrical device, not just computers. A circuit board replaces the need for separate wiring with the metallic traces printed on the board—sometimes also on the bottom of the board and in a hidden middle layer. The traces lead to connections for processors, resistors, capacitors, and other electrical components. The importance of the circuit board is that its entire creation can be automated, and the board packs more components into an ever-smaller space.

**conditional statement**

A conditional statement allows a program to achieve different results each

time it is run. Based on the conditional statement, the path of the program (that is, what statements are executed next) can be determined based upon a condition or situation that is detected at the very moment the program is running.

**CPU**

An acronym for *central processing unit*, it is used to mean the *microprocessor—also, processor*—which is a microchip that processes the information and the code (instructions) used by a computer. The "brains" of a computer.

**'ENIAC' [Electronic Numerical Integrator and Computer](#)** was typical of first generation computers, it weighed 30 tones contained 18,000 electronic valves and consumed around 25KW of electrical power. It was, however, capable of an amazing 100,000 calculations a second.

**loop**

A section of software commands that returns to its beginning and runs again as long as no specific condition has been encountered.

**motherboard**

A sheet of plastic onto which metallic circuits have been printed and to the rest of the PC's components are connected. These components could be connected via a socket, such as with the CPU, a slot, as with graphics cards and memory modules or they may be built directly onto the motherboard, as with external ports, such as USB.

**memory unit**

(storage unit), a unit of a computer or an independent device designed to record, store, and reproduce information. Memory units are used most widely in digital computers but also have applications in automation, remote-control, nuclear-physics, and other devices, where they store discrete (for the most part) information, coordinate in time the operation of several facilities, or accumulate data to be transmitted through remote-control channels.

**ROM and RAM**

Acronyms for Read Only Memory and Random Access Memory. ROM is memory chips or data stored on disks that can be read by the computer's processor. The PC cannot write new data to those chips or disk drives. RAM is memory or disks that can be both read and written to. Random access memory really is a misnomer because even ROM can be accessed randomly. The term was originally used to distinguish RAM from data and software that was stored on magnetic tape, and which

could be accessed only sequentially. That is, to get to the last chunk of data or code on a tape, a computer must read through all the information contained on the tape until it finds the location where it stored the data or code for which it is looking. In contrast, a computer can jump directly to any information stored in random locations in RAM chips or on disk.

**source** The part of a transistor from which electrical current flows when the transistor is closed.

**transistors**

The transistor is the basic building block from which all microchips are built. The combined chips cover less than a couple of square inches but hold more than 582 million transistors. In computers, transistors are used to represent zeros and ones, the two numbers that make up the binary number system: a 1 if current passes through, or 0 if current doesn't pass through.

These zeros and ones are commonly known as **bits**. Various groupings of these transistors make up the subcomponents within the Core 2, as well as those in coprocessors, memory chips, and other forms of digital silicon.

**write and read**

Writing is the process by which a computer stores data in either RAM chips or on a disk drive. Reading is the process by which a computer transfers data or software code from a drive to RAM or from RAM to the microprocessor.

! To emphasize the action in a sentence, rather than the person who carries/carried out the action we use the Passive. To say who does the action of a passive verb, add **by + noun phrase (agent)** after the verb phrase. The passive form of the verb contains this pattern: **subject + be + Past Participle + object**.

Simple	{	<i>is used</i> <i>are used</i> ; <i>was used</i> <i>were used</i>	Modal pattern	{	<i>will be used</i> <i>could be used</i> <i>can be used</i> ; <i>may be used</i> ; <i>must be used</i> <i>might be used</i>
Progressive	{	<i>is being used</i> <i>are being used</i> ; <i>was being used</i> ; <i>were being used</i>	Perfect	{	$\frac{\textit{has}}{\textit{have}}$ <i>been used</i> <i>had been used</i> ; <i>will have been used</i>
Modal Perfect	{	<i>might have been used</i> <i>could have been used</i>			

The passive is rare in ‘speech’, but common in academic ‘writing’. These are some basic passive forms. You will have noticed the absence of some continuous forms:

- future continuous
- future perfect continuous
- past perfect continuous
- present perfect continuous.

These forms are less common in the passive because they are tricky to say owing to the close proximity of *been* to *being*. *My car has been being repaired* is rather difficult to say, so we don’t!

- You should also note the following points about the passive:
- We only use the passive with transitive verbs, that is, verbs that have an object.
- We sometimes replace the verb *be* with the verb *get*. This is more common in spoken English, e.g. *Trespassers will get prosecuted*.

**8. Look at these sentences from the article, underline and name the Passive forms:**

- *The abacus is often wrongly attributed to China.*
- *The patterns on Jacquard’s cards were determined when a tapestry was designed and then were not changed.*

**9. Find and underline other examples in the text.**

**10. Find the mistakes and correct the sentence.**

- 1) Subtraction it is when one number or amount taken away from another.
- 2) A circuit is a series of places that have been visited regularly by a person or group, especially as a part of their job.
- 3) Theoretically, this machine could handle the four arithmetic functions, had been delayed carrying device, and contained the stepped drum principle, which 150 years later used in the first commercially successful calculator.
- 4) Surveyors may make mistakes or data may enter into the database incorrectly.
- 5) Padded dashboards will have been advocated in the 1930s by car safety pioneer Claire L. Straith.

**11. Make up another sentence with the same meaning using passive structures.**

- 1) They also use this term in the different advancements of computer technology.
- 2) People are constantly developing new discoveries that affect the way we live, work and play.
- 3) Thomas Edison invented the light bulb the same time as Lee De Forest invented vacuum tubes that worked very similar to light bulbs.
- 4) People made transistors of solid material, some of which is silicon, an abundant element (second only to oxygen) found in beach sand and glass.
- 5) People found transistors to conduct electricity faster and better than vacuum tubes.
- 6) However, no one could predict that thousands even now millions of transistors (circuits) people could compact in such a small space.
- 7) Computers could reach more calculation and faster speeds by putting millions of transistors onto one single chip.
- 8) Ted Hoff, employed by Intel (Robert Noyce's new company) invented a chip the size of a pencil eraser that could do all the computing and logic work of a computer.

**12. Translate the following sentences into Russian.**

- 1) Fourth generation computers were developed, using a microprocessor to locate much of the computer's processing abilities on a single (small) chip.
- 2) Transistorized computers are normally referred to as 'Second Generation' and dominated the late 1950s and early 1960s.
- 3) The U.S. Constitution states that a census should be taken of all U.S. citizens every 10 years in order to determine the representation of the states in Congress.

### 13) Translate the following sentences into English.

- 1) Транзистор изобрели в 1947 году.
- 2) Шаблоны перфокарт Жаккарда были использованы при создании gobelena и с тех пор совсем не изменились.
- 3) Было совершенно очевидно, что необходимо автоматизировать процесс переписи населения.

## SPEAKING SECTION

### 14. Answer the following questions.

1. What the first computer originally was?
2. What was the computer originally created for?
3. When was the first computer offered to the public and what was its name?
4. What were Robert Noyce and Jack Kilby known for?
5. What was the microprocessor's original purpose?
6. In each of the 4 generations what was the cause for the increase of speed, power, or memory?
7. How do you think society will be different if scientists are able to create a chip that will perform a trillion operations in a single second?

### 15. Watch a video review on computers:

<http://www.brainpop.com/technology/computersandinternet/computerhistory/>

### 16. Take a quiz at the end:

<http://www.brainpop.com//technology/computersandinternet/computerhistory/quiz/>

### 17. Topics for discussion.

1. How computers used to work.
2. How circuit boards work.
3. Computer generations.



### 18. Prepare a presentation on the discussed topic.

## REFERENCES, USEFUL LINKS AND FURTHER READING

1. 'Computer history in 140 seconds'  
<http://www.youtube.com/watch?v=uudHO9PBGRc>
2. A Chronology of Digital Computing Machines (to 1952), by Mark Brader, 1994.
3. BIT by BIT, An Illustrated History of Computers, by Stan Augarten (1985).
4. 'How computer works', by Ron White, 2008, pp.8-9.

## UNIT 2

### INFORMATION IS A FUNDAMENTAL PROPERTY OF THE WORLD AROUND

#### VOCABULARY

#### 1. Match the words with their definitions:

1) condensed (adj.)	[kən'den(t)st]	a) relating to the sense of smell
2) acoustic(al) (adj.)	[ə'ku:stɪkəl]	b) spread out
3) olfactory (adj.)	[ɒ'l'fækt(ə)rɪ]	c) relating to sound or the sense of hearing
4) flavouring (adj.)	[fl'eɪv(ə)rɪŋ]	d) the arrival of a notable person or thing
5) demise (n.)	[dɪ'maɪz]	e) including or dealing with all or nearly all elements or aspects of something
6) deploy (v.)	[dɪ'plɔɪ]	f) relating to or expressed as a number or numbers
7) integrity (n.)	[ɪn'tegrəti]	g) of or connected with the sense of touch
8) advent (n.)	['ædvənt]	h) the distinctive taste of a food or drink
9) comprehensive (adj.)	[ˌkɒmprɪ'hensɪv]	i) the quality or condition of being whole or undivided; completeness
10) tactile (n.)	['tæktəl]	j) made shorter, usually by including only the most important parts.
11) numerical (n.)	[nju:'merɪkl]	k) the end of smth

## WATCHING

2. Now watch a video ‘What is information?’ and mark True (T) or False (F).

1.	We can touch information.	
2.	Information is used to convey a message from one person to another.	
3.	Information is stored, processed and retrieved by the brain	
4.	Information theory says that universe is made up of two substances.	
5.	Information dictates how matter and energy interact	
6.	Intelligence is necessary for information to exist.	
7.	Intelligence is only required when you are trying to describe information	

## BEFORE YOU READ

1. Discuss with your partner the following questions.

- What do you know about the information?
- What are the reasons for storing information?

2. Skim the text to check your ideas.

## READING

### WHAT IS INFORMATION?



We live in the Information Age and we are surrounded by **information**. The information is everywhere: in human speech, books, magazines and newspapers, radio and TV messages, readings, etc. Information is stored on CD- and DVD-disks, special devices and data carriers. Thanks to new media like the Internet, the Web, blogs, email, cell phones, iPods, e-Readers, Blackberries and iPhones we are blanketed in information.



Information can be found in primary, secondary or tertiary sources depending on how original the materials are or how much they have been interpreted or *condensed* by others. Information can come from many sources and be obtained from everyday life such as watching the channel news or reading the Chicago Tribune and it can be distributed in several ways such as TV news, the Internet, newspaper, and through word of mouth. The farther away from the original source

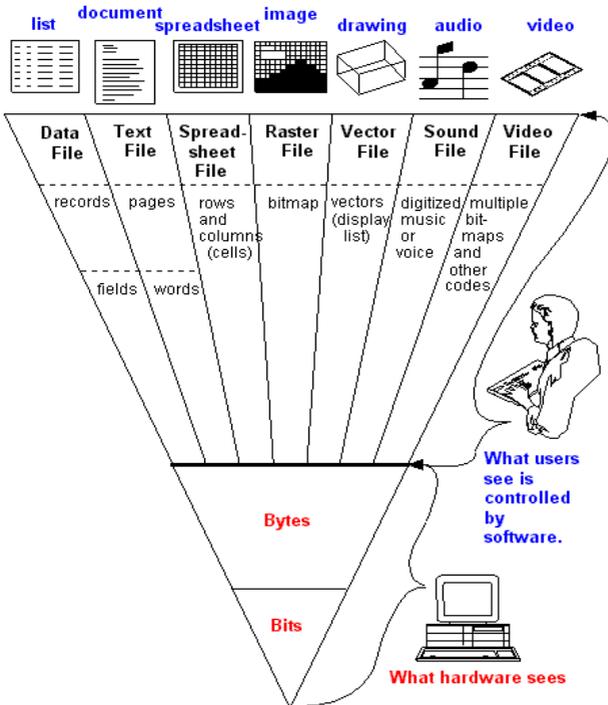
the information is found, the more likely it is to be filtered, interpreted, condensed or otherwise changed.

The person perceives the information by means of sense organs, stores and processes it by means of brain and the central nervous system. In addition to this everyday interaction with information by the users of computer-based digital new media there is the role that information plays in the sciences of **AI** and **AL**. The irony of our total immersion in information as well as the prominent role it plays in AI and AL is that for the most part we do not really have a clear understanding of exactly what information is.

The use of the word 'information' as a descriptive adjective has exploded to the point of near absurdity: information age, information society, information economy, information superhighway, information millennium, information revolution. But what does this word "information" mean in these constructions and how did it become the new keyword of our social formation's self-definition? At first, it appears that the definition of "information" is clear and unproblematic: we all know, in common sense, what it is. But, immediately, it becomes evident that we can't exactly specify the term in its everyday usage, and that the term is being used in some other way when attached to the words "society," "age," etc.

The word 'information' is derived from Latin *informare* which means "give

From Computer Desktop Encyclopedia  
© 2004 The Computer Language Co. Inc.



form to the mind". The way the word information is used can refer to both "facts" in themselves and the transmission of the facts. There have been many descriptions and characterizations of information through the years, some very broad, some narrow and some humorous.

One of the earliest definition of information was produced by Shannon and Weaver. They defined information as what is known beyond random chance predictions. In this sense, infor-

mation must surprise the recipient. It must reduce the uncertainty recipient has about the state of the world. Random chance events represent the state of complete uncertainty. To the extent that recipient can reduce his/her uncertainty, he/she is informed. If the conclusion of a report is known a priori, then it has no surprise value, and therefore it is not, according to this definition, informative. It may be helpful to view information the way it is structured and used, namely: data, text, spreadsheets, pictures, voice and video. Data are discretely defined fields. Text is a collection of words. Spreadsheets are data in matrix (row and column) form. Pictures are lists of vectors or frames of bits. Voice is a continuous stream of sound waves. Video is a sequence of image frames. When information is stored electronically, it is structured according to the way it is used. Databases support all kinds of information. Information may be hard to define but it can be categorized and divided into types by different criteria: on the validity - true, false; on ways of perception - visual, *acoustical*, *tactile*, *olfactory*, *flavouring*; on public value - mass, special, confidential, personal; under its form of representation, ways of its coding and storage - graphic, sound, text, *numerical*, a video information, etc. Information is of value to decision makers if it is accurate, timely, complete, and relevant. These four criteria are used to distinguish valuable information from information that is of less value. **Accurate** information provides a reliable and valid representation of reality. The cost of inaccurate or distorted information can be extremely high. Consider the *demise* of the multimillion dollar Mars Climate Orbiter launched by NASA in 1998. The tragic outcome of this mission was blamed on the failure of one scientific team to recognize and correct an error in information from another team. Findings indicate that one team used English units (e.g., inches, feet and pounds) while the other used metric units for key spacecraft operations affecting navigation. This oversight caused the orbiter to burn up in Mars atmosphere before it could *deploy* to the surface. **Timely** information is information that is available when it is needed. When information is needed almost always depends on the situation. In the fast-paced world of air travel, commercial airlines need virtually daily updates on what other commercial airlines are doing with their ticket prices. If one airline reduces its fares from Newark International Airport to Los Angeles International Airport, other airlines flying the same route would find out quickly about it and respond in a similar manner. Other projects, such as real estate development, might require months of information gathering and assessment. Spending six months preparing a report on the environmental impacts of a real estate development project might be an acceptable timeframe in this situation. **Complete** information tends to be *comprehensive* in covering the issue or topic of interest. Complete information tells a complete story. Without complete information, a decision maker will get a distorted view of reality. Incomplete market information can lead businesses to introduce products and services that customers don't want. For example, the fashion site Boo.com was to be an international Internet fashion superstore that would offer street fashion on the web. Even though the site was designed for 18 different languages and currencies, the site was not

**user-friendly**, the prices were too high in some cases, and you couldn't really get the true feel for the products. The initial concept was very good and did make some money, but like most Internet start-ups, they spent more than they made, which is just not good business. All is not lost; with the right strategy and a test market, Boo might have done quite well...had they had the complete information. Information is **relevant** if it has significance or can be applied to a specific situation, problem, or issue of interest. Here are some examples of relevant information. Human resource managers need information on hiring and employee turnover; operations managers need information on costs and productivity; marketing managers need information on sales projections and advertising rates; top executives need information on the strategic actions of their competitors. In contrast, product inventory information is not very relevant to a computer programmer.

From the point of view of computer science the information has a number of fundamental properties: *integrity*, reliability, availability, up-to-dateness, pertinency, privacy, usability etc.

[Integrity](#) – the quality or condition of being whole or undivided; completeness;

[Reliability](#) – is the ability of system to perform and maintain its functions in routine circumstances;

[Availability](#) of data, data channels, and input-output devices in computers is the condition of being ready for use and not immediately committed to other tasks;

[Up-to-dateness](#) – being in accord with consumer needs just-in-time;

[Pertinency](#) – having logical precise relevance to the matter at hand;

[Privacy](#) – the state of being free from unsanctioned intrusion or [altering](#);

[Usability](#) – fit for use, convenient to use.

The type, quality and amount of information has changed greatly through the ages, but a human's ability to process the information has remained the same, creating what some call information overload.

With the *advent* of the Information age it seems that there is way too much information for us to absorb. The increase in publishing, the growth of news and other media, the advances in worldwide communications and now the explosion of information on the Internet all contribute to the sense that finding the information we need is just about impossible.

Despite the massive change in information storage and retrieval technology over time, our capacity to understand and synthesize information has remained constant.

On the assumption of the aforesaid, the information is data of the objects and of the environment phenomena, their parameters, properties and condition, which are perceived information systems (the live organisms, operating cars, etc.) in the course of ability to live and work.

Information obviously fuels most of our daily activities. Knowledge workers who understand the importance of information will have a unique advantage over those who do not. And even more importantly, those who understand the differ-

ence between good information sources and poor ones will become knowledge leaders.

## LANGUAGE DEVELOPMENT

**5. Find and learn Russian equivalents for the following words and expressions:**

1) data carriers	a)
2) artificial intelligence	b)
3) silicon-based	c)
4) distorted information	d)
5) pertinency	e)
6) altering	f)
7) privacy	g)
8) availability	h)
9) information sources	i)
10) information gathering and assessment	j)

**6. Find and learn English equivalents for the following words and expressions:**

1) специальное приспособление	a)
2) полное погружение	b)
3) передаваемый из уст в уста	c)
4) концепция, моделируемая в системах машин и воспроизводящая их поведение по типу взаимодействия живых организмов	d)
5) удобный для пользователя (дружественный)	e)
6) недавно созданная фирма	f)
7) эргономичность	g)
8) актуальность	h)
9) достоверность	i)
10) полнота	j)

## KEY CONCEPTS

### **(AI) Artificial Intelligence**

AI is used in games for everything from making a computer opponent behave believably like a human opponent to having automated units perform tasks in a realistic manner. In AI intelligence is posited to be a form of information that can be downloaded from the human brain onto a computer.

**(AL) Artificial Life** In AL life is posited to be a form of information that first became embodied in carbon-based chemicals but now can exist in silicon-based computers. Some AL scientists like Edward Fredkin insist that the universe is a computer and that life including human life is merely a programme running on that computer.

### **information**

It is the summarization of data. Technically, data are raw facts and figures that are processed into information, such as summaries and totals. But since information can also be the raw data for the next job or person, the two terms cannot be precisely defined, and both are used interchangeably.

### **silicon**

A brownish crystalline semi metal used to make the majority of semiconductor wafers.

### **semiconductor**

A material (such as silicon) that can be altered to either conduct electrical current or block its passage. Microchips are typically fabricated on semiconductor materials such as silicon, germanium, or gallium arsenide.

### **user friendly**

A system that is easy to learn and easy to use. In the past, this term was so abused in the software business that vendors became reluctant to use it today. Except for mobile phone applications, which due to their tiny screens forces developers to design concisely, a great amount of desktop software is downright "user hostile." It is the rarest of desktop applications that can be used by someone immediately who has never worked with a similar program.

### **wafer**

In semiconductor technology, a very thin piece of silicon that functions as the base material for building microchips. Also called a "slice."

! **Study the following sentence:** *If the conclusion of a report is known a priori, it will not have any surprise value.*

This is the example of a conditional clause (**if + subject + present simple, future/modal/imperative + bare infinitive/V1/base form**), the 1<sup>st</sup> type of adverbial clause (real present).

The first (type 1) conditional is used to talk about things that are possible and may happen in the future. It is used to express different attitudes and ideas, or it is used for particular purposes. These different attitudes, ideas and purposes are sometimes called functions. You can either use other conjunctions: unless (if not), provided/ing that, so long as, as long as, on condition that.

	If	Subject	Present simple (passive/active)	Subject	Will not	bare infinitive/V1/ base form
-	If	the conclusion of a report	is not known a priori,	it	will not	have any surprise value... .
+	If	the conclusion of a report	is known a priori,	it	will	have a surprise value...
	If	Subject	Present simple (passive/active)	Will (not)	Subject	Have
?	If	the conclusion of a report	is known a priori,	will	it	have any surprise value...?

! **Study the following sentence:** *If Sarah didn't have the Internet, she wouldn't chat with her friends.*

This is the example of a conditional clause (**if + subject + past simple, would + bare infinitive/V1**), the 2<sup>nd</sup> type of adverbial clause (unreal present). (*didn't have*) is V2, but in this sentence it is NOT a Past tense, for it does NOT refer to past time. We know that Sarah, at the time when she is speaking, HAS the Internet, and she is able to chat with her friends.

The second (type 2) conditional is used to talk about things that are unlikely, impossible or contrary to facts. This is similar to the way we use the second conditional when we are talking about the present. Although the examples in the task refer to the future, the subordinate *if* clause is formed with the past simple and the main clause uses *would* not *will*.

	If	Subject	Past simple	Subject	would (not)	bare infinitive/V1/ base form
+	If	Sarah	had the Internet,	she	would	chat with her friends.
-	If	Sarah	didn't have the Internet,	she	wouldn't	chat with her friends.
?	If	Subject	Past simple	Would (not)	Subject	bare infinitive/V1/ base form
?	If	Sarah	had the Internet,	Would	she	chat with her friends?

You can say this sentence about Sarah in 3 ways:

*If Sarah didn't have the Internet, she wouldn't chat with her friends.*

*If Sarah didn't have the Internet, she might not chat with her friends.*

*If Sarah didn't have the Internet, she could not chat with her friends.*

There is a difference in meaning:

*wouldn't chat* - it is certain that she wouldn't chat

*might not chat* - perhaps she wouldn't chat

*could not chat* - it wouldn't be possible for her to chat

Sometimes in American English, especially in spoken English, the form of the second conditional is different, with *would* being used in both the main clause and the subordinate clause.

	If	Subject	would	Base form	Subject	would (not)	Base form
+	If	Sarah	would	have the Inter- net,	she	would	chat.

**! Study the following sentence:** *If one scientific team of NASA hadn't made a mistake, they wouldn't have been blamed on the demise of the multimillion dollar Mars Climate Orbiter.*

This is the example of a conditional clause (**if + subject + past perfect, would + have + Pii/V3**), the **3<sup>rd</sup> type** of adverbial clause (unreal past). We use this conditional to reflect on things that happened or didn't happen in the past and how changing these events might have made a difference. It is often used to express regret or to criticise.

The team didn't recognize and correct an error in information from another team, in 1998. We suppose the opposite, using the Past Perfect tense (*hadn't made*). We use *wouldn't have been blamed* to express the past possible result.

	If	Subject	Past perfect	Subject	Would(not)	Have	Past participle
-	If	team of NASA	hadn't made a mistake,	they	would not	have been	blamed....
+	If	team of NASA	had made a mistake,	they	would	have been	blamed....
	If	Subject	Past perfect	Would (not)	Subject	Have	Past participle
?	If	team of NASA	had made a mistake,	would	they	have been	blamed ...?

Sometimes in American English and in colloquial British English, the form of the third conditional is different, with *would* being used in both the main clause and the subordinate *if* clause.

	If	Subject	would (not)	have	Past participle	Subject	would (not)	have	Past participle
+	If	he	would	have	known that information	he	would	have	used it properly

**!** Study the following sentence: *When he left school in 1980, he was offered a job in a bank, but he didn't take it. If he had taken the job, he would (or might) be a bank manager now.*

This is the example of a mixed conditional clause (**if + subject + past perfect, would + infinitive/V1**). Mixed conditional is used when things that happened in the past have a result or a consequence in the present.

He didn't take the job in 1980, in the past. We suppose the opposite, and express this with the Past Perfect tense *had taken*. The present result is expressed by the Conditional tense *would be* (certain) or *might be* (perhaps).

The subordinate *if* clause of this conditional is a third (type 3) conditional that we use to talk about the past and the main clause is a second (type 2) conditional that we use to talk about the present. If he *had taken* the job, he *would (or might) be* a bank manager now.

	<b>If</b>	<b>Sub- ject</b>	<b>Past perfect</b>	<b>Subject</b>	<b>would (not)</b>	<b>Base form</b>	
-	If	he	hadn't taken the job,	he	would not	be a bank manager now.	
+	If	he	hadn taken the job,	he	would	be a bank manager now.	
<b>?</b>	<b>If</b>	<b>Sub- ject</b>	<b>Past perfect</b>	<b>Question word</b>	<b>would (not)</b>	<b>Subject</b>	<b>Past par- ticiples</b>
?	If	he	had tak- en the job,	who	would	he	be now?

**7. Find and underline other examples in the text.**

**8. Find the mistakes and correct the sentence.**

- 1) If I see wrong information, I would correct it.
- 2) If you found an article that provides relevant information for your research topic, you should be taken care to investigate the source to make sure it is valid and reliable.
- 3) If you have identified your device as a first generation iPhone, to click here see the iPhone2G jailbreak page.
- 4) He said that if they are running up-to-date, patched versions they should not face increased risk.
- 5) If start-up idea been did really well and we do not know about it, then would it has been did well?

**9. Use the prompts to make conditional sentences.**

- 1) If the information on the ticket \_\_\_\_\_ (to be incorrect) such as a misspelling on the name or the car color \_\_\_\_\_ (be wrong) this \_\_\_\_\_ (to void/abolish) the ticket. (*unreal past*)
- 2) Can a car license \_\_\_\_\_ (be dismissed) if the police officer \_\_\_\_\_ (to write) the wrong info on it? (*mixed*)
- 3) If your documents \_\_\_\_\_ (to be) in the glove box or console, \_\_\_\_\_ (to get) them and close the compartment. (*real present, imperative*)

4) If your vehicle \_\_\_\_\_ (to have) smoke pouring out and \_\_\_\_\_ (to smell) like a Cheech and Chong reunion, he the officer \_\_\_\_\_ (to search) it without asking. (*unreal past*)

5) If you \_\_\_\_\_ (not/get) a reply or the information is still *wrong*, you \_\_\_\_\_ (may/wish) to contact the credit reference agency from where you obtained your file. (*real present*)

### 10. Translate the following sentences into Russian.

1) Total Immersion, Inc. is an Atlanta based computer consulting firm specializing in software development, systems support and network security.

2) Altering computer input in an unauthorized way requires little technical expertise and is not an uncommon form of theft by employees altering the data before entry or entering false data, or by entering unauthorized instructions or using unauthorized processes.

3) Each time a user chooses a document as being of interest he is making a statement about the pertinence of the document and the material contained within the document.

4) A key aspect of Information Security is to preserve the confidentiality, integrity and availability of an organisation's information.

### 11. Translate the following sentences into English.

1) Если интерфейс не удобен для пользователя, он с ним не будет работать.

2) Если бы ты тогда использовал специальное приспособление, у тебя бы все получилось.

3) Если бы сбор и оценка информации проходили бы в быстром режиме, у нас не было бы сейчас проблем.

4) Если бы носители информации были созданы не на основе кремния, было бы не очень удобно их использовать.

## SPEAKING SECTION

### 12. Answer the following questions.

1. Name some organisations where the collection of data is essential for their continued running?

2. Can you think of a situation where information was not used wisely and an organisation has suffered the consequences?

3. Is the cost of keeping data up-to-date, accurate, and complete worthwhile to the company? Why? Why not?

### 13. Topics for discussion.

1. AL & AI
2. Information and Information technology.



14. – Prepare a presentation on the discussed topic.

## REFERENCES, USEFUL LINKS AND FURTHER READING

1. ‘What is information?’ <http://www.youtube.com/watch?v=tooPnNhPjbs>
2. Shannon CE, Weaver W, 1949. A mathematical theory of communication. Chicago University of Illinois Press.
3. Computer Desktop Encyclopedia copyright ©1981-2012 by The Computer Language Company Inc.
4. How computer works’, by Ron White, 2008, p.106.

## UNIT 3

### MEASURING AMOUNT OF INFORMATION

#### Vocabulary

#### 1. Match the words with their definitions:

1) fiendishly (adv.)	['fi:ndɪʃli]	a) a group of binary digits or bits (usually eight) operated on as a unit
2) bit (n.)	[bit]	b) stated clearly and in detail, leaving no room for confusion or doubt
3) byte (n.)	[bait]	c) develop gradually
4) trial (n.)	[traɪəl]	d) suppose to be the case, without proof
5) digit (n.)	['dɪdʒɪt]	e) a test of the performance, qualities, or suitability of someone or something
6) evolve (v.)	[ɪ'vɒlv]	f) extremely, utterly
7) placeholder (n.)	['pleɪs, həʊldə]	g) one of two digits (0 or 1) in a binary system of notation

8) binary digit (n.)	['bainəri'di:dʒɪt]	h) a unit of information expressed as either a 0 or 1 in binary notation
9) explicit (adj.)	[ɪk'splɪsɪt]	i) a significant zero in the decimal representation of a number
10) assume (v.)[with obj]	[ə'sju:m]	j) a digit is a written symbol for any of the ten numbers from 0 to 9.

## WATCHING

2. NASA Kids Science News segment explains the difference between bits and bytes. Now watch a video ‘What’s the difference between bits and bytes?’ and mark True (T) or False (F).

1.	There is no difference between bits and bytes?	
2.	Byte means 1 or 0	
3.	Computers operate using the base-2 number system, also known as the binary number system	
4.	Bytes is 8 bits long	
5.	1 can stand for a letter or other character	

## BEFORE YOU READ

3. Discuss with your partner the following question.

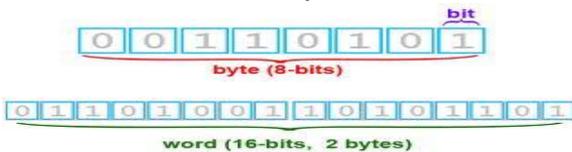
- How bits and bytes relate to the size of digital storage devices?

4. Skim the text to check your ideas.

## READING

### HOW BITS & BYTES WORK

If you have used a computer for more than five minutes, then you have heard the words *bits* and *bytes*. Both RAM and hard disk capacities are measured



in bytes, as are file sizes when you examine them in a file viewer. You might hear an advertisement that says, "This computer has a **32-bit**

Pentium processor with 64 **megabytes** of RAM and 2.1 **gigabytes** of hard disk space." In this article, we will discuss bits and bytes so that you have a complete understanding.

The easiest way to understand bits is to compare them to something you know: **digits**. A digit is a single place that can hold numerical values between 0 and 9. Digits are normally combined together in groups to create larger numbers. For example, 6,357 has four digits. It is understood that in the number 6,357, the 7 is filling the "1s place," while the 5 is filling the 10s place, the 3 is filling the 100s place and the 6 is filling the 1,000s place. So you could express things this way if you wanted to be *explicit*:

$$(6 * 1000) + (3 * 100) + (5 * 10) + (7 * 1) = 6000 + 300 + 50 + 7 = 6357$$

Another way to express it would be to use powers of 10. *Assuming* that we are going to represent the concept of "raised to the power of" with the "^" symbol (so "10 squared" is written as "10^2"), another way to express it is like this:

$$(6 * 10^3) + (3 * 10^2) + (5 * 10^1) + (7 * 10^0) = 6000 + 300 + 50 + 7 = 6357$$

What you can see from this expression is that each digit is a **placeholder** for the next higher power of 10, starting in the first digit with 10 raised to the power of zero.

That should all feel pretty comfortable -- we work with decimal digits every day. The neat thing about number systems is that there is nothing that forces you to have 10 different values in a digit. Our **base-10** number system likely grew up because we have 10 fingers, but if we happened to *evolve* to have eight fingers instead, we would probably have a base-8 number system. Computers happen to operate using the base-2 number system, also known as the **binary number system** (just like the base-10 number system is known as the decimal number system).

The reason computers use the base-2 system is because it makes it a lot easier to implement them with current electronic technology. You could wire up and build computers that operate in base-10, but they would be *fiendishly* expensive right now. On the other hand, base-2 computers are relatively cheap.

Therefore, computers use binary numbers, and therefore use *binary digits* in place of decimal digits. The word bit is a shortening of the words "Binary dig-IT." Whereas decimal digits have 10 possible values ranging from 0 to 9, bits have only two possible values: 0 and 1. Therefore, a binary number is composed of only 0s and 1s, like this: 1011. How do you figure out what the value of the binary number 1011 is? You do it in the same way we did it above for 6357, but you use a base of 2 instead of a base of 10. So:

$$(1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (1 * 2^0) = 8 + 0 + 2 + 1 = 11$$

You can see that in binary numbers, each bit holds the value of increasing powers of 2. That makes counting in binary pretty easy. Starting at zero and going through 20, counting in decimal and binary looks like this:

0 = 0            When you look at this sequence, 0 and 1 are the same for  
 1 = 1            decimal and binary number systems. At the number 2, you see  
 2 = 10          carrying first take place in the binary system. If a bit is 1, and  
 3 = 11          you add 1 to it, the bit becomes 0 and the next bit becomes 1. In  
 4 = 100        the transition from 15 to 16 this effect rolls over through 4 bits,  
 5 = 101        turning 1111 into 10000.

6 = 110            Bits are rarely seen alone in computers. They are almost  
 7 = 111           always bundled together into 8-bit collections, and these collec-  
 8 = 1000          tions are called bytes. Why are there 8 bits in a byte? A similar  
 9 = 1001          question is, "Why are there 12 eggs in a dozen?" The 8-bit byte is  
 10 = 1010        something that people settled on through *trial* and error over the  
 11 = 1011        past 50 years.

12 = 1100            With 8 bits in a byte, you can represent 256 values ranging  
 13 = 1101          from 0 to 255, as shown here:

14 = 1110	0 = 00000000
15 = 1111	1 = 00000001
16 = 10000	2 = 00000010
17 = 10001	...
18 = 10010	254 = 11111110
19 = 10011	255 = 11111111
20 =	
10100	

Next, we'll look at one way that bytes are used. Bytes are frequently used to hold individual characters in a text document. In the **ASCII character set (see Appendix)**, each binary value between 0 and 127 is given a specific character. Most computers extend the ASCII character set to use the full range of 256 characters available in a byte. The upper 128 characters handle special things like accented characters from common foreign languages.

You can see in the Appendix the 127 standard ASCII codes below. Computers store text documents, both on disk and in memory, using these codes. For example, if you use Notepad in Windows 95/98 to create a text file containing the words, "Four score and seven years ago," Notepad would use 1 byte of memory per character (including 1 byte for each space character between the words -- ASCII character 32). When Notepad stores the sentence in a file on disk, the file will also contain 1 byte per character and per space.

Try this experiment: Open up a new file in Notepad and insert the sentence, "Four score and seven years ago" in it. Save the file to disk under the name **getty.txt**. Then use the explorer and look at the size of the file. You will find that the file has a size of 30 bytes on disk: 1 byte for each character. If you add another word to the end of the sentence and re-save it, the file size will jump to the appropriate number of bytes. Each character consumes a byte.

If you were to look at the file as a computer looks at it, you would find that each byte contains not a letter but a number -- the number is the ASCII code corresponding to the character (see below). So on disk, the numbers for the file look like this:

F o u r            a n d            s e v e n  
 70 111 117 114 32 97 110 100 32 115 101 118 101 110

By looking in the ASCII table, you can see a one-to-one correspondence between each character and the ASCII code used. Note the use of 32 for a space -- 32 is the ASCII code for a space. We could expand these decimal numbers out to binary numbers (so 32 = 00100000) if we wanted to be technically correct -- that is how the computer really deals with things.

The first 32 values (0 through 31) are codes for things like **carriage return** and **line feed**. The space character is the 33rd value, followed by punctuation, digits, uppercase characters and lowercase characters. To see all 127 values, check out Unicode.org's chart (<http://www.unicode.org/charts/PDF/U0000.pdf>).

When you start talking about lots of bytes, you get into prefixes like kilo, mega and giga, as in kilobyte, megabyte and gigabyte (also shortened to K, M and G, as in Kbytes, Mbytes and Gbytes or KB, MB and GB). The following table shows the binary multipliers:

Name	Abbr.	Size	
Kilo	K	$2^{10} = 1,024$	You can see in this chart that kilo is about a thousand, mega is about a million, giga is about a billion, and so on. When you consider that one CD holds 650 megabytes, you can see that just three CDs worth of data will fill the whole thing! Terabyte databases are fairly common these days, and there are probably a few petabyte databases floating around the Pentagon by now.
Mega	M	$2^{20} = 1,048,576$	
Giga	G	$2^{30} = 1,073,741,824$	
Tera	T	$2^{40} = 1,099,511,627,776$	
Peta	P	$2^{50} = 1,125,899,906,842,624$	
Exa	E	$2^{60} = 1,152,921,504,606,846,976$	
Zetta	Z	$2^{70} = 1,180,591,620,717,411,303,424$	
Yotta	Y	$2^{80} = 1,208,925,819,614,629,174,706,176$	

Binary math works just like decimal math, except that the value of each bit can be only 0 or 1. To get a feel for binary math, let's start with decimal addition and see how it works. Assume that we want to add 452 and 751:

$$\begin{array}{r}
 452 \\
 + 751 \\
 --- \\
 1203
 \end{array}$$

To add these two numbers together, you start at the right:  $2 + 1 = 3$ . No problem. Next,  $5 + 5 = 10$ , so you save the zero and carry the 1 over to the next place. Next,  $4 + 7 + 1$  (because of the carry) = 12, so you save the 2 and carry the 1. Finally,  $0 + 0 + 1 = 1$ . So the answer is 1203.

Binary addition works exactly the same way:

$$\begin{array}{r}
 010 \\
 + 111 \\
 --- \\
 1001
 \end{array}$$

Starting at the right,  $0 + 1 = 1$  for the first digit. No carrying there. You've got  $1 + 1 = 10$  for the second digit, so save the 0 and carry the 1. For the third digit,  $0 + 1 + 1 = 10$ , so save the zero and carry the 1. For the last digit,  $0 + 0 + 1 = 1$ . So the answer is 1001. If you translate everything over to decimal you can see it is correct:  $2 + 7 = 9$ .

To sum up, here's what we've learned about bits and bytes:

- Bits are binary digits. A bit can hold the value 0 or 1.
- Bytes are made up of 8 bits each.
- Binary math works just like decimal math, but each bit can have a value of only 0 or 1.

## LANGUAGE DEVELOPMENT

**5. Find and learn Russian equivalents for the following words and expressions:**

1) numerical value	a)
2) decimal digit	b)
3) binary number system	c)
4) bundled together	d)
5) accented characters	e)
6) carriage return (CR)	f)
7) uppercase character	g)
8) prefix	h)
9) zettabyte	i)
10) petabyte	j)
11) hexadecimal	k)

**6. Find and learn English equivalents for the following words and expressions:**

1) указатель места заполнения, структурный нуль	a)
2) степень десятки	b)
3) в квадрате	c)
4) десятичная система счисления	d)
5) стандартный набор символов ASCII	e)
6) взаимнооднозначное соответствие	f)
7) перевод строки	g)
8) нижний регистр	h)
9) двоичный умножитель	i)
10) приставка, обозначающая квинтильон, или 2 в степени 60	j)
11) септибайт единица ёмкости памяти, равная 1024 секстибайт или 2 в степени 80	k)

## KEY CONCEPTS

### **ASCII character set**

Acronym for American Standard Code for Information Interchange. 128 characters that make up the ASCII coding scheme.

### **binary**

Consisting of only two integers, 0 and 1. Binary math is the basis for manipulating all data in computers.

### **binary number system**

A method of representing numbers in which only the digits 0 and 1 are used. Successive units are powers of 2. Also called binary system.

### **carriage return**

The operation that prepares for the next character to be printed or dis-

played as the first character on a line.

### **gigabyte**

A unit of computer memory or data storage

capacity equal to 1,024 megabytes ( $2^{30}$  bytes).

### **line feed**

The operation that prepares for the next character to be printed or displayed on the next line.

### **megabyte**

The prefix *mega-* often does not have its standard scientific meaning of 1,000,000, but refers instead to the power of two closest to 1,000,000, which is  $2^{20}$ , or 1,048,576. The calculation of data

storage capacity (measured in bytes) is based on powers of two because of the binary nature of bits (1 byte is 8, or  $2^3$ , bits). Thus, a megabyte is 1,048,576 bytes, although it is also used less technically to refer to a million bytes. Other numerical prefixes are interpreted similarly. With data transmission rates (measured in bits per second), a bit is considered as a signal pulse, and calculations are generally based on powers of

ten. Thus, a rate of one megabit per second is equal to one million bits per second. However, in certain technical contexts, megabit can also refer to 1,048,576 bits.

### **placeholder**

A symbol in a logical or mathematical expression that can be replaced by the name of any member of specified set.

## **Participles**

! **Study the following sentence:** *Assuming* that we are going to represent the concept of "raised to the power of" with the "^" symbol ...

**Present Participles (verb +ing)** describe what something or somebody is.

**Past Participles (verb +ed/V3)** describe how someone feels.

! Most **past participles** have **passive** meaning and can often be used as **adjectives** before nouns, or after **be** and other linking verbs, e.g. *measured bytes* = (bytes that have been measured, *accented symbol* = (symbol that have been accented)

! An **-ing/ed** forms with an object can be used as an adjective. Note the word order, e.g. *contact-logging tool*, *a room-sized computer*

! Used like adjectives or adverbs have similar meaning to **active** verbs, e.g. *floating databases* = (databases that float)

! We often use participles after nouns in order **to define** or identify the nouns, in the same way as we use identifying relative clauses, e.g. *We couldn't agree on any of the trials presented.* = (trials that were presented)

! A few **past participles** can be used as adjectives with **active** meaning, especially before nouns, e.g. *advanced agrotechnics* = (agrotechnics that have advanced to a high level, developed economy)

### **7. Find and underline other examples of participles in the text.**

### **8. Underline the correct item.**

- 1) The new program was exciting / excited.
- 2) What happens to your discarded / discarding old computer?

- 3) Bytes and bits are the starting / started point of the computer world.
- 4) Americans own billions of electronic products, including / included 200 million computers.
- 5) E-waste also includes cell phones, DVD players, video cameras and answering / answered machines.
- 6) Is what we're recycling / recycled actually getting recycled / recycling?
- 7) The "ray" is a stream of electrons generated / generating by an electron gun that naturally pour off a heated /heating cathode into the vacuum.
- 8) Random access memory is the most well-known form of computer memory, because it determines how quickly and efficiently your computer can perform your requested / requesting tasks.
- 9) There are several types of computer memory that are arranged based / basing on both technical and financial concerns.
- 10) There are several types of computer memory that are arranged based / basing on both technical and financial concerns.

### **9. Find the mistakes and correct the sentence.**

- 1) By reversed the direction of the current, the drive magnetizes particles in the metallic coating on a disk so the particles' north and south poles are oriented to create combinations represented a 0 ar a 1 bit.
- 2) After subtracted the overhead for packet packaging, the basic PCI-Express slot has a top bandwidth of 250 megabytes a second.
- 3) A lowercase "g" and "q" can both contain closing loops whose distinctions are too subtle far a computer to recognize.
- 4) It also performs the same CRC operation that was performed on the packet before its journey and compares its result to the earlier one bundling into the packet.
- 5) What is really happening is actually digital -- representing by discrete, separate, and different numerical values.

### **10. Translate the following sentences into Russian.**

- 1) After an image processor in the camera massages the visual data passed on by the sensor, the processor converts the image to a system of numerical values that represent different colors and hues.
- 2) Some OCR programs also display a magnified bitmap onscreen and ask you to press the key of the character needed to substitute for the placeholder character.
- 3) A capital "A" is 65. A capital "B" is 66. A lowercase "b" is 98.
- 4) Their ability to move data is indicated by the multiplier factor in their designations.
- 5) In machine language, the hexadecimal code 40 increases by one the value contained in the register named AX.

### 11. Translate the following sentences into English.

1. Лейбниц описал современную двоичную систему счисления и в 1679 году он представил модель своей бинарной арифметики.
2. Коды представляют символы, знаки препинания и движения принтера такие как табуляции, возврат каретки, перевод строки и прогон страницы являющийся управляющим символом, используемым при печати на принтере для перехода к новой странице.
3. Верхний и нижний регистры имеют разный набор ASCII-символов.
4. CD диск вмещает 650 мегабайтов данных.
5. Формы и цвета, несомненно, являются такой же информацией, как и слова.

## SPEAKING SECTION

### 12. Answer the following questions.

1. What are bits and bytes?
2. Why do computers use the base-2 system?
3. How can binary code represent so many more numbers than just 0 and 1?
4. When someone says, "This computer has a 2 gig hard drive," what does he or she mean?
5. How could you possibly need 2 gigabytes of space?
6. When you start talking about lots of bytes, what prefixes do you get into?

### 13. Topics for discussion.

1. Working with bits and bytes.
2. Undesired Fonts used for *ASCII* Text.



14. Prepare a presentation on the topic being discussed.

## REFERENCES, USEFUL LINKS AND FURTHER READING

1. Brain, Marshall. "How Bits and Bytes Work", April 2000. <http://www.howstuffworks.com/>
2. How computer works', by Ron White, 2008, pp.8-9.

## Unit 4

### MICROSOFT OFFICE

#### VOCABULARY

#### 1. Match the words with their definitions:

1) acolyte(n.)	['æk(ə)laɪt]	a) frighten or overawe (someone), especially in order to make them do what one wants
2) hobbyist(n.)	['hɒbiɪst]	b) cause (an event or situation, typically one that is undesirable) to happen suddenly, unexpectedly, or prematurely
3) resemble (v.)	['rɪ'zembəl]	c) fashionable and exciting
4) conquest(n.)	['kɒŋkwɛst]	d) an authority that is given by an organization to someone, allowing them to sell its goods or services which the organization controls
5) DOS	[dɒs]	e) involve (someone) deeply in an argument, conflict, or difficult situation
6) groovy (adj.)	['gru:vɪ]	f) like a very interested person in a particular thing and spends a lot of time on it
7) outfox (v.) [with obj.]	[,aʊt'fɒks]	g) no longer produced or used; out of date
8) bully (v.)	['bulɪ]	h) to affect very strongly (you do not know how to deal with it)
9) franchise (n.)	['frɑn(t)ʃaɪz]	i) the subjugation and assumption of control of something
10) embroil (v.) [with obj.]	[ɪm'brɔɪl], [em'brɔɪl]	j) the part of a computer operating system that controls and manages files and programs stored on disk
11) precipitate (v.)	[prɪ'sɪpɪteɪt]	k) have qualities or features, esp. those of appearance, in common with (someone or something); look or seem
12) obsolete (adj.)	['ɒbs(ə)li:t]	l) defeat (someone) by being more clever or cunning than them
13) overwhelm (v.)	[,əʊvə'welɪm]	m) a follower or assistant of an important person

## WATCHING

2. Microsoft Office 2010 brings more impact, energy and efficiency to every project. Watch a video ‘Microsoft office 2010 Beta Release’ and discuss with your partner if it is really an advantage.

## BEFORE YOU READ

3. Discuss with your partner the following question.

- So Microsoft rules the world, but just what does it do, anyway?

4. Skim the text to check your ideas.

## READING

### MICROSOFT SOFTWARE SUIT

By John Postley



is a business empire without equal. Its products are used in nearly every computer on the planet, and it has yet to reach the height of its power and influence. Microsoft has become a symbol of all that is great about the information age, as well as all that is less-than-great about it. In this article, we will look at Microsoft organization and its products to find out how they

benefit from the company's core advantages.

Bill Gates and Paul Allen founded Microsoft in 1975 as the classic "garage start-up." They were part of the first leap towards personal computing with a difference. While Apple founders Steve Jobs and Steve Wozniak built both a machine and software (the computer code that gives the computer the ability to do the things it does) in it, Microsoft only developed software.

Another key difference between Microsoft and other "garage start-up" computer companies was that neither Gates nor Allen held illusions about the business world. Many of their peers were *hobbyists* and *acolytes* of the "Home Brew Computer Club," which *resembled* a 1960s era "rap session" instead of a business association. Gates and Allen (along with Steve Balmer) were running a commercial enterprise in a competitive market. If there was a competing product, then they had two choices: buy it or destroy it. Their first *conquest* was also their biggest: DOS.

To make a long and dramatic story short, IBM was building its first personal computer and needed an operating system to run it. The company approached Microsoft, thinking that it had one available. But Microsoft didn't make operating systems -- they just wrote computer languages. Bill Gates directed them to the

makers of **CPM**, which was the reigning OS at the time. The culture clash between the IBM suits and the *groovy* style of the CPM manufacturers threatened to send the IBM PC into the dustbin of history...until Microsoft came to the rescue.

In what would become a typical move for the company, Gates and Balmer bought an OS for \$50,000, which they in turn licensed to IBM for \$80,000. Even in the early 1980s, \$80,000 wasn't very much money. So what was Gates thinking? It turns out that he was thinking far ahead. He said as much when interviewed for the PBS series "Triumph of the Nerds": "the key to our...deal was that IBM had no control over...our licensing to other people."

Microsoft realized that the IBM PC was going to create a mass market for personal computers. Gates gambled that the business cycle would follow the main-frame model and spawn clones. Out of necessity, these clones would be obliged to pay Microsoft any price to use DOS, which meant that the company was the gate-keeper that PC makers had to pay to compete in the personal computing world. It worked. From that moment on, Microsoft would *out-fox, bully*, best and co-opt all comers. It was simply a matter of finding new ways to reproduce the business model and the revenue it generated. Microsoft's shift from **DOS** to the GUI-based Windows operating system completed the cycle.

Apple's world-changing Macintosh hit the market in 1984 and was the first commercially successful PC to use a GUI, making it user friendly. We use GUIs to interact with computers and networks today. Bill Gates immediately saw the Mac as both a danger and an opportunity. The GUI could popularize computing on a level beyond the mass market created by the IBM PC and its clones, making it a threat to Microsoft's *franchise*. It also presented an opportunity to help Microsoft finish off its current opponent, IBM, and ensure its supremacy for years to come.

IBM and Microsoft had been *embroiled* in a battle over the PC's next operating system. IBM was collaborating with Microsoft on OS/2, a successor to DOS. IBM needed something other hardware makers couldn't clone. Microsoft played along until it became clear that the companies were at cross-purposes: Microsoft was trying to exploit IBM's market position and IBM was trying to put the DOS-genie back in the bottle. This *precipitated* a break, and Microsoft was on its own. If Microsoft could create its own **GUI** to work on top of DOS, then Bill Gates was killing two birds with one OS. The impact of Windows from a business perspective was as strong as the Mac's impact from a user's perspective. Almost overnight, a boring, *obsolete* PC could become Mac-like at a tenth of the cost of buying a Mac.

Bill Gates had again adapted to re-define the computer market, and just in time. Just before Windows 95 launched, a company called Netscape went public. Microsoft also had to deal with another oncoming threat: the Internet. Microsoft said -- we'll give our stuff away too. The "browser wars" didn't last long and were particularly one-sided, with Internet Explorer ultimately triumphing over Netscape. Microsoft have the market position and the cash on hand to exploit the market to their advantage.

Organization may be the most important key to Microsoft's success. Bill Gates never really consolidated the company after its start-up phase. Instead, he sought to replicate the most productive aspects of Microsoft's first team. Microsoft designs its teams to be overworked. Project managers calculate the number of staff members needed to accomplish a task, then reduce it. This results in a team that has to scramble immediately or be *overwhelmed*.

Microsoft also takes great care to choose the "right" people. The take-away is that Microsoft has very specific criteria their workers must fulfill, and they have developed an innovative means of assessing candidates: They ask them riddles. A company built on dominating via improvisation and high-energy needs high-energy problem solvers who don't mind working 72 hours straight. Although this type of employee seems difficult to find, somehow Microsoft keeps finding them. In the meantime, let's look at what Microsoft has in store for us.

In their annual report, Microsoft boils itself down into "Business Units," or "Product Segments," which represent its main product line: Client, Server & Tools, Information Worker, Microsoft Business Solutions, MSN, Mobile & Embedded Devices, and Home & Entertainment.

Whether you love it or hate it, you can't deny that Microsoft has had a sweeping impact. At the emergence of the computer age, the company got a head start by developing the OS for IBM's PC in 1981. Since then, Microsoft's ability to adapt to new developments and challenges has kept it at the top of the industry.

In the race to develop software for the non-techie community, Microsoft used its OS to dominate the market -- specifically, its Office suite of applications for word processing (Microsoft Word) and spreadsheets (Microsoft Excel). What's more, any other company that wants to develop software that's compatible with the OS has to pay royalties (licensing fees).

If you've ever used a word processing program, spreadsheet application or graphic design software, you've had some experience with productivity software. Although there are hundreds of examples of productivity software available, there's little doubt that Microsoft Office dominates the industry. In fact, more than 86 percent of all computers in homes and offices run on some form of the Windows OS.



Microsoft Office is often referred to as Microsoft Office Suite. It is a widely used group of interconnected desktop applications designed for both Windows and Mac computer systems. It includes Microsoft Word, Access, Publisher, PowerPoint, and Outlook.

The first Microsoft Office was released in 1990, and numerous versions have been designed since then. Microsoft Office Suite consists of applications for word processing, spreadsheets, databases, webpage creation, desktop publishing, contact management and many others.

There are five types of the Microsoft Office Suites. The most basic and least expensive suite is the Home and Student Suite. This includes Word, PowerPoint, Excel, and OneNote. The next step up is the Standard Suite which includes Word,

PowerPoint, Excel, and Outlook. The Small Business Suite includes Word, PowerPoint, Excel, Outlook, Accounting Express and Publisher. The Professional Suite includes Word, PowerPoint, Excel, Outlook, Access, Accounting Express and Publisher. To get all of the applications, you can purchase the Ultimate Suite.

Each program serves a different purpose and is compatible with other programs included in the package.



**Microsoft Word** is the most widely known application included in Microsoft Office Suite. Microsoft Word is a word processing application that includes a variety of tools to help create documents. In short, it:

- Gives your words some added flair.
- Edits and applies effects to photos without additional software.
- Organizes your documents and navigates easily with the new navigation pane.
- Adds powerful special effects to words with new text effects.
- Edits a document with other authors at the same time.



**Microsoft Access** is an **RDBMS**. Any database stores information in tables. With a Relational database like Access, "related" data is stored in many separate tables and linked together. For example, one table in the database might contain a list of customers, while another table might list the orders any customer has made. The link between those two tables would be a customer number that appears in both. In short, it:

- Takes control of your data.
- Makes exploring your data easy using Access drag-and-drop navigation forms.
- Creates insightful reports with new design, layout, and data analysis tools.
- More easily automates common tasks using the revamped Macro Designer.



**Microsoft Excel** is another popular application which is used to create spreadsheets and includes tools to help analyze the data. Software developed and manufactured by Microsoft Corporation that allows users to organize, format, and calculate data with formulas using a spreadsheet system broken up by rows and columns. Microsoft Excel usually comes bundled with Microsoft Office and is compatible with other applications offered in the suite of products. The first software program similar to Excel was released in 1982 and was called Multiplan. In short, it:

- Sees data more clearly.
- Highlights important data trends in seconds using Sparklines.
- Clarifies information with color schemes and data bars in Conditional Formatting.
- Spends less time shifting through data with new PowerPivot features.

- Edits documents with others simultaneously using the new Excel Web Apps.



**Office Web Apps.** For business users who use SharePoint servers, this is a quick editing tool for an existing PowerPoint presentation. In short, you can:

- Access, edit, and share documents
- Post documents online right from Word, Excel, PowerPoint, or OneNote.
- Access and share with others from virtually anywhere.
- Edit and work together online with new Office Web Apps.



In addition to creating slide show presentations, **Microsoft PowerPoint** can be used to create interactive presentations that have an impact. In short, you can:

- Embed, trim, add bookmarks and special effects to videos right in PowerPoint.
- Apply sophisticated effects to pictures, like color saturation and artistic filters.
- Broadcast presentation instantly by creating a link for real-time, online viewing.
- Share with others and stay on top of changes using the Co-Authoring feature.



**Microsoft Outlook** is an email client and calendar tool. It works with Microsoft Exchange Server but can also be set up to use POP or **IMAP**. Microsoft Outlook contains task management tools that help keep you on track and also allows you to keep an eye on the progress of others. In short, you can:

- Manage e-mail and calendars with ease.
- Streamline your Inbox with Conversation View and Ignore features.
- Manage multiple e-mail accounts, like Hotmail or Gmail, right from Outlook.
- Make scheduling simple with new Calendar Preview and Multiple Calendar View.
- Stay connected with social and business network updates using new Outlook Social Connector.



**Microsoft OneNote** helps you to:

- Organize notes in one digital notebook
- Capture and store text, images, videos, and audio notes in one easy-to-find place.
- Link notes back to their source with Linked Notes.
- Share and edit notes together using new Version Control tools.
- Make edits with others at the same time using the new OneNote Web App.



**Publisher** helps you to:

- Create and share professional-quality publications
- Edit pictures easily by zooming, panning, and cropping while maintaining your document’s layout.
- Align shapes, images, and text intuitively thanks to new alignment guidance technology.
- Save time and have more control in the new combined print and print-preview experience.

## LANGUAGE DEVELOPMENT

**5. Find and learn Russian equivalents for the following words and expressions:**

1) core advantages	a)
2) spawn clones	b)
3) Graphical User Interface	c)
4) embedded device	d)
5) sweeping impact	e)
6) non-techie community	f)
7) pay royalties	g)
8) desktop application	h)
9) revamped Macro Designer	i)

**6. Find and learn English equivalents for the following words and expressions:**

1) дисковая операционная система	a)
2) оценивать кандидатов	b)
3) заключать сделки вопреки	c)
4) использование электронной таблицы	d)
5) программное обеспечение для графического моделирования	e)
6) инструментальная панель	f)
7) рынок товаров массового производства	g)
8) детальный отчет	h)
9) условное форматирование	i)
10) функция графического интерфейса для работы с экранными объектами в среде Windows (перетаскивание)	j)

## 7. Find and learn the definitions for the following abbreviations.

1) RDBMS	a)
2) PC	b)
3) OS	c)
4) DOS	d)
5) CPM	e)
6) IBM	f)
7) GUI	g)
8) OS/2	h)
9) POP	i)
10) IMAP	j)

### KEY CONCEPTS

#### **CP/M**

A single user operating system for the 8080 and Z80 microprocessors. Created by Gary Kildall of Digital Research, CP/M had its heyday in the early 1980s, and although unsophisticated, was a major contributor to the personal computer revolution. However, CP/M computer vendors never standardized a floppy disk or screen format, requiring software publishers to create and stock multiple CP/M versions. After entering this uncontrolled market in 1981, IBM readily set the personal computer standards with its PC, and some of those standards remain to this day.

#### **DOS**

An earlier single-user operating system from Microsoft for the PC. It was the first operating system for IBM and IBM-compatible PCs and remained the underlying control program for Windows 3.1, 95, 98 and ME. In order to support DOS applications, Windows NT, 2000, XP and Vista include their

own version of DOS, called "DOS emulation."

#### **GUI**

A method of controlling software using onscreen icons, menus, dialog boxes, and objects that can be moved or resized, usually with a pointing device such as a mouse.

#### **IMAP**

A protocol allowing a client to access and manipulate electronic mail messages on a server. It permits manipulation of remote message folders (mailboxes), in a way that is functionally equivalent to local mailboxes. IMAP includes operations for creating, deleting, and renaming mailboxes; checking for new messages; permanently removing messages; searching; and selective fetching of message attributes, texts, and portions thereof. It does not specify a means of posting mail; this function is handled by a mail transfer protocol such as SMTP.

## RDBMS

System for quick search and retrieval of information from a database. The DBMS determines how data are stored and retrieved. It must address problems such as security, accuracy, consistency among different records, response time, and memory requirements. These issues are most significant for database systems on computer networks. Ever-higher processing speeds are required for efficient database management. Relational DBMSs, in which data are organized into a series of tables (“relations”) that are easily reorganized for accessing data in different ways, are the most widely used today.

## SMTP

The standard e-mail protocol on the Internet and part of the TCP/IP protocol suite, as defined by IETF RFC 2821. SMTP defines the message format and the message transfer agent (MTA), which stores and forwards the mail. SMTP was originally designed for only plain text (ASCII text), but MIME and other encoding methods enable executable

programs and multimedia files to be attached to and transported with the e-mail message. SMTP servers route SMTP messages throughout the Internet to a mail server that provides a message store for incoming mail. The mail server uses the POP3 or IMAP4 access protocol to communicate with the user's e-mail program.

## IDIOMS

### buck the trend

To perform well when the market as a whole is doing poorly, or, rarely to perform badly when the market as a whole is doing well. Bucking the trend may apply to individual securities or to whole industries. In technical analysis, bucking the trend is often seen as a bullish signal, as it indicates that investors are still interested in the security or industry involved despite the downturn in the rest of the market.

### ! Study the following:

We can use *Neither* and *nor...* as adverbs to mean “also not”. They come at the beginning of a clause, and are followed by inverted word order: *auxiliary verb + subject*.

*Neither... nor...* structure is used to join two negative ideas. It is the opposite of *both...and...* It is usually rather formal. When singular subjects are connected by *Neither* and *nor...*, the verb is usually singular.

**8. Find the example of this structure in the text and translate the sentence.**

**! Study the following:**

It is difficult to learn to use prepositions correctly in a foreign language. It is not always easy to know which preposition to use after a particular noun, verb or adjective. Most English prepositions have several different functions, and these may correspond to several different prepositions in another language. Often the correct preposition cannot be guessed, and one has to learn the expression as a whole.

**9. Complete the following sentences with the right preposition.**

1) Have a look \_\_ the history of Microsoft and find \_\_ how it competes \_\_ the market.

2) Bill Gates directed them \_\_ the makers of CPM, which was the reigning OS \_\_ the time.

3) Microsoft also had to deal \_\_ another oncoming threat: the Internet.

4) Microsoft had a lot \_\_ cash \_\_ hand to exploit the market \_\_ their advantage.

5) All empires fall \_\_ some point, and organizations atrophy like muscles \_\_ old age.

6) Bill Gates is not likely to buck that trend, so you can count \_\_ Microsoft's reign to extend \_\_ years to come.

7) Since then, Microsoft's ability to adapt \_\_ new developments and challenges has kept it \_\_ the top \_\_ the industry.

8) Microsoft Excel usually comes bundled \_\_ Microsoft Office and is compatible \_\_ other applications offered \_\_ the suite \_\_ products.

**10. Translate the following sentences into Russian.**

1) They represented the first graphic user interface (GUI), a term reserved today for the way that Windows and Macintosh systems display.

2) Selecting Windows Projects will show you a list of project types that can be created for desktop applications, including Windows Presentation Foundation (WPF), Windows Forms, and Console programs.

3) With the revamped Macro Designer, it's now even easier for you to add basic logic to your database.

4) The computer's operating system, beginning with DOS and continuing with Windows and every other OS, determines the scheme.

5) Both DOS and CPM-86 were more concerned with providing a learning path that would be familiar to people who already knew CP/M, the original computer geeks.

### **11. Translate the following sentences into English.**

1) Встроенные устройства – это маленькие программные чипы, которые могут быть запрограммированы на выполнение определенных заданий.

2) После лицензирования своего продукта изобретатель получает деньги путем лицензионных платежей от лицензедержателя/лицензиата.

3) Электронная Интернет почта использует два основных стандарта: простой протокол или SMTP для отправки сообщений, и почтовый протокол или POP для получения сообщений.

4) Существенное преимущество – это когда никто не может сделать подобное тому, что ты делаешь.

5) NT была более стабильной версией Windows, но она не могла запускать приложения, разработанные для старых процессоров.

## **SPEAKING SECTION**

### **12. Answer the following questions.**

1. How would Microsoft deal with an open global network, seemingly powered by altruistic creators bent on giving their products away?

2. Why Microsoft broke with IBM?

3. What Is Microsoft Office Suite?

4. What is the BEST way to find out what version of Office you're using?

5. If you needed to create a slide presentation, which Office program would you use?

6. What VBA stands for?

7. In Access, you've been working on your database a lot lately and you notice it's getting quite large. What's the best thing to do to be sure it's at it's most efficient size?

8. If you wanted to create your own new toolbar in an Office application, with all your favorite features, what would you do first?

9. What Excel can also be used to do?

### **13. Topics for discussion.**

1. MS Excel as a spreadsheet and analysis program released by Microsoft.

2. Outlook 2010 Cleanup Tools.

3. Protecting a Word document.



14. Prepare a presentation on the topic being discussed.

## REFERENCES, USEFUL LINKS AND FURTHER READING

1. How computer works', by Ron White, 2008, pp.135, 165, 339.
2. Joe Mayo, Microsoft Visual Studio 2010. A beginner's Guide. p.34.

## UNIT 5

### ELECTRONIC DICTIONARIES & MACHINE TRANSLATION

#### VOCABULARY

##### 1. Match the words with their definitions:

1) parse (v.) [with obj.]	['pɑ:z]	a) reliability
2) incrementally (adv.)	[,ɪŋkrɪ'ment(ə)li]	b) a large collection of texts
3) stem from (v.)	['stemfrəm]	c) single-stepwise; in a series of distinct stages; not continuously
4) compliance (n.)	[kəm'plaiənt(s)]	d) resolve (a sentence) into its component parts and describe their syntactic roles
5) robustness (n.)	[rə'bʌstnəs]	e) not behaving in the same way
6) inconsistency (n.)	[,ɪnkən'sɪst(ə)n(t)sɪ]	f) compatibility
7) corpus (n.) corpora (pl.)	['kɔ:pəs] ['kɔ:p(ə)rə]	g) to be the result of smth

## WATCHING

2. Machine translation is automated translation. It is the process by which computer software is used to translate a text from one natural language to another one. Watch a video ‘Translate with Google’ and discuss with your partner how it exactly works.

## BEFORE YOU READ

3. Discuss with your partner the following question.
  - What are people using the machine translation system for?
4. Skim the text to check your ideas.

## READING

### WHAT IS MACHINE TRANSLATION?

The history of MT began in 1666, when Leibniz in his dissertation, *The Art of Combinations*, had clearly indicated the possible mechanization of both arithmetic and thought processes, that his logical processor could be used to transform one language into another. He also considered the extraction of ideas from text and their expression in terms of a metalanguage (Saw 1954). The culmination occurred in his design of the first practicable mechanical multiplying machine (1694) and in a binary multiplier (Eriksson et al. 1996). On June 20, 1946, New York the discussion between Weaver and A.D. Booth identified the fact that the code-breaking process in no way resembled language translation because it was known a priori that the decrypting process must result in a unique output. The main purpose of this meeting, however, was to interest the Rockefeller Foundation

in supporting development of an electronic computer at the University of London. In March 1947, Warren Weaver, director of the Division of Natural Sciences of the Rockefeller Foundation, in correspondence with Andrew D. Booth and Norbert Wiener first formulated the concept of machine translation.

The earliest “translation engines” were based on a direct, so called “Transformer” approach. Input



sentences of the source language were transformed directly into output sentences of the target language. At first the machine did a rough analysis of the source sentence dividing it into subject-object-verb, etc. Then source words were replaced by target words selected from a dictionary and their order was rearranged according to the rules of the target language. These rough operations were resulted in a simplified transformation with lots of silly sentences so much laughed at now. To process any translation, human or automated, the meaning of a text in the original (source) language must be fully restored in the target language, i.e. the translation. While on the surface this seems straightforward, it is far more complex. Translation is not a mere word-for-word substitution. A translator must interpret and analyze all of the elements in the text and know how each word may influence another. This requires extensive expertise in grammar, syntax (sentence structure), semantics (meanings), etc., in the source and target languages, as well as familiarity with each local region. Human and machine translation each have their share of challenges. For example, no two individual translators can produce identical translations of the same text in the same language pair, and it may take several rounds of revisions to meet customer satisfaction. But the greater challenge lies in how machine translation can produce publishable quality translations.

The ideal aim of machine translation systems is to produce the best possible translation without human assistance. Basically every machine translation system requires programs for translation and automated dictionaries and grammars to support translation. Machine translation systems that produce translations between only two particular languages are called bilingual systems and those that produce translations for any given pair of languages are called multilingual systems. Multilingual systems may be either uni-directional or bi-directional. Multilingual systems are preferred to be bi-directional and bi-lingual as they have ability to translate from any given language to any other given language and vice versa.

If talking about rule-based machine translation, it relies on countless built-in linguistic rules and millions of bilingual dictionaries for each language pair. The software parses text and creates a transitional representation from which the text in the target language is generated. This process requires extensive lexicons with morphological, syntactic, and semantic information, and large sets of rules. The software uses these complex rule sets and then transfers the grammatical structure of the source language into the target language. Translations are built on gigantic dictionaries and sophisticated linguistic rules. Users can improve the out-of-the-box translation quality by adding their terminology into the translation process. They create user-defined dictionaries which override the system's default settings. In most cases, there are two steps: an initial investment that significantly increases the quality at a limited cost, and an ongoing investment to increase quality incrementally. While rule-based MT brings companies to the quality threshold and beyond, the quality improvement process may be long and expensive.

Statistical machine translation utilizes statistical translation models whose parameters stem from the analysis of monolingual and bilingual corpora. Building

statistical translation models is a quick process, but the technology relies heavily on existing multilingual corpora. A minimum of 2 million words for a specific domain and even more for general language are required. Theoretically it is possible to reach the quality threshold but most companies do not have such large amounts of existing multilingual corpora to build the necessary translation models. Additionally, statistical machine translation is CPU intensive and requires an extensive hardware configuration to run translation models for average performance levels. Rule-based MT provides good out-of-domain quality and is by nature predictable. Dictionary-based customization guarantees improved quality and compliance with corporate terminology. But translation results may lack the fluency readers expect. In terms of investment, the customization cycle needed to reach the quality threshold can be long and costly. The performance is high even on standard hardware. Statistical MT provides good quality when large and qualified corpora are available. The translation is fluent, meaning it reads well and therefore meets user expectations. However, the translation is neither predictable nor consistent. Training from good corpora is automated and cheaper. But training on general language corpora, meaning text other than the specified domain, is poor. Furthermore, statistical MT requires significant hardware to build and manage large translation models.

<b>Rule-Based MT</b>	<b>Statistical MT</b>
+ Consistent and predictable quality	- Unpredictable translation quality
+ Out-of-domain translation quality	- Poor out-of-domain quality
+ Knows grammatical rules	- Does not know grammar
+ High performance and <i>robustness</i>	- High CPU and disk space requirements
+ Consistency between versions	- <i>Inconsistency</i> between versions
- Lack of fluency	+ Good fluency
- Hard to handle exceptions to rules	+ Good for catching exceptions to rules
- High development and customization costs	+ Rapid and cost-effective development costs provided the required corpus exists

Given the overall requirements, there is a clear need for a third approach through which users would reach better translation quality and high performance (similar to rule-based MT), with less investment (similar to statistical MT). <http://www.systran.co.uk/>

## LANGUAGE DEVELOPMENT

**5. Find and learn Russian equivalents for the following words and expressions:**

1) rule-based	a)
2) user-defined	b)
3) statistical machine translation	c)
4) bilingual corpora	d)
5) source language	e)
6) target language	f)
7) quality threshold	g)
8) binary multiplier	h)
9) decrypting process	i)

**6. Find and learn English equivalents for the following words and expressions:**

1) необычный, исключительный	a)
2) установки (параметров) по умолчанию	b)
3) устойчивость системы	c)
4) качество внешней среды (в которой должна работать программа)	d)
5) обеспечение словаря, основанное на конкретных требованиях отдельного клиента	e)
6) встроенные грамматические правила	f)
7) дословная замена (подстановка)	g)
8) процесс дешифрации сообщений	h)

## KEY CONCEPTS

### **metalinguage**

A form of language or set of terms used for the description or analysis of another language.

### **rule-based (logic programming)**

The study or implementation of computer programs capable of dis-

covering or checking proofs of formal expressions or segments.

### **bilingual/parallel corpus**

It is a corpus that contains a collection of original texts in language  $L_1$  and their translations into a set of languages  $L_2 \dots L_n$ . In most cases,

parallel corpora contain data from only two languages. Closely related to parallel corpora are 'comparable corpora', which consists of texts from two or more languages which are similar in genre, topic, register etc. without, however, containing the same content. Parallel corpora can be bilingual or multilingual, i.e. they consist of texts of two or more languages. They can be either unidirectional (e.g. an English text translated into German), bidirectional (e.g. an English text translated into German and vice versa), or multidirectional

(e.g. an English text such as an EU regulation translated into German, Spanish, French, etc.).

**target language**

The computer language, often a machine language, into which a document written in another computer language is to be translated. **source language**

The language from which a translation is to be made or from which a word is borrowed.

**! Study the following:** *Translation is not a mere word-for-word substitution.*

The important thing to remember in a compound noun (i.e. two nouns put together) is that the second noun is the real noun. The first is used like an adjective to describe what kind of thing or person the second noun is. There is no simple rule to tell you when the two parts of a compound noun are written together, with a hyphen or separately. You must learn the written form every time you learn a new compound noun.

**7. Find, underline and translate the other examples in the text.**

**8. Make compound nouns from the following words and phrases. Use the text above if necessary.**

1) a programme on computer	a)
2) translation carried out by a computer	b)
3) when translated, using exactly equivalent words	c)
4) decoding	d)
5) a machine for automatical calculations	e)
6) a language used to define another language	f)
7) a language from which a translation is to be made	g)
8) a grammatical arrangement of words in sentences	h)

9) a language into which a translation is to be made	i)
10) source language and target language	j)
11) approach based on two languages	k)
12) the programs and other operating information used by a computer	l)
13) models for translation	m)
14) tools, machinery, and other durable equipment for computer	n)

**9. Find the mistakes and correct the sentence.**

1. This website is dedicated to research in statistical translation machine.
2. Breaking code is an exciting logic game for kids and adults.
3. The term Meta language was originally used by linguist Roman Jakobson and other Russian Formalists to characterize a language that makes assertions about other languages.
4. Structure sentence is the order of the elements in a group of words that expresses a complete thought.
5. Synonyms for word-of-word at Thesaurus.com with free online thesaurus, antonyms, and definitions.

**10. Translate the following sentences into Russian.**

- 1) A User-Defined Function, or UDF, is a function provided by the user of a program or environment, in a context where the usual assumption is that functions are built into the program or environment.
- 2) The tool employs well-defined linguistic rules and comprises several linguistically motivated phases.
- 3) Through experiments in 2 domains and 8 language pairs it is shown that the out-of domain improves coverage and translation of rare words, but may degrade the translation quality for more common words.
- 4) The tool also contains built-in linguistic rules for analyzing English sentences.
- 5) Robustness is something that should be designed into software from the ground up; it is not something that can be successfully tacked on at a later date.

**11. Translate the following sentences into English.**

- 1) Для того, чтобы изменить установки параметров по умолчанию, сделайте следующие шаги...
- 2) Процедура шифрования и дешифрования документов напрямую связана с этой умозрительной моделью.
- 3) Это алгоритм требует априорное описание порогового интервала в пределах кластера и минимального количества элементов в каждом кластере.

4) Для того чтобы показать важность устойчивости системы в практическом применении, мы процитируем конкретный пример из работы Бен-Таля и Немировского (2000).

5) Данные внешней среды увеличивают границы и улучшают перевод редких слов, но могут ухудшить качество перевода общеупотребительной лексики.

## SPEAKING SECTION

### 12. Answer the following questions.

- 1) Why they say that the history of MT began in 1666?
- 2) What was the main purpose of the meeting between Weaver and A.D. Booth?
- 3) How did the earliest “translation engines” work?
- 4) What do we need to do to process any translation?
- 5) What is the difference between bilingual and multilingual systems?
- 6) What does rule-based machine translation rely on?
- 7) What does statistical machine translation utilize?
- 8) Are there any advantages or disadvantages of rule-based and statistical MTs?

### 13. Topics for discussion.

- 1) Machine Translation.
- 2) Electronic Dictionaries.
- 3) Internet Translation Engines.



14. Prepare a presentation on the topic being discussed.

## REFERENCES, USEFUL LINKS AND FURTHER READING

1. ‘How computer works’, by Ron White, 2008, pp.8-9.
2. Barry Haddow, Philipp Koehn. Analysing the effect of out-of-domain data on SMT systems.-Edinburgh.-2009.

## UNIT 6

### NUMERAL SYSTEMS

#### VOCABULARY

##### 1. Match the words with their definitions:

1) numeral (n.)	['nju:m(ə)r(ə)l]	a) (476-c.550), Indian astronomer and mathematician
2) integer (n.)	['intɪdʒə]	b) combine
3) recur (v.)	['rɪ'kɜ:]	c) an electromotive force or potential difference expressed in volts
4) Aryabhata I (n.)	[.arɪə'bɑ:tə]	d) relating to or based on the number twenty
5) unary	['ju:nəri]	e) consisting of or involving a single component or element
6) occurrence (n.)	[ə'kʌr(ə)n(t)s]	f) occur again periodically or repeatedly
7) voltage (n.)	[vɒʊltɪdʒ]	g) the action, fact, or instance of occurring
8) hexadecimal (adj.)	[.heksə'desɪm(ə)l]	h) a symbol or group of symbols used to express a number: for example, 6 ( <i>Arabic</i> ), VI ( <i>Roman</i> ), 110 ( <i>binary</i> )
9) vigesimal (adj.)	[vɪ'dʒesɪm(ə)l]	i) relating to or using a system of numerical notation that has 16 rather than 10 as its base
10) merge (v.)	[mɜ:dʒɪŋ]	j) a complete unit or entity

#### WATCHING

2. Watch a video 'Number Systems', discuss with your partner and find the true answer to the following question:

- Who were the true inventors of Number system?

#### BEFORE YOU READ

3. Discuss with your partner the following question.

- What are people using the Number system for?

4. Skim the text to check your ideas.

## READING

### DIFFERENT SYSTEMS OF NUMERATION

#### Numeral systems by culture

##### Hindu-Arabic numerals

Western Arabic	Indian family
Eastern Arabic	Brahmi
<u>Khmer</u>	<u>Thai</u>

##### East Asian numerals

Chinese	Korean
Chinese counting rods	Japanese

##### Alphabetic numerals

<u>Abjad</u>	<u>Hebrew</u>
Armenian	Ionian/Greek
Cyrillic	Sanskrit
<u>Ge'ez</u>	

##### Other systems

Attic	Babylonian
Etruscan	Egyptian
<u>Urnfield</u>	<u>Mayan</u>
<u>Roman</u>	

List of numeral system topics

#### Positional systems by base

##### Decimal (10)

2, 4, 8, 16, 32, 64

3, 9, 12, 24, 30, 36, 60, more...

- Give every number represented a unique representation (or at least a standard representation);
- Reflect the algebraic and arithmetic structure of the numbers.

For example, the usual decimal representation of whole numbers gives every whole number a unique representation as a finite sequence of digits, with the operations of arithmetic (addition, subtraction, multiplication and division) being present as the standard algorithms of arithmetic. However, when decimal representation is used for the rational or real numbers, the representation is no longer unique: many rational numbers have two numerals, a standard one that terminates, such as 2.31, and another that *recurs*, such as 2.309999999... . Numerals which terminate have no non-zero digits after a given position. For example, numerals like 2.31 and 2.310 are taken to be the

A *numeral* is a symbol or group of symbols, or a word in a natural language that represents a number. Numerals differ from numbers just as words differ from the things they refer to. The symbols "11", "eleven" and "XI" are different numerals, all representing the same number. This article attempts to explain the various systems of numerals.

A numeral system (or system of numeration) is a framework where a set of numbers are represented by numerals in a consistent manner. It can be seen as the context that allows the numeral "11" to be interpreted as the binary numeral for three, the decimal numeral for *eleven*, or other numbers in different bases.

Ideally, a numeral system will:

- Represent a useful set of numbers (e.g. all whole numbers, *integers*, or real numbers);

same, except in the experimental sciences, where greater precision is denoted by the trailing zero. Numeral systems are sometimes called number systems, but that name is misleading. Let's have a look at the types of numeral systems.

Nowadays, the most commonly used system of numerals is known as Hindu-Arabic numerals and two great Indian mathematicians could be given credit for developing them. *Aryabhata* of Kusumapura who lived during the 5th century developed the place value notation and Brahmagupta a century later introduced the symbol zero.

The simplest numeral system is the *unary* numeral system, in which every natural number is represented by a corresponding number of symbols. If the symbol ' is chosen, for example, then the number seven would be represented by '''''''. The unary system is normally only useful for small numbers. It has some uses in theoretical computer science. Elias gamma coding is commonly used in data compression; it includes a unary part and a binary part.

The unary notation can be abbreviated by introducing different symbols for certain new values. Very commonly, these values are powers of 10; so for instance, if ' stands for one, - for ten and + for 100, then the number 304 can be compactly represented as +++ '' and number 123 as + - '' without any need for zero. This is called sign-value notation. The ancient Egyptian system is of this type, and the Roman system is a modification of this idea.

More useful still are systems which employ special abbreviations for repetitions of symbols; for example, using the first nine letters of our alphabet for these abbreviations, with A standing for "one occurrence", B "two occurrences", and so on, we could then write C+ D' for the number 304. The numeral system of English is of this type ("three hundred [and] four"), as are those of virtually all other spoken languages, regardless of what written systems they have adopted.

More elegant is a positional system, also known as place-value notation. Again working in base 10, we use ten different digits 0, ..., 9 and use the position of a digit to signify the power of ten that the digit is to be multiplied with, as in  $304 = 3 \times 100 + 0 \times 10 + 4 \times 1$ . Note that zero, which is not needed in the other systems, is of crucial importance here, in order to be able to "skip" a power. The Hindu-Arabic numeral system, borrowed from India, is a positional base 10 system; it is used today throughout the world.

Arithmetic is much easier in positional systems than in the earlier additive ones; furthermore, additive systems have a need for a potentially infinite number of different symbols for the different powers of 10; positional systems need only 10 different symbols (assuming that it uses base 10). The numerals used when writing numbers with digits or symbols can be divided into two types that might be called the arithmetic numerals 0,1,2,3,4,5,6,7,8,9 and the geometric numerals 1,10,100,1000,10000... respectively. The sign-value systems use only the geometric numerals and the positional system use only the arithmetic numerals. The sign-value system does not need arithmetic numerals because they are made by repetition (except for the Ionic system), and the positional system does not need geo-

metric numerals because they are made by position. However, the spoken language uses both arithmetic and geometric numerals.

In certain areas of computer science, a modified base- $k$  positional system is used, called **bijective numeration**, with digits  $1, 2, \dots, k$  ( $k \geq 1$ ), and zero being represented by the empty string. This establishes a bijection between the set of all such digit-strings and the set of non-negative integers, avoiding the non-uniqueness caused by leading zeros. Bijective base- $k$  numeration is also called  $k$ -adic notation, not to be confused with  $p$ -adic numbers. Bijective base-1 is the same as unary.

Modern computers use transistors that represent two states with either high or low *voltages*. We have already learnt that the smallest unit of memory for this binary state is called a bit. Bits are arranged in groups to aid in processing, and to make the binary numbers shorter and more manageable for humans. More recently these groups of bits, such as bytes and words, are sized in multiples of four. Thus base 16 (*hexadecimal*) is commonly used as shorthand. Base 8 (octal) has also been used for this purpose. A computer does not treat all of its data as numerical. For instance, some of it may be treated as program instructions or data such as text. However, arithmetic and Boolean logic constitute most internal operations. Whole numbers are represented exactly, as integers. Real numbers, allowing fractional values, are usually approximated as floating point numbers. The computer uses different methods to do arithmetic with these two kinds of numbers. A base-8 system (octal) was devised by the Yuki of Northern California, who used the spaces between the fingers to count. The base-10 system (decimal) is the one most commonly used today. It is assumed to have originated because humans have ten fingers. These systems often use a larger superimposed base. Base-12 systems (duodecimal or dozenal) have been popular because multiplication and division are easier than in base-10, with addition just as easy. Base-12 could have originated from the number of knuckles in the four fingers of a hand excluding the thumb, which is used as a pointer in counting. Multiples of 12 have been in common use as English units of resolution in the analog and digital printing world, where 1 point equals  $1/72$  of an inch and 12 points equal 1 pica, and printer resolutions like 360, 600, 720, 1200 or 1440 dpi (dots per inch) are common. These are combinations of base-12 and base-10 factors:  $(3 \times 12) \times 10$ ,  $12 \times (5 \times 10)$ ,  $(6 \times 12) \times 10$ ,  $12 \times (10 \times 10)$  and  $(12 \times 12) \times 10$ .

The Maya civilization and other civilizations of Pre-Columbian Mesoamerica used base-20 (vigesimal), possibly originating from the number of a person's fingers and toes. Evidence of base-20 counting systems is also found in the languages of central and western Africa. Danish numerals display a similar base-20 structure. Base 60 (sexagesimal) was used by the Sumerians and their successors in Mesopotamia and survives today in our system of time (hence the division of an hour into 60 minutes and a minute into 60 seconds) and in our system of angular measure (a degree is divided into 60 minutes and a minute is divided into 60 seconds). Base-60 systems are believed to have originated through the *merging* of base-10 and base-12 systems. Many ancient counting systems use 5 as a primary

base, almost surely coming from the number of fingers on a person's hand. Often these systems are supplemented with a secondary base, sometimes ten, sometimes twenty. In some African languages the word for 5 is the same as "hand" or "fist" (Dyola language of Guinea-Bissau, Banda language of Central Africa). Counting continues by adding 1, 2, 3, or 4 to combinations of 5, until the secondary base is reached. In the case of twenty, this word often means "man complete". This system is referred to as *quinquavi* gesimal system. It is found in many languages of the Sudan region. (see APPENDIX and REFERENCES)

## LANGUAGE DEVELOPMENT

**5. Find and learn Russian equivalents for the following words and expressions:**

1) real number	a)
2) Elias gamma code	b)
3) floating point	c)
4) decimal representation	d)
5) data compression	e)
6) rational number	f)
7) pica	g)
8) trailing zero	h)
9) bijective numeration	i)

**6. Find and learn English equivalents for the following words and expressions:**

1) целое число	a)
2) позиционная система (счисления)	b)
3) система знакового обозначения	c)
4) натуральное число	d)
5) пустая строка	e)
6) арифметическое число	f)
7) цифровая строка	g)
8) натуральное число	h)

## KEY CONCEPTS

**Bijjective numeration**

It is any numeral system that establishes a bijection between the set of non-negative integers and *the set of finite strings over a finite set of digits*. In particular, bijjective base- $k$  numeration represents a non-negative integer by using a string of digits from the set  $\{1, 2, \dots, k\}$  ( $k \geq 1$ ) to encode the integer's expansion in powers of  $k$ .

**Gesimal system**

The system which is based on five and twenty.

**integer**

Any of the natural numbers (positive or negative) or zero; "an integer is a number that is not a fraction".

**Elias gamma code**

is a universal code encoding positive integers developed by Peter Elias. It is used most commonly when coding integers whose upper-bound cannot be determined beforehand.

**trailing zeros**

A sequence of 0s in the decimal representation (or more generally, in any positional representation) of a number, after which no other digits follow.

**floating point**

Describes a system for representing numbers

that would be too large or too small to be represented as integers. Numbers are in general represented approximately to a fixed number of significant digits and scaled using an exponent.

The base for the scaling is normally 2, 10 or 16. The typical number that can be represented exactly is of the form: *significant digits*  $\times$  *base*<sup>*exponent*</sup>

The term floating point refers to the fact that the radix point (decimal point, or, more commonly in computers, binary point) can "float"; that is, it can be placed anywhere relative to the significant digits of the number. This position is indicated separately in the internal representation, and floating-point representation can thus be thought of as a computer realization of scientific notation. Over the years, several different floating-point representations have been used in computers; however, for the last ten years the most commonly encountered representation is that defined by the IEEE 754 Standard. The speed of floating-point operations is an important measure of performance for computers in many application domains. It is measured in FLOPS.

**FLOPS (or flops or flop/s)**

An acronym meaning **F**loating point **O**perations per **S**econd. The FLOPS is a measure of a computer's performance, especially in fields of scientific calculations that make heavy use of floating point calculations, similar to the older, simpler, instructions per second. Since the final *S* stands for "second", conservative speakers consider "FLOPS" as both the singular and plural of the term, although the singular "FLOP" is frequently encountered.

**! Study the following:** *The papers were late and the post was too.*

If we want to say that people, things, actions or events are similar, we can use *as* or *like*; *so*, *neither do I* and similar structures; or adverbs such as *too*, *also*, and *as well*.

To say that they are identical, we can use *the same (as)*.

To say that people, things etc are equal in a particular way, we often use the structure *as (much/many)...as*.

To say that people, things etc are unequal in a particular way, we can use comparative adjectives and verbs, or *more (...than)* with adjectives, adverbs, verbs or nouns.

To say which one of a group is outstanding in a particular way, we can use *most*.

We can use double comparatives to say that something is changing. *...er and ...er/more and more....*

**! Study the following:** *The more illustrated it is, the more I like it.*

We can use comparatives with *the +comparative expression + subject + verb, the...* to say that things change or vary together, or that two variable quantities are systematically related.

**! Study the following:** *This experiment made it all the more important to continue the research.*

Another use of *the* meaning 'by that much' is in **all/any/none the +comparative**. This structure can be used only to express abstract ideas when we say something is or should be 'more...'

**! Study the following:** *It was ten times more difficult than we expected.*

Instead of *three/four* etc *times as much*, we can use **three/four etc times + comparative**.

**7. Find, underline and translate the other examples in the text.**

**8. Find the mistakes and correct the sentence.**

- 1) It is much difficult to measure changes, which affected counting styles.
- 2) This is all the most important as there has been very little research.
- 3) As an example, diamond is about 4-5 times the hardest than corundum.
- 4) Round off your answer so it has only as much significant digits the quantity, which contains the least number of significant digits.
- 5) Are double digits too much to expect?

**9. Translate the following sentences into Russian.**

- 1) The stored bits in the illustration represent the binary numeral 1011, which is 11 in decimal numbers.

2) The only data recorded is the delta-the differences that occur in the frames trailing the original reference frame.

3) When you work with decimal numbers, you need to understand the difference between trailing zeros and leading zeros, as well as how they affect the value of the number.

4) Computers are designed to manipulate digital data-specific values are expressed in real numbers.

5) When adjacent digits in a digit string are touching the recognition task becomes particularly challenging.

### **10. Translate the following sentences into English.**

1. В теории вычислительных машин и систем, а также в теории формального языка нульстрока представляет собой однозначную строку в нуль длины.

2. В информатике, плавающая запятая описывает способ представления и обработки действительных чисел.

3. Биективная, взаимно однозначная система счисления с основанием  $b$  использует различные числа  $b$  для того чтобы представить все натуральные числа.

4. Метод сжатия данных является одной из самых важных областей и инструментов современной информатики.

5. В математике, нулевой байт в конце строки – это последовательность 0 в десятичном представлении числа.

## **SPEAKING SECTION**

### **11. Answer the following questions.**

1. What is a numeral system?
2. Can you explain the reason when decimal representation is used for the rational or real numbers, the representation is no longer unique?
3. What kind of types of numerical systems you can name?
4. What is the most commonly used system of numerals nowadays? Why?
5. Is there any difference between positional and additive systems? If yes, what?

### **12. Topics for discussion.**

1. Numeral Systems.
2. The History of Numbers.
3. Ancient Counting Systems.



13. **Prepare a presentation on the topic being discussed.**

## REFERENCES, USEFUL LINKS AND FURTHER READING

1. 'How computer works', by Ron White, 2008, pp.8-9.
2. Georges Ifrah. The Universal History of Numbers : From Prehistory to the Invention of the Computer, Wiley, 1999. ISBN 0-471-37568-3.
3. Knuth. The Art of Computer Programming. Volume 2, 3rd Ed. Addison-Wesley. pp.194–213, "Positional Number Systems".
4. J.P. Mallory and D.Q. Adams, Encyclopedia of Indo-European Culture, Fitzroy Dearborn Publishers, London and Chicago, 1997.
5. Hans J. Nissen, P. Damerow, R. Englund, Archaic Bookkeeping, University of Chicago Press, 1993, ISBN 0-226-58659-6.
6. Denise Schmandt-Besserat, How Writing Came About, University of Texas Press, 1992, ISBN 0-292-77704-3.
7. Claudia Zaslavsky, Africa Counts: Number and Pattern in African Cultures, Lawrence Hill Books, 1999, ISBN 1-55652-350-5.

## UNIT 7

### LOGICAL OPERATIONS

#### Vocabulary

##### 1. Match the words with their definitions:

1) apparent (adj.)	[ə'pær(ə)nt]	a) unacceptably or very shockingly
2) outrageously (adv.)	[,aʊ'treɪdʒəsli]	b) the modification or control of a process or system by its results or effects
3) implementation(n.)	[,ɪmplɪmen'teɪʃ(ə)n]	c) clearly visible or understood; obvious
4) pesky (adj.)	['peski]	d) annoying, causing trouble
5) feedback (n.)	['fi:dbæk]	e) the act of accomplishing some aim or executing some order

#### WATCHING

2. Watch this 'Boolean operators' clip, discuss with your partner and find the true answer to the following question:

- How to combine search words and get the results you are looking for?

#### BEFORE YOU READ

3. Discuss with your partner the following question.

- What are people using the Boolean operators for?

4. Skim the text 1 to check your ideas.

#### READING

### BOOLEAN LOGIC (1)



Have you ever wondered how a computer can do something like balance a check book, or play chess, or spell-check a document? These are things that, just a few decades ago, only humans could do. Now computers do them with *apparent* ease. How can a "chip" made up of silicon and wires do

something that seems like it requires human thought? If you want to understand the answer to this question down at the very core, the first thing you need to understand is something called Boolean logic. Boolean logic, originally developed by George Boole in the mid-1800s, allows quite a few unexpected things to be mapped into bits and bytes.

The great thing about Boolean logic is that, once you get the hang of things, Boolean logic (or at least the parts you need in order to understand the operations of computers) is *outrageously* simple. In this article, we will first discuss simple logic "gates," and then see how to combine them into something useful.

There are three, five or seven simple gates that you need to learn about, depending on how you want to count them. With these simple gates you can build combinations that will implement any digital component you can imagine. The simplest possible gate is called an "inverter", or a NOT gate. It takes one bit as input and produces as output its opposite. The table below shows a logic table for the NOT gate and the normal symbol for it in circuit diagrams.

**NOT Gate**

A	Q
0	1
1	0



You can see in this figure that the NOT gate has one input called **A** and one output called **Q** ("Q" is used for the output because if you used "O," you would easily confuse it with zero). The table shows how the gate behaves. When you apply a 0 to A, Q produces a 1. When you apply a 1 to A, Q produces a 0. Simple.

**AND Gate**

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

The AND gate performs a logical "and" operation on two inputs, A and B. The idea behind an AND gate is, "If A AND B are both 1, then Q should be 1." You can see that behavior in the logic table for the gate.

You read this table row by row, like this:

**OR Gate**

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1



The next gate is an OR gate. Its basic idea is, "If A is 1 OR B is 1 (or both are 1), then Q is 1."

Those are the three basic gates (that's one way to count them).

**NAND Gate**

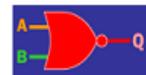
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0



It is quite common to recognize two others as well: the NAND gate and the NOR gate (see on the right).

**NOR Gate**

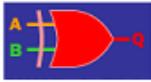
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0



These two gates are simply combinations of an AND or an OR gate with a NOT gate. If you include these two gates, then the count rises to five.

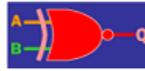
**XOR Gate**

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0



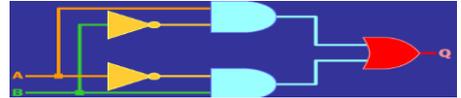
**XNOR Gate**

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1



The final two gates, that are sometimes added to the list are the XOR and XNOR gates, also known as "exclusive or" and "exclusive nor" gates, respectively.

The idea behind an XOR gate is, "If either A OR B is 1, but NOT both, Q is 1." The reason why XOR might not be included in a list of gates is because you can implement it easily using the original three gates listed. Here is one *implementation*:



If you try all four different patterns for A and B and trace them through the circuit, you will find that Q behaves like an XOR gate. Since there is a well-understood symbol for XOR gates, it is generally easier to think of XOR as a "standard gate" and use it in the same way as AND and OR in circuit diagrams.

Now let's learn how we can create a circuit capable of binary addition using the gates described. Let's start with a single-bit adder. Let's say that you have a project where you need to add single bits together and get the answer. The way you would start designing a circuit for that is to first look at all of the logical combinations. You might do that by looking at the following four sums:

0	0	1	1
+0	+1	+0	+1
0	1	1	10
0	0	1	1
+0	+1	+0	+1
00	01	01	10

**1-bit Adder with Carry-Out**

A	B	Q	CO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

That looks fine until you get to 1 + 1. In that case, you have that pesky carry bit to worry about. If you don't care about carrying (because this is, after all, a 1-bit addition problem), then you can see that you can solve this problem with an XOR gate. But if you do care, then you might rewrite your equations to always include 2 bits of output, like this (see on the left).

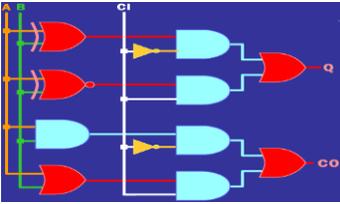
From these equations you can form the logic table. By looking at this table you can see that you can implement Q with an XOR gate and CO (carry-out) with an AND gate. If you want to add two 8-bit bytes together this might become slightly harder. The easiest solution is to modularize the problem into reusable components and then replicate components. In this case, we need to create only one component: a full binary adder. The difference between a full adder and the previous adder we looked at is that a full adder accepts an A and a B input plus a carry-in (CI) input.

Once we have a full adder, then we can string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next.

One-bit Full Adder with Carry-In and Carry-Out

CI	A	B	Q	CO
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Now let's look at how a full adder is implemented into a circuit. The logic table for a full adder is slightly more complicated than the tables we have used before, because



now we have 3 input bits. It looks like this:

There are many different ways that you might implement this table.

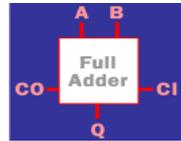
If you look at the Q bit, you can see that the top 4 bits are behaving like an XOR gate with respect to A and B, while the bottom 4 bits are behaving like an XNOR gate with respect to A and B. Similarly, the top 4 bits of CO are behaving like an AND gate with respect to A and B, and the bottom 4 bits behave like an OR gate. Taking those facts, the following circuit implements a full adder shown on the left.

This definitely is not the most efficient way to implement a full adder, but it is extremely easy to understand and trace through the logic using this method.

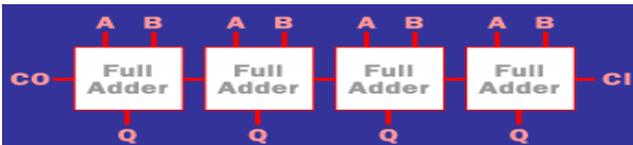
**5. Read the text and find more information on the topic being discussed. Be ready for further discussion.**

## BOOLEAN LOGIC (2)

If you are so inclined, see what you can do to implement the logic spoken above with fewer gates. Now we have a piece of functionality called a "full adder." What a computer engineer then does is "black-box" so that he or she can stop worrying about the details of the component.



A black box for a full adder would look like this shown below:

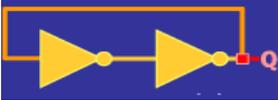


With that black box, it is now easy to draw a 4-bit full adder. In this

diagram the carry-out from each bit feeds directly into the carry-in of the next bit over. A 0 is hard-wired into the initial carry-in bit. If you input two 4-bit numbers on the A and B lines, you will get the 4-bit sum out on the Q lines, plus 1 additional bit for the final carry-out. You can see that this chain can extend as far as you like, through 8, 16 or 32 bits if desired. The 4-bit adder we just created is called a ripple-carry adder. It gets that name because the carry bits "ripple" from one adder to the next. This implementation has the advantage of simplicity but the disadvantage of speed problems. In a real circuit, gates take time to

switch states (the time is on the order of nanoseconds, but in high-speed computers nanoseconds matter). So 32-bit or 64-bit ripple-carry adders might take 100 to 200 nanoseconds to settle into their final sum because of carry ripple. For this reason, engineers have created more advanced adders called carry-look-ahead adders. The number of gates required to implement carry-look-ahead is large, but the settling time for the adder is much better.

One of the more interesting things that you can do with Boolean gates is to create memory with them. If you arrange the gates correctly, they will remember an input value. This simple concept is the basis of RAM in computers, and also makes it possible to create a wide variety of other useful circuits.



The simplest possible feedback circuit using two inverters is shown on the left.

If you follow the feedback path, you can see that if Q happens to be 1, it will always be 1. If it happens to be 0, it will always be 0. Since it's nice to be able to control the circuits we create, this one doesn't have much use -- but it does let you see how feedback works.

It turns out that in "real" circuits, you can actually use this sort of simple inverter feedback approach. A more useful feedback circuit using two NAND gates. This circuit has two inputs (R and S) and two outputs (Q and Q'). Because of the feedback, its logic table is a little unusual compared to the ones we have seen previously.

R	S	Q	Q'
0	0		Illegal
0	1	1	0
1	0	0	1
1	1		Remembers

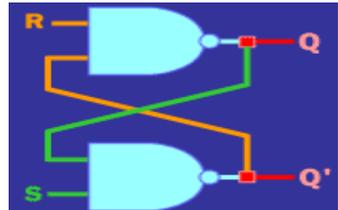
R and S.

There is also the funny illegal state. In this state, R and S both go to 0, which has no value in the memory sense.

Because of the illegal state, you normally add a little conditioning logic on the input side to prevent it, as shown on the right.

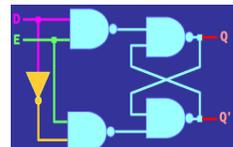
In this circuit, there are two inputs (D and E). You can think of D as "Data" and E as "Enable." If E is 1, then Q will follow D. If E changes to 0, however, Q will remember whatever was last seen on D. A circuit that behaves in this

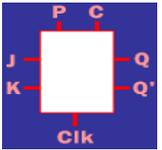
Memory relies on a concept called *feedback*. That is, the output of a gate is fed back into the input. The simplest possible feedback circuit using



What the logic table shows is that: If R and S are opposites of one another, then Q follows S and Q' is the inverse of Q.

If both R and S are switched to 1 simultaneously, then the circuit remembers what was previously presented on





way is generally referred to as a flip-flop. A very common form of flip-flop is the J-K flip-flop.

It is unclear, historically, where the name "J-K" came from, but it is generally represented in a black box like on the left.

In this diagram, P stands for "Preset," C stands for "Clear" and Clk stands for "Clock."

The logic like shown on Here is saying: and Clear and Clk com-Preset goes to goes to 1; and

P	C	Clk	J	K	Q	Q'
1	1	1-to-0	1	0	1	0
1	1	1-to-0	0	1	0	1
1	1	1-to-0	1	1		Toggles
1	0	X	X	X	0	1
0	1	X	X	X	1	0

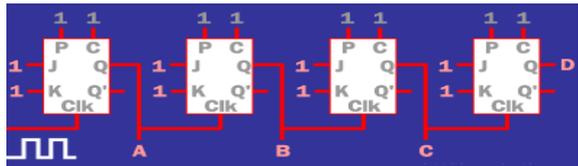
table looks the right. what the table First, Preset override J, K pletely. So if 0, then Q if Clear goes

to 0, then Q goes to 0 no matter what J, K and Clk are doing.

However, if both Preset and Clear are 1, then J, K and Clk can operate. The 1-to-0 notation means that when the clock changes from a 1 to a 0, the value of J and K are remembered if they are opposites. At the low-going edge of the clock (the transition from 1 to 0), J and K are stored. However, if both J and K happen to be 1 at the low-going edge, then Q simply toggles. That is, Q changes from its current state to the opposite state.

It turns out that the concept of "edge triggering" is very useful. The fact that J-K flip-flop only "latches" the J-K inputs on a transition from 1 to 0 makes it much more useful as a memory device. J-K flip-flops are also extremely useful in counters (which are used extensively when creating a digital clock).

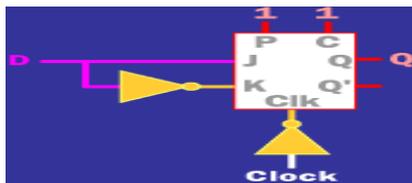
There is an example of a 4-bit counter using J-K flip-flops on the right:



The outputs for this circuit are A, B, C and D, and they represent a 4-bit binary number.

Into the clock input of the left-most flip-flop comes a signal changing from 1 to 0 and back to 1 repeatedly (an oscillating signal).

The counter will count the low-going edges it sees in this signal. That is, every time the incoming signal changes from 1 to 0, the 4-bit number represented by A, B, C and D will increment by 1. So the count will go from 0 to 15 and then cycle back to 0. You can add as many bits as you like to this counter and count anything you like. For example, if you put a magnetic switch on a door, the counter will count the number of times the door is opened and closed. If you put an optical sensor on a road, the counter could count the number of cars that drive by. Another use of a J-K flip-flop is to create an edge-triggered latch, as shown here:



In this arrangement, the value on D is "latched" when the clock edge goes from low to high. Latches are extremely important in the design of things like central processing units (CPUs) and peripherals in computers.

<http://computer.howstuffworks.com/>

**6. Find and learn Russian equivalents for the following words and expressions:**

1) the hang of things	a)
2) logic(al) gate	b)
3) digital component	c)
4) inverter	d)
5) input bit	e)
6) reusable components	f)
7) a full binary adder	g)
8) a carry-in (CI) input	h)
9) carry-look-ahead adder	i)
10) flip-flop	j)
11) to toggle	k)
12) an oscillating signal	l)
13) edge triggering	m)

**7. Find and learn English equivalents for the following words and expressions:**

1) принципиальная схема	a)
2)исключающее ИЛИ НЕ (логическая операция), отрицание альтернативной дизъюнкции	b)
3)одноразрядный сумматор	c)
4)разряд переноса	d)
5)компонент репликации	e)

6) байтовый сумматор	f)
7) сумматор со сквозным переносом	g)
8) байтовый	h)
9) синхронизатор	i)
10) низко интенсивный фронт	j)
11) счётчик (переменная в программе или аппаратный регистр, определяющий число повторений какой-либо операции, например, счётчик цикла)	k)
12) цифровая логическая микросхема (защелка), трактующая перепадом напряжения	l)
13) логическое сложение	m)
14) преобразующая логическая схема	n)

## KEY CONCEPTS

### Adder, half-adder, full-adder

With different combinations of logic gates, a computer performs the math that is the foundation of all its operations. This is accomplished with gate designs called *half-adders* and *full-adders*. A half-adder consists of an XOR gate and an AND gate, both of which receive the same input representing a one-digit binary number. A full-adder consists of half-adders and other switches. A combination of a half-adder and a full-adder handles larger binary numbers and can generate results that involve carrying over numbers. To add the decimal numbers 2 and 3 (10 and 11 in the binary system), first the half-adder processes the digits on the right side through both XOR and AND gates.

### Boolean operators

In 1847, British mathematician George Boole proposed a system of algebra

that could be used to manipulate propositions, that is, assertions that could be either true or false. In his system, called propositional calculus or Boolean Algebra, propositions can be combined using the “and” and “or” operators (called Boolean operators), yielding a new proposition that is also either true or false. For example: “A cat is an animal” AND “The sun is a star” is true because both of the component propositions are true.

“A square has four sides” AND “The Earth is flat” is false because only one of the component propositions is true. However “A square has four sides” OR “The Earth is flat” is true, because *at least one* of the component propositions is true. A chart called a truth table can be used to summarize the AND and OR operations. Here 1 means true and 0 means false, and you read across

from the side and down from the top to see the result of each combination.

### **Logic**

A collection of circuit elements that perform a function, especially a set of elements that use digital logic and perform Boolean logic functions

### **Program counter register**

holds the memory address of the next value the processor will fetch. As soon as a value is retrieved, the processor increments the program counter's con-

tents by 1 so it points to the next program location. (A computer launches a program by putting the program's first value into the counter register.)

### **flip-flop**

The "flip-flop" circuit was invented by two American physicists, W. H. Eccles and R. W. Jordan. The ability of the circuit to switch smoothly between two (binary) states would form the basis for computer arithmetic logic units.

## **! Adverbial clauses of purpose**

**8. Study the following sentence. Say if it is right or wrong: *For testing the new computer, I tried to switch it on.***

We use *for+ing* when we mention the purpose of an object: *This inverter is for turning battery's DC.*

When we mention the purpose of an action, we normally use a to-clause: *To test the new computer, I tried to switch it on.*

We never use *for* immediately before a to-infinitive: *He telephoned me to tell me about new computer design.*

To introduce a clause of purpose, we can use *so that* or (in formal styles) *in order that*, but NOT *that* by itself: *I used Boolean operations **so that** they manipulate individual bits in memory.*

**9. Find and underline the other examples in the text.**

**10. Find the mistakes and correct the sentence.**

- 1) I want to go to Computer classes for learning how to use a computer.
- 2) For testing the new microphone, I tried to record my voice.
- 3) He telephoned me for to tell me about understanding the operations of computers.

**11. Translate the following sentences into Russian.**

- 1) If the power fails, within 4-20 milliseconds the microprocessor closes a switch that sends direct current power from the battery through an inverter, which turns the battery's DC into the alternating current the computer expects to receive.

2) With different combinations of logic gates, a computer performs the math that is the foundation of all its operations. This is accomplished with gate designs called half-adders and full-adders.

3) This NOT gate is designed to take one input from the clock and one from another transistor.

4) For example, CPUs are often compared according to clock speed, but a chip with a superior architecture and algorithm for handling instructions might actually outperform chips with faster clock speeds.

5) Besides being essential to computer design, Boolean operations are also used to manipulate individual bits in memory (see bitwise operations), storing and extracting information needed for device control and other purposes.

## **12. Translate the following sentences into English.**

1) Пользователи базы данных и поисковых интернет служб прекрасно знакомы с использованием логических операторов для определения поискового запроса.

2) Алгоритм анализирует обрабатываемые на данный момент процессором инструкции и пытается предвосхитить необходимую в будущем информацию и инструкции.

3) Для всех остальных ключей, БИОС проверяет те два байта, для того чтобы определить статус сдвига и клавиши-переключателя между двумя состояниями (режимами).

4) Полоса расположена так, чтобы расстояние, которое проходит до нее свет равнялось расстоянию до датчика изображений.

5) В основном, цифровой сигнальный процессор сжимает входящий сигнал для того чтобы он занял меньше пространства.

## **SPEAKING SECTION**

### **13. Answer the following questions.**

1) What is the great thing about Boolean logic?

2) What is Boolean Search?

3) Where does the term Boolean originate?

4) What if you want to add two 8-bit bytes together?

5) What's the difference between a full adder and a single-bit adder?

6) Name the different ways that you might implement 'One-bit full adder with carry-in and carry-out' table.

### **14. Topics for discussion.**

1. Boolean Search.

2. Boolean Algebra and Logic Gates.



15. Prepare a presentation on the topic being discussed.

## REFERENCES, USEFUL LINKS AND FURTHER READING

1. Brain, Marshall. "How Bits and Bytes Work", April 2000./<http://www.howstuffworks.com/>
2. How computer works', by Ron White, 2008, pp.8-9.
3. How Boolean Logic Works on <http://computer.howstuffworks.com/boolean.htm>
4. F. Hamer, M. Lavelle & D McMullan. Boolean Algebra and Logic Gates.

## UNIT 8

### COMPUTER ARCHITECTURE

#### VOCABULARY

##### 1. Match the words with their definitions:

1) precipitate (v.)	[prɪ'sɪpɪteɪt]	a) intensify, increase, or further improve the quality, value, or extent of
2) conversant (adj.)	[kən'vɜ:s(ə)nt]	b) supersede and replace
3) enhance (v.)	[ɪn'hɑ:n(t)s]	c) familiar with or knowledgeable about something
4) fetch (v.)	[fetʃ]	d) cause (an event or situation, typically one that is undesirable) to happen suddenly, unexpectedly, or prematurely
5) supplant (v.)	[sə'plɑ:nt]	e) get smth. from the place where it is

## WATCHING

2. Watch an overview of hardware and software components of a computer system, discuss with your partner and find the true answer to the following question:

- How the following tutorial can help beginning programmers?

## BEFORE YOU READ

3. Discuss with your partner the following question.

- What are people using the computers for?
4. Skim the text to check your ideas.

## READING

### COMPUTER ARCHITECTURE

by Ronald A. Thisted

A computer architecture is a detailed specification of the computational, communication, and data storage elements (hardware) of a computer system, how those components interact (machine organization), and how they are controlled (instruction set). A machine's architecture determines which computations can be performed most efficiently, and which forms of data organization and program design will perform optimally.

The System/360 also *precipitated* a shift from the preoccupation of computer designers with computer arithmetic, which had been the main focus since the early 1950s. In the 1970s and 1980s, computer architects focused increasingly on the instruction set. In the current decade, however, designers' main challenges have been to implement processors efficiently, to design communicating memory hierarchies, and to integrate multiple processors in a single design.

In the late 1970s, statisticians often had to be skilled FORTRAN programmers. Many were also sufficiently *conversant* with assembly language programming for a particular computer that they wrote subprograms directly using the computer's basic instruction set.

The Digital VAX 11/780 was a typical scientific computer of the era. By the middle of the 1980s, such machines (see Key Concepts) were described as "complex instruction-set computers" (CISC). These architectures did have the advantage that each instruction/addressing-mode combination performed its special task efficiently, making it possible for us to fine-tune performance on large tasks with very different characteristics and computing requirements.

Patterson and Ditzel proposed the "reduced instruction-set computer" (RISC) in 1980. The RISC idea is for it to design a small set of instructions which make

implementation of these most frequently performed tasks maximally efficient. The most common features of RISC designs are a single instruction size, a small number of addressing modes, and no indirect addressing.

Early RISC architectures were designed to make Unix operating systems perform efficiently, and today most major Unix-based computer manufacturers rely on a RISC architecture. However, it is important for us to note that many different operating systems can run on a single computer, whether that computer's architecture is a CISC or a RISC design. For instance, Unix, Linux, MacOS, Windows NT, OS/2, and other operating systems all have versions that operate on PowerPC processors. Increasingly, architectures are designed to run multiple operating systems efficiently, and the most common operating systems do operate on many different processors. This means that choice of operating system is becoming less closely tied to the processor family on which a computer is based.

Up to this point we have talked largely about instruction-set aspects of computer architecture. The architectural advances of primary interest to statisticians today involve hardware and machine organization.

The hardware architecture consists of low-level details of a machine, such as timing requirements of components, layouts of circuit boards, logic design, power requirements, and the like. Fortunately, few of these details affect the day-to-day work of statisticians aside from their processors continue to get smaller and faster, and memory continues to get larger and less expensive.

At the level of machine organization, the computers we use are built of interdependent systems, of which the processor itself is just one. Others include memory and memory management systems, specialized instruction processors, busses for communication within and between systems and with peripheral devices, and input/output controllers. In multiprocessing architectures, the protocols for interaction between multiple processors (the multiprocessing control system) is included as well.

In a typical computer based on Intel's Pentium Pro processor, the processor itself is tightly linked to a memory cache consisting of synchronous static random access memory (SRAM). This memory component is very fast, but because it is part of the same chip as the processor itself, it must also be quite small. This level-one cache memory is the first element of the Pentium Pro's memory system. This system in the Pentium architecture consists of a hierarchy of successively larger (but slower) memory layers: a level-two cache, coupled to main memory (dynamic RAM, or extended data-out (EDO) DRAM). The main memory bank is backed up in turn by virtual memory residing on disk; a memory management chipset (separate from the Pentium processor) controls transactions between these two layers of the memory system.

Input and output are handled via plug-in slots for processor cards attached to the system bus, which also serves the graphics controller. Typically, graphics output has its own, separate, graphics memory for preparing and displaying the contents of graphics displays.

In addition, the graphics subsystem may incorporate specialized processors that accelerate graphical computations as well as specialized memory modules linked to the graphics processors.

For most statisticians using a computer on the desktop (or increasingly, on the laptop), then, improving performance can often be achieved by increasing the amount of level-two cache memory, by increasing the amount of main memory, by acquiring faster and larger disk drives, by adding graphics accelerators and graphics memory, or by adding a high-speed input/output controller.

On more powerful workstation systems, many of these features are integrated directly on the main processor chip. The UltraSPARC-1, for instance, has the memory management unit, the floating-point processor, as well as graphics and imaging support on the same chip as the instruction processor.

From the earliest days of digital computers, statistical computation has been dominated by floating-point calculations. "Scientific computers" are often defined as those which deliver high performance for floating-point operations. The aspects of computer design that make these operations possible have followed the same evolutionary path as the rest of computer architecture. Early computer designs incorporated no floating-point instructions. Since all numbers were treated as integers, programmers working with nonintegers had to represent a number using one integer to hold the significant digits coupled to a second integer to record a scaling factor. In effect, each programmer had to devise his or her own floating-point representation.

By the 1960s, some designs introduced instructions for floating-point operations, but many had none. (The IBM 1620 computer, for example, had fixed-precision integer arithmetic, but the programmer could control the number of digits of precision.) By the mid-1970s, virtually all scientific computers had floating-point instructions. Unfortunately for the practical statistician, the representation of floating-point numbers, the meaning of floating point operations, and the results of an operation differed substantially from one machine to the next. The burden of knowing numerical analysis and good numerical algorithms fell heavily on the data analyst's shoulders.

In the early 1980s the Institute of Electrical and Electronics Engineers (IEEE) developed a standard for floating-point arithmetic [1]. To implement these standards, computer architects of that time developed a floating-point architecture separate from that of the principal processor in effect, moving the floating-point issues from the level of machine-specific definition of arithmetic to a common set of operations (an "instruction set") whose output could be strictly defined. Examples include the Motorola 6888x and the Intel 80x87 floating-point processors (FPPs), which were designed in parallel with the 680x0 and 80x86 central processors, respectively. The PowerPC family also includes an integrated IEEE-compliant FPP. These processors illustrate the increased role of hardware and machine organization. The FPPs are logically external to the basic instruction processor. Therefore, the computer design must include channels for

communication of operands and results, and must incorporate machine-level protocols to guarantee that the mix of floating-point and non-floating-point instructions are completed in the correct order. Today, many of these FPPs are part of the same integrated-circuit package as the main instruction processor. This keeps the lines of communication very short and achieves additional effective processor speed.

The discussion above concerns the predominant computer architecture used by practicing statisticians, one based on the sequential single processor. The notion that speed and reliability could be *enhanced* by coupling multiple processors in a way that enabled them to share work is an obvious extension, and one that has been explored actively since at least the 1950s.

Vector computers include instructions (and hardware!) that make it possible to execute a single instruction (such as an add) simultaneously on a vector of operands. In a standard scalar computer, computing the inner product of two vectors  $x$  and  $y$  of length  $p$  requires a loop within which the products  $x_i y_i$  are calculated. The time required is that of  $p$  multiplications, together with the overhead of the loop. On a vector machine, a single instruction would calculate all  $p$  products at once. The highest performance scientific computers available since 1975 incorporate vector architecture (see Key Concepts). Examples of early machines with vector capabilities include the CRAY-1 machine and its successors from Cray Research and the CYBER-STAR computers from CDC.

Computers with multiple processors having individual data memory, and which *fetch* and execute their instructions independently (MIMD computers), are more flexible than SIMD machines and can be built by taking regular scalar microprocessors and organizing them to operate in parallel. Such multiprocessors are rapidly approaching the capabilities of the fastest vector machines, and for many applications already have *supplanted* them.

In the early days of digital computers, few useful computations could be accomplished without a thorough understanding of the computer's instruction set and a detailed knowledge of its architecture. Advances in computer architecture, coupled with advances in programming languages, compiler technology, operating systems, storage, and memory, have made such specific knowledge of much reduced importance for producing high-quality scientific work.

## LANGUAGE DEVELOPMENT

**5. Find and learn Russian equivalents for the following words and expressions:**

1) single design	a)
2) fine-tune	b)
3) reduced instruction set computer	c)

4) instruction set	d)
5) peripheral device	e)
6) scaling factor	f)
7) scalar computer	g)

**6. Find and learn English equivalents for the following words and expressions:**

1) режим (способ) адресации	a)
2) оптимизация (программы) в процессе компиляции	b)
3) вычислительный ресурс	c)
4) взаимозависимые системы	d)
5) кэш; кэш-память	e)
6) процессор для обработки чисел с плавающей запятой	f)
7) последовательный процессор	g)
8) обработка векторных данных	h)

**7. Find and learn the definitions for the following abbreviations.**

1) VAX	a)
2) CISC	b)
3) RISC	c)
4) VMS	d)
5) SRAM	e)
6) DRAM	f)
7) EDO	g)
8) IEEE	h)
9) FPPs	i)
10) CDC	j)
11)SIMD	k)
12)MIMD	l)
13)FPU	m)

## KEY CONCEPTS

### Architecture

As applied to computer design, was first used in 1964 by Gene Amdahl, G.

Anne Blaauw, and Frederick Brooks, Jr., the designers of the IBM System/360. They coined the term to refer

to those aspects of the instruction set available to programmers, independent of the hardware on which the instruction set was implemented. The System/360 marked the introduction of families of computers, that is, a range of hardware systems all executing essentially the same basic machine instructions. Von Neumann provided the comprehensive theoretical architecture for the modern computer, which would become known as the von Neumann architecture. Von Neumann's reports would be distributed widely and would guide the beginnings of computer science research in many parts of the world.

**IEEE (The Institute of Electrical and Electronics Engineers)** (read *I-Triple-E*)

A non-profit professional association dedicated to advancing technological innovation related to electricity. It has more than 395,000 members in more than 160 countries, 45% outside the United State.

**CISC (complex instruction set computing)**

A processor architecture design in which large, complicated instructions are broken down into smaller tasks before the processor executes them.

**Floating point unit (FPU)**

A part of a computer system specially designed to carry out operations on floating point numbers. Typical operations are addition, subtraction, multiplication, division, and square root. Some systems (particularly older, microcode-based architectures) can also perform various transcendental functions such as exponential or trigonometric calculations, though in most

modern processors these are done with software library routines. In most modern general purpose computer architectures, one or more FPUs are integrated with the CPU; however many embedded processors, especially older designs, do not have hardware support for floating point operations. In the past, some systems have implemented floating point via a coprocessor rather than as an integrated unit; in the microcomputer era, this was generally a single microchip, while in older systems it could be an entire circuit board or a cabinet. Not all computer architectures have a hardware FPU. In the absence of an FPU, many FPU functions can be emulated, which saves the added hardware cost of an FPU but is significantly slower. Emulation can be implemented on any of several levels - in the CPU as microcode, as an operating system function, or in user space code. In most modern computer architectures, there is some division of floating point operations from integer operations. This division varies significantly by architecture; some, like the Intel x86 have dedicated floating point registers, while some take it as far as independent clocking schemes.

**Memory cache**

A small amount of fast memory used for instructions or data that are likely to be needed was introduced in the IBM 360/85 series (1967).

**OpenVMS (Open Virtual Memory System)**, previously known as **VAX-11/VMS**, **VAX/VMS** or (informally) **VMS**

A high-end computer server operating system that runs on VAX, Alpha and Itanium-based families of computers. Contrary to what its name suggests, OpenVMS is not open source software; however, the source listings are available for purchase. Unlike some other mainframe-oriented operating systems, OpenVMS has a graphical user interface (GUI) with complete graphics support. Digital Equipment Corporation's VAX was one of the three top-selling workstations lines in the 1980s and 1990s. VMS had support for professional DTP and CAE software running. OpenVMS is a multi-user, multiprocessing virtual memory-based operating system (OS) designed for use in time sharing, batch processing, real-time (where process priorities can be set higher than OS kernel jobs), and transaction processing. It offers high system availability through clustering, or the ability to distribute the system over multiple physical machines. This allows the system to be "disaster-tolerant" against natural disasters that may disable individual data-processing facilities.

VMS also includes a process priority system that allows for real-time processes to run unhindered, while user processes get temporary priority "boosts" if necessary.

OpenVMS commercialized many features that are now considered standard requirements for any high-end server operating system.

**RISC (reduced instruction set computing)**

A processor design in which only small, quickly executing instructions are used.

RISC architectures became popular in the middle to late 1980s. Examples of RISC architectures include Digital's Alpha processor family, Hewlett-Packard's PA series, the Motorola/Apple/IBM PowerPC family, and Sun's Ultra-SPARC family. RISC architectures are particularly suitable for taking advantage of compiler-based optimization, which means that programs such as computationally-intensive statistical procedures written in compiled languages are likely to see best performance in RISC environments. In later RISC-based architectures, floating point processes are tightly coupled to the central instruction processor. The UltraSPARC-1 design incorporates an IEEE-compliant FPP which performs all floating-point operations (including multipliers, dividers, and square roots).

### **Operating system**

A program that allocates and schedules the computer's computational resources such as processors and memory. For instance, the operating system controls how much of the computer's main memory is available to programs and how much is used to cache portions of files being read from or written to disk storage. Operating systems also determine which programs will be processed at any given time, and how long the processor will "belong" to the program currently claiming it. Computers such as the VAX and System/360 were designed in conjunction with an architecture-specific operating system (VMS and OS/360, respectively). Indeed, until the late 1980s, most operating systems were written specifically for computers with a particular architecture, and only a small number of these operating systems were ever converted to control machines based on other designs.

**SIMD (Single Instruction Multiple Data)** A processor architecture that allows the same operation to be performed on multiple pieces of data simultaneously.

### **VAX**

Had over 300 different machine-level instructions, ranging in size from 2 to 57 bytes in length, and 22 different addressing modes. Machines such as the VAX, the Intel 80x86 family of processors (the processors on which the IBM PC and its successors are based), and the Motorola 680x0 processors (on which the Apple Macintosh is based) all had multiple addressing modes, variable-length instructions, and large instruction sets.

### **Vector processors**

Special cases of parallel architecture, in which multiple processors cooperatively perform computations. Vector processors are examples of machines which can execute a single instruction on multiple data streams (SIMD computers). In computers with these architectures, there is a single queue of instructions which are executed in parallel (that is, simultaneously) by multiple processors, each of which has its own data memory cache. Except for special purposes (such as array processing), the SIMD model is neither sufficiently flexible nor economically competitive for general-purpose designs.

## **! For-clause**

**8. Study the following: It is possible for us to fine-tune performance on large tasks.**

*For*-structures with preparatory *it* are common with many adjectives expressing possibility, necessity, importance, urgency, frequency and value judgments.

**(...) it (...) + adjective +for + object +infinitive**

Other common adjectives that are used in this way include *vital, necessary, pointless, unimportant, common, normal, unusual, rare, right, wrong*. Note that *likely* and *probable* are not used like this. ***She is likely to choose that operating system. It's probable that she'll choose that operating system or She'll probably choose that operating system.***

The structure can also be used after nouns in expressions with meanings similar to the adjectives listed above: time, a good/bad idea, plan, aim, need, request, mistake, shame: Our **aim is for students to learn** as quickly as possible.

**9. Find, underline and translate the other examples in the text.**

**10. Find the mistakes and correct the sentence.**

- 1) It is likely for her to implement this task.
- 2) It's probable for him to learn Computer Science.
- 3) I thought it is strange for this programme to be written specifically for computers.
- 4) It's no good this machine to execute a single queue of instructions.

**11. Translate the following sentences into Russian.**

1) They also chose the Intel 8086 and 8088 microprocessors, which had an instruction set similar to that of the Intel 8080 used in many CP/M systems.

2) The earliest home computer games were typically the product of a single designer's vision, such as Chris Crawford's *Balance of Power* and Richard Garriott ("Lord British") in the Ultima series.

3) Compatibility was ensured by devising a single 360 instruction set that was implemented at the machine level by microcode stored in ROM (read-only memory) and optimized for each model.

4) Input/Output (I/O) is built into the language rather than provided in an external library, and the flexible options include character, streams of characters, blocks, and records with either sequential or random access.

5) During the 1980s, Grove would be challenged to be adaptable when Japanese companies eroded Intel's share of the DRAM (memory) chip market, often "dumping" product below their cost.

## 12. Translate the following sentences into English.

- 1) Все компоненты информационной системы являются взаимозависимыми потому что все они зависят друг от друга.
- 2) Вектор и обработка векторных данных по существу являются одним и тем же, хотя небольшие технические различия им присущи.
- 3) Каждый кластер имеет свою совместно используемую память.
- 4) Скалярный процессор это процессор, который обрабатывает один элемент данных за одну инструкцию.
- 5) Архитектура компьютера это и искусство и наука сборки логических элементов, для того чтобы создать вычислительное устройство.

## SPEAKING SECTION

### 13. Answer the following questions.

1. What is a computer architecture?
2. What is the term architecture applied to?
3. How has this transition from concentrating on instruction sets to high-level integration taken place?
4. What machines were described as CISC?
5. What architectures became popular in the middle to late 1980s?
6. What were they designed for?
7. Name the examples of early machines with vector capabilities.
8. What is UltraSPARC?
9. What every computer scientist should know about floating-point arithmetic?

### 14. Topics for discussion.

1. Computing Trends & The Mobile Experience
2. Scale out processors.
3. ISA wars.



15. Prepare a presentation on the topic being discussed.

## REFERENCES, USEFUL LINKS AND FURTHER READING

1. 'How computer works', by Ron White, 2008, pp.8-9.
2. Institute of Electrical and Electronics Engineers (IEEE). (1985). IEEE Standard for Binary Floating-Point Arithmetic (Standard 754-1985). IEEE, New York. Reprinted in SIGPLAN Notices, 22(2), 5{48.

3. Goldberg, D. (1991) "What every computer scientist should know about floating-point arithmetic," *Computing Surveys*, 23(1), 5{48.

4. Patterson, David A., and Ditzel, D. R. (1980). "The case for the reduced instruction set computer," *Computer Architecture News*, 8(6), 25{33.

5. Thisted, Ronald A. (1988). *Elements of Statistical Computing: Numerical Computation*. Chapman and Hall: London.

6. Ronald A. Thisted Departments of Statistics, Health Studies, and Anesthesia & Critical Care The University of Chicago 7 April 1997 To appear, *Encyclopedia of Biostatistics*.

## UNIT 9

### OPERATING SYSTEM

#### VOCABULARY

**1. Match the words with their definitions:**

1) bandwidth (n.)	[ˈbændwɪdθ]	a) a person or company offering something for sale
2) kernel (n.)	[ˈkɜːn(ə)l]	b) use (resources) economically
3) scrap (v.)	[skræp]	c) a range of frequencies within a given band, in particular that used for transmitting a signal
4) husband (v.)	[ˈhʌzbənd]	d) the most basic level or core of an operating system, responsible for resource allocation, file management, and security
5) vendor (n.)	[ˈvendɔː]	e) discard or remove from service (a redundant, old, or inoperative vehicle, vessel, or machine)

#### WATCHING

**2. Watch an overview and learn about operating systems. Discuss with your partner and find the true answer to the following question:**

- Why an operating system is the most important software that runs on a computer?

## BEFORE YOU READ

### 3. Disuss with your partner the following question.

- What are people using the Operating systems for?

### 4. Skim the text to check your ideas.

## READING

## THE PURPOSE OF AN OPERATING SYSTEM

*by Curt Franklin and Dave Coustan*

When you turn on your computer, it's nice to think that you're in control. Although it's easy to feel like a director in front of your desktop or laptop, there's a lot going on inside, and the real man behind the curtain handling the necessary tasks is the operating system (OS) -- the first thing loaded onto the computer without it, a computer is useless. The purpose of an operating system is to organize and control hardware and software so that the device it lives in behaves in a flexible but predictable way. Not all computers have operating systems. The computer that controls the microwave oven in your kitchen, for example, doesn't need an operating system. For a computer like this, an operating system would be unnecessary baggage, driving up the development and manufacturing costs significantly and adding complexity where none is required. For other devices, an operating system creates the ability to serve a variety of purposes, interact with users in more complicated ways, keep up with needs that change over time.

All desktop computers have operating systems. The most common are the Windows family of operating systems developed by Microsoft, the Macintosh operating systems developed by Apple and the UNIX family of operating systems (which have been developed by a whole history of individuals, corporations and collaborators). There are hundreds of other operating systems available for special-purpose applications, including specializations for mainframes, robotics, manufacturing, real-time control systems and so on. In any device that has an operating system, there's usually a way to make changes to how the device works. This is far from a happy accident; one of the reasons operating systems are made out of portable code rather than permanent physical circuits is so that they can be changed or modified without having to scrap the whole device. For a desktop computer user, this means you can add a new security update, system patch, new application or even an entirely new operating system rather than junk your computer and start again with a new one when you need to make a change. As long as you understand how an operating system works and how to get at it, in many cases you can change some of the ways it behaves.

At the simplest level, an operating system does two things: Firstly, it manages the hardware and software resources of the system. In a desktop computer, these resources include such things as the processor, memory, disk space and more (on a cell phone, they include the keypad, the screen, the address book, the phone dialer, the battery and the network connection). Secondly, it provides a stable, consistent way for applications to deal with the hardware without having to know all the details of the hardware. The first task, managing the hardware and software resources, is very important, as various programs and input methods compete for the attention of the **central processing unit (CPU)** (see Key Concepts), and demand memory, storage and input/output (I/O) *bandwidth* for their own purposes. In this capacity, the operating system plays the role of the good parent, making sure that each application gets the necessary resources while playing nicely with all the other applications, as well as husbanding the limited capacity of the system to the greatest good of all the users and applications. The second task, providing a consistent application interface, is especially important if there is to be more than one of a particular type of computer using the operating system, or if the hardware making up the computer is ever open to change. A consistent **application program interface (API)** allows a software developer to write an application on one computer and have a high level of confidence that it will run on another computer of the same type, even if the amount of memory or the quantity of storage is different on the two machines. Even if a particular computer is unique, an operating system can ensure that applications continue to run when hardware upgrades and updates occur. This is because the operating system -- not the application -- is charged with managing the hardware and the distribution of its resources. One of the challenges facing developers is keeping their operating systems flexible enough to run hardware from the thousands of vendors manufacturing computer equipment. Today's systems can accommodate thousands of different printers, disk drives and special peripherals in any possible combination.

Within the broad family of operating systems, there are generally four types, categorized based on the types of computers they control and the sort of applications they support. The categories are: **Real-time operating system (RTOS)**; **Single-user, single task**; **Single-user, multi-tasking**; **Multi-user**.

It's important to differentiate between multi-user operating systems and single-user operating systems that support networking. Windows 2000 and Novell Netware can each support hundreds or thousands of networked users, but the operating systems themselves aren't true multi-user operating systems. The **system administrator** is the only "user" for Windows 2000 or Netware. The network support and all of the remote user logins the network enables are, in the overall plan of the operating system, a program being run by the administrative user.

While there are some who argue that an operating system should do more than these six tasks, and some operating-system vendors do build many more utility programs and auxiliary functions into their operating systems, these six tasks

define the core of nearly all operating systems. Next, let's look at the tools the operating system uses to perform each of these functions.

The heart of managing the processor comes down to two related issues: Ensuring that each process and application receives enough of the processor's time to function properly and using as many processor cycles as possible for real work. The basic unit of software that the operating system deals with in scheduling the work done by the processor is either a **process** or a **thread**, depending on the operating system.

In a single-tasking system, the schedule is straightforward. The operating system allows the application to begin running, suspending the execution only long enough to deal with **interrupts** and user input. While interrupts add some complication to the execution of processes in a single-tasking system, the job of the operating system becomes much more complicated in a multi-tasking system. Now, the operating system must arrange the execution of applications so that you believe that there are several things happening at once. This is complicated because the CPU can only do one thing at a time. Today's multi-core processors and multi-processor machines can handle more work, but each processor core is still capable of managing one task at a time.

All of the information needed to keep track of a process when switching is kept in a data package called a **process control block**. The process control block typically contains: an ID number that identifies the process; pointers to the locations in the program and its data where processing last occurred; register contents; states of various flags and switches; pointers to the upper and lower bounds of the memory required for the process; a list of files opened by the process; the priority of the process; the status of all I/O devices needed by the process.

Each process has a status associated with it. Many processes consume no CPU time until they get some sort of input. In a system with two or more CPUs, the operating system must divide the workload among the CPUs, trying to balance the demands of the required processes with the available cycles on the different CPUs: **Asymmetric operating systems** and **Symmetric operating systems**.

If the operating system is the only software with execution needs, the CPU is not the only resource to be scheduled. Memory management is the next crucial step in making sure that all processes run smooth.

When an operating system manages the computer's memory, there are two broad tasks to be accomplished:

- 1) Each process must have enough memory in which to execute, and it can neither run into the memory space of another process nor be run into by another process.

- 2) The different types of memory in the system must be used properly so that each process can run most effectively.

The first task requires the operating system to set up memory boundaries for types of software and for individual applications. In most computers, it's possible to add memory beyond the original capacity. Disk storage is only one of the

**memory types** that must be managed by the operating system, and it's also the slowest. Ranked in order of speed, the types of memory in a computer system are: **High-speed cache; Main memory; Secondary memory.** The operating system must balance the needs of the various processes with the availability of the different types of memory, moving data in blocks (called pages) between available memory as the schedule of processes dictates.

The path between the operating system and virtually all hardware not on the computer's motherboard goes through a special program called a driver. Much of a driver's function is to be the translator between the electrical signals of the hardware subsystems and the high-level programming languages of the operating system and application programs. Drivers take data that the operating system has defined as a file and translate them into streams of bits placed in specific locations on storage devices, or a series of laser pulses in a printer. Because there are such wide differences in the hardware, there are differences in the way that the driver programs function. Most run when the device is required, and function much the same as any other process. The operating system will frequently assign high-priority blocks to drivers so that the hardware resource can be released and readied for further use as quickly as possible.

Managing all the resources of the computer system is a large part of the operating system's function and, in the case of real-time operating systems, may be virtually all the functionality required. For other operating systems, though, providing a relatively simple, consistent way for applications and humans to use the power of the hardware is a crucial part of their reason for existing. Just as drivers provide a way for applications to make use of hardware subsystems without having to know every detail of the hardware's operation, APIs (see above) let application programmers use functions of the computer and operating system without having to directly keep track of all the details in the CPU's operation.

Just as the API provides a consistent way for applications to use the resources of the computer system, a user interface (UI) brings structure to the interaction between a user and the computer. In the last decade, almost all development in user interfaces has been in the area of the graphical user interface (GUI), with two models, Apple's Macintosh and Microsoft's Windows, receiving most of the attention and gaining most of the market share. The popular open-source Linux operating system also supports a graphical user interface. There are other user interfaces, some graphical and some not, for other operating systems. . The core operating-system functions -- the management of the computer system -- lie in the *kernel* of the operating system. The display manager is separate, though it may be tied tightly to the kernel beneath. The ties between the operating-system kernel and the user interface, utilities and other software define many of the differences in operating systems today, and will further define them in the future.

**5. Find and learn Russian equivalents for the following words and expressions:**

1) bootstrap loader	a)
2) special-purpose applications	b)
3) application interface	c)
4) real-time operating system	d)
5) multiuser	e)
6) single-task system	f)
7) thread	g)
8) high-priority	h)
9) interrupt	

**6. Find and learn English equivalents for the following words and expressions:**

1) структура данных в программировании	a)
2) мобильный код (легко переносимый на другую платформу)	b)
3) код для оперативного исправления или нейтрализации ошибки в исполняемой программе	c)
4) полоса (частот); полоса пропускания	d)
5) согласованная компиляция	e)
6) однопользовательский	f)
7) многозадачность, многозадачная работа, многозадачный режим	g)
8) поставщик операционной системы	h)
9) интерфейс прикладного программирования	i)
10) блок управления технологическим процессом	j)

**7. Find and learn the definitions for the following abbreviations.**

1) RTOS	a)
2) API	b)
3) ID	c)
4) I/O	d)
5) CPU	e)

6) AOS	f)
7) POST	g)
8) SOS	h)
9) BIOS	i)
10) PCB	j)
11) NMIs	k)

## KEY CONCEPTS

### Application program interface (API)

In order for an application program to function, it must interact with the computer system in a variety of ways, such as reading information from disk files, sending data to the printer, and displaying text and graphics on the monitor screen (see user interface). The program may need to find

out whether a device is available or whether it can have access to an additional portion of memory. In order to provide these and many other services, an operating system such as Microsoft Windows includes an extensive application program interface (API). The API basically consists of a variety of functions or procedures that an application program can call upon, as well as data structures, constants, and various definitions needed to describe system resources. Applications programs use the API by including calls to routines in a program library (see library, program and procedures and functions). In Windows, “dynamic link libraries” (DLLs) are used. In practice, the API for a major operating system such as Windows contains hundreds of functions, data structures, and definitions. In recent years Microsoft has greatly extended the reach of its Windows API by providing many higher level functions

(including user interface items, network communications, and data access) previously requiring separate software components or program libraries.

### Asymmetric operating systems

Use one CPU for their own needs and divide application processes among the remaining CPUs.

### Bootstrap loader

In most modern computers, when the computer activates the hard disk drive, it finds the first piece of the operating system: the bootstrap loader. The bootstrap loader is a small program that has a single function: It loads the operating system into memory and allows it to begin operation. In the most basic form, the bootstrap loader sets up the small driver programs that interface with and control the various hardware subsystems of the computer. It sets up the divisions of memory that hold the operating system, user information and applications. It establishes the data structures that will hold the myriad signals, flags and semaphores that are used to communicate within and between the subsystems and applications of the computer. Then it turns control of the computer over to the operating system.

### Central processing unit (CPU)

An acronym for *central processing unit*, it is used to mean the *microprocessor*-

also, *processor*-which is a microchip that processes the information and the code (instructions) used by a computer. The "brains" of a computer.

**Interrupts**

Special signals sent by hardware or software to the CPU. It's as if some part of the computer suddenly raised its hand to ask for the CPU's attention in a lively meeting. Sometimes the operating system will schedule the priority of processes so that interrupts are masked -- that is, the operating system will ignore the interrupts from some sources so that a particular job can be finished as quickly as possible. There are some interrupts (such as those from error conditions or problems with memory) that are so important that they can't be ignored. These non-maskable interrupts (NMIs) must be dealt with immediately, regardless of the other tasks at hand.

**Multi-user**

Allows many different users to take advantage of the computer's resources simultaneously. The operating system must make sure that the requirements of the various users are balanced, and that each of the programs they are using has sufficient and separate resources so that a problem with one user doesn't affect the entire community of users. Unix, VMS and mainframe operating systems, such as *MVS*, are examples of multi-user operating systems.

**Operating system**

Software that exists to control the operations of hardware. Essentially, the operating system directs any operation, such as writing data to memory or to disk, and regulates the use of hardware among several application programs that are running at the same time. This frees program de-

velopers from having to write their own code for these most basic operations.

**Power-on self test (POST)**

Checks the CPU, memory, and basic input-output systems (BIOS) for errors and stores the result in a special memory location. Once the POST has successfully completed, the software loaded in ROM (sometimes called the BIOS or **firmware**) will begin to activate the computer's disk drives.

**Process**

A process is software that performs some action and can be controlled -- by a user, by other applications or by the operating system.

It is processes, rather than applications, that the operating system controls and schedules for execution by the CPU.

**Real-time operating system (RTOS)**

Used to control machinery, scientific instruments and industrial systems. An RTOS typically has very little user-interface capability, and no end-user utilities, since the system will be a "sealed box" when delivered for use. A very important part of an RTOS is managing the resources of the computer so that a particular operation executes in precisely the same amount of time, every time it occurs. In a complex machine, having a part move more quickly just because system resources are available may be just as catastrophic as having it not move at all because the system is busy.

**Single-user, single task**

As the name implies, this operating system is designed to manage the computer so that one user can effectively do one thing at a time. The Palm OS for Palm handheld computers is a good example of a modern single-user, single-task operating system.

### **Single-user, multi-tasking**

This is the type of operating system most people use on their desktop and laptop computers today. Microsoft's Windows and Apple's MacOS platforms are both examples of operating systems that will let a single user have several programs in operation at the same time. For example, it's entirely possible for a Windows user to be writing a note in a word processor while downloading a file from the Internet while printing the text of an e-mail message.

### **Symmetric operating systems**

Divide themselves among the various CPUs, balancing demand versus CPU availability even when the operating system itself is all that's running.

### **System administrator**

A system administrator is the person responsible for managing the operations of a computer facility to ensure that it runs properly, meets user needs, and protects the integrity of users' data. Such facilities range from offices with just a few users to large campus or corporate facilities that may be served by a large staff of administrators. System administrators often write scripts to automate many of the above tasks (see scripting languages). Because of the complexity of modern computing environments, an administrator usually specializes in a particular operating system such as UNIX or Windows. A good system administrator needs not only technical understanding of the many components of

the system, but also the ability to communicate well with users—good “people skills.” Larger organizations are more likely to have separate network and database administrators, while the administrator of a small facility must be a jack (or jill) of all trades.

### **Types of memory**

**High-speed cache** -- This is fast, relatively small amounts of memory that are available to the CPU through the fastest connections.

Cache controllers predict which pieces of data the CPU will need next and pull it from main memory into high-speed cache to speed up system performance.

**Main memory** -- This is the RAM that you see measured in megabytes when you buy a computer.

**Secondary memory** -- This is most often some sort of rotating magnetic storage that keeps applications and data available to be used, and serves as virtual RAM under the control of the operating system.

### **Virtual memory management**

Most of the information that an application stores in memory is not being used at any given moment. A processor can only access memory one location at a time, so the vast majority of RAM is unused at any moment. Since disk space is cheap compared to RAM, then moving information in RAM to hard disk can greatly expand RAM space at no cost.

## **! Phrasal verbs**

**8. Study the following: *I gave up Computer Science because I didn't like it.***

Common short verbs like *bring, come, do, get, give, go, have, keep, make, put, and take* are very often used with prepositions or adverb particles (e.g. *on,*

*off, up, away*) to make two-word verbs. These are called ‘prepositional verbs’ or ‘phrasal verbs’, and many of them are idiomatic.

The meaning of a two-word verb is sometimes very different from the meanings of the two parts taken separately, e.g. *give up, put off, work out*.

With many phrasal verbs, there is a choice of word order: *A: I turned off the computer; B: I turned the computer off*. However, when the direct object is a *pronoun*, only pattern B is possible.

**9. Find, underline and translate the other 9 examples in the text.**

**10. Find the mistakes and correct the sentence.**

1. The fan was very noisy and so I turned off it.
2. Our teachers brought up us to respect the science.
3. The woman was handing free samples of a new type of software out.

**11. Translate the following sentences into Russian.**

1. If you like to tinker with electronic devices, you're probably pleased that operating systems can now be found on many of the devices we use every day, from cell phones to wireless access points.

2. The computer in a microwave oven simply runs a single hard-wired program all the time.

3. Bootstrap loader is a very short program loading routine, used for loading other loaders in a computer; often implemented in a read-only memory.

4. Multithreaded is a type of program designed to take advantage of Intel processors' ability to execute more than one string of data-thread-at the same time.

5. The FSF (Free software foundation) high-priority projects list serves to foster the development of projects that are important for increasing the adoption and use of free software and free software operating systems.

6. After subtracting the overhead for packet packaging, the basic PCI-Express slot has a top bandwidth of 250 megabytes a second.

**12. Translate the following sentences into English.**

1. Многопоточные (тредовые) программы должны быть спроектированы таким образом, чтобы треды могли исполняться независимо и не мешать друг другу.

2. Форумы предоставляют техническую поддержку для любого аппаратного и программного обеспечения одновременно с загрузкой библиотек (хранилищ), содержащих системные коды, драйвера, сервисные программы (утилиты), шаблоны, макросы и другие дополнения.

3. Блок управления процессом это структура данных, содержащих информацию в ядре операционной системы необходимую для того, чтобы управлять определенным процессом.

4. В системном программировании, сигнал прерывания это сигнал процессору, произведенный аппаратным или программным обеспечением, означающим событие, требующего немедленного внимания.

5. Мобильность важна, так как немобильный код причиняет много проблем в техническом обслуживании – исполнение многочисленных версий, плохая считываемость/распознавание кода.

## SPEAKING SECTION

### 13. Answer the following questions.

1. What a piece of software must do to be called an operating system?
2. What is the purpose of an operating system?
3. Regardless of what device an operating system runs, what exactly can it do?
4. Name the types of operating systems.
5. What are the basic functions provided by an operating system and what six categories do the operating system's tasks fall into?
6. Give an example of numerous processes that run without giving you direct evidence that they ever exist.
7. What services does an operating system provide?
8. Give some examples of how to take control of the other operating systems around you?
9. Why do many companies want to provide applications to the public at no charge?
10. What is the difference between symmetric and asymmetric operating systems for shared memory multiprocessors? What are their respective advantages and disadvantages?

### 14. Topics for discussion.

1. The operating system's tasks.
2. The history of first operating systems.
3. Symmetric and asymmetric operating systems.



15. Prepare a presentation on the topic being discussed.

## REFERENCES, USEFUL LINKS AND FURTHER READING

1. 'How computer works', by Ron White, 2008, p.22.
2. <http://computer.howstuffworks.com/>

3. Computer Desktop Encyclopedia copyright ©1981-2012 by The Computer Language Company Inc. p.33.

4. “DevCentral Tutorials: MFC and Win32.” Available online. URL: <http://devcentral.itech.com/learning/tutorials/submfc.asp>. Accessed January 26, 2012.

5. Petzold, Charles. Programming Windows: the Definitive Guide to the Win32 API. 5th ed. Redmond, Wash.: Microsoft Press, 1999. “Windows API Guide.” Available online. URL: <http://www.vbapi.com/>. Accessed January 26, 2012.

## UNIT 10

### MONITOR DISPLAY TECHNOLOGY

#### VOCABULARY

**1. Match the words with their definitions:**

1) venerable (adj.)	['ven(ə)rəbl]	a) place or arrange (things) in a straight line
2) fuzzy (adj.)	[fʌzi]	b) a framework of spaced bars that are parallel to or cross each other
3) backlight (n.)	[bæklait]	c) a conducting wire or thread with a high melting point, forming part of an electric bulb or vacuum tube and heated or made incandescent by an electric current
4) align (v.)	[ə'lam]	d) old
5) grid (n.)	[grid]	e) illumination from behind
6) filament (n.)	['filəmənt]	f) unclear and hard to see or hear

**2. Discuss with your partner the following questions.**

- What do you know about displays?
- What are the reasons for inventing new types of displays?

**3. Skim the text to check your ideas.**

#### READING

### HOW A COMPUTER DISPLAY WORKS

Often referred to as a monitor when packaged in a separate case, the display is the most-used output device on a computer. Monitor ensures the data correlation between the user and the computer. Specifically, through the monitor we accept entire visual information from the computer. There are following types of monitors: CRT, LCD, PDP, FED, LEP (monitors on the basis of shone plastic (while only experimental models)).

Most desktop displays use liquid crystal display or cathode ray tube technology, while nearly all portable computing devices such as laptops incorporate LCD technology. Because of their slimmer design and lower energy consumption, monitors using LCD technology (also called flat panel or flat screen displays) are replacing the *venerable* CRT on most desktops.

As monitor sizes have increased over the years, display standards and resolutions have changed.

Resolution refers to the number of individual **dots** of color, known as pixels, contained on a display. Resolution is expressed by identifying the number of pixels on the horizontal axis (rows) and the number on the vertical axis (columns), such as 800x600. Resolution is affected by a number of factors, including the size of the screen. In addition to the screen size, display standards and resolutions are related to something called the aspect ratio. Two measures describe the size of your display: the aspect ratio and the screen size. Historically, computer displays, like most televisions, have had an aspect ratio of 4:3. This means that the ratio of the width of the display screen to the height is 4 to 3.

For widescreen LCD monitors, the aspect ratio is 16:9 (or sometimes 16:10 or 15:9). Widescreen LCD displays are useful for viewing DVD movies in wide-screen format, playing games and displaying multiple windows side by side. High definition television also uses a widescreen aspect ratio. All types of displays include a projection surface, commonly referred to as the screen. Screen sizes are normally measured in inches from one corner to the corner diagonally across from it. This diagonal measuring system actually came about because the early television manufacturers wanted to make the screen size of their TVs sound more impressive. Popular screen sizes are 15, 17, 19 and 21 inches. Notebook screen sizes are smaller, typically ranging from 12 to 17 inches. As technologies improve in both desktop and notebook displays, even larger screen sizes are becoming available. For professional applications, such as medical imaging or public information displays, some LCD monitors are 40 inches or larger! Obviously, the size of the display directly affects resolution. The same pixel resolution is sharper on a smaller monitor and *fuzzier* on a larger monitor because the same number of pixels is spread out over a larger number of inches. An image on a 21-inch monitor with an 800x600 resolution will not appear nearly as sharp as it would on a 15-inch display at 800x600. To display information on a monitor, your computer sends the monitor a signal. The signal can be in analog or digital format. Because most CRT monitors require the signal information in analog (continuous electrical signals or waves) form and not digital (pulses equivalent to the binary digits 0 and 1), they

typically use an analog connection. The computer and video adapter convert digital data into analog format. A video adapter is an expansion card or component that provides the ability to convert display information into a signal that is sent to the monitor. It can also be called a graphics adapter, video card or graphics card. Once the display information is in analog form, it is sent to the monitor through a VGA cable. The cable connects at the back of the computer to an analog connector (also known as a D-Sub connector) that has 15 pins in three rows.

Because a VGA (analog) connector does not support the use of digital monitors, the Digital Video Interface standard was developed. DVI keeps data in digital form from the computer to the monitor. There's no need to convert data from digital information to analog information. LCD monitors work in a digital mode and support the DVI format. The DVI specification is based on Silicon Image's Transition Minimized Differential Signaling and provides a high-speed digital interface. A transmitter on the video adapter sends the digital information to a receiver in the monitor. TMDS takes the signal from the video adapter, determines the resolution and refresh rate that the monitor is using, and spreads the signal out over the available bandwidth to optimize the data transfer from computer to monitor.

DVI cables can be a single link cable that uses one TMDS transmitter or a dual link cable with two transmitters. A single link DVI cable and connection supports a 1920x1080 image, and a dual link cable/connection supports up to a 2048x1536 image. The combination of the display modes supported by your graphics adapter and the color capability of your monitor determine how many colors it displays. Liquid crystal display technology works by blocking light. Specifically, an LCD is made of two pieces of polarized glass (also called substrate) that contain a liquid crystal material between them. A backlight creates light that passes through the first substrate. At the same time, electrical currents cause the liquid crystal molecules to *align* to allow varying levels of light to pass through to the second substrate and create the colors and images that you see.

Most LCD displays use active matrix technology. A thin film transistor arranges tiny transistors and capacitors in a matrix on the glass of the display. To address a particular pixel, the proper row is switched on, and then a charge is sent down the correct column. Since all of the other rows that the column intersects are turned off, only the capacitor at the designated pixel receives a charge. The capacitor is able to hold the charge until the next **refresh** cycle.

The other type of LCD technology is passive matrix. This type of LCD display uses a *grid* of conductive metal to charge each pixel. Although they are less expensive to produce, passive matrix monitors are rarely used today due to the technology's slow response time and imprecise voltage control compared to active matrix technology.

A CRT monitor contains millions of tiny red, green, and blue phosphor dots that glow when struck by an electron beam that travels across the screen to create a visible image.

The terms anode and cathode are used in electronics as synonyms for positive and negative terminals. In a cathode ray tube, the "cathode" is a heated *filament*. The heated filament is in a vacuum created inside a glass "tube." The "ray" is a stream of electrons generated by an electron gun that naturally pour off a heated cathode into the vacuum. Electrons are negative. The anode is positive, so it attracts the electrons pouring off the cathode. This screen is coated with phosphor, an organic material that glows when struck by the electron beam.

There are three ways to filter the electron beam in order to obtain the correct image on the monitor screen: **shadow mask**, **aperture grill** and **slot mask**. These technologies also impact the sharpness of the monitor's display.

Advantages of LCD Monitors: require less power, smaller and weigh less, more adjustable, less eye strain.

Advantages of CRT Monitors: less expensive, better color representation, more responsive, multiple resolutions, more rugged.

The basic concepts of plasma display panels are decades old, but only recently have they begun to find commercial use for television. There are many other display technologies, such as ferroelectric liquid crystal, field emission, and vacuum fluorescent, but they have not reached the commercial viability of the CRT, LCD, and PDP. Improvements may well occur in the CRT, renewing the life and utility of this old technology. However, LCDs and PDPs seem more appropriate for the new digital and compression technologies, and so their future in television seems bright. PDPs overcome some of the disadvantages of both CRTs and LCDs. They can be manufactured easily in large sizes (up to 125 cm, or 50 inches, in diagonal size), are less than 10 cm (4 inches) thick, and have wide horizontal and vertical viewing angles. Being light-emissive, like CRTs, they produce a bright, sharply focused image with rich colours. But much larger voltages and power are required for a plasma television screen (although less than for a CRT), and, as with LCDs, complex drive circuits are needed to access the rows and columns of the display pixels. Large PDPs are being manufactured particularly for wide-screen, high-definition television.

The basic principle of a plasma display is similar to that of a fluorescent lamp or neon tube. An electric field excites the atoms in a gas, which then becomes ionized as a plasma. The atoms emit photons at ultraviolet wavelengths, and these photons collide with a phosphor coating, causing the phosphor to emit visible light.

Touch-screen Monitors: Displays with touch-screen technology let you input information or navigate applications by touching the surface of the display. The technology can be implemented through a variety of methods, including infrared sensors, pressure-sensitive resistors or electronic capacitors.

Wireless Monitors: Similar in looks to a tablet PC, wireless monitors use technology such as 802.11b/g to connect to your computer without a cable. Most include buttons and controls for mousing and web surfing, and some also include

keyboards. The displays are battery-powered and relatively lightweight. Most also include touch-screen capabilities.

Television and HDTV Integration: Some displays have built-in television tuners that you can use for viewing cable TV on your computer. You can also find displays that accept S-video input directly from a video device. Additional features include picture-in-picture or picture-on-picture capability, a remote control and support for HDTV.

## LANGUAGE DEVELOPMENT

**4. Find and learn Russian equivalents for the following words and expressions:**

1) liquid crystal display	a)
2) projection surface	b)
3) active matrix technology	c)
4) dual link cable	d)
5) aspect ratio	e)
6) cathode ray tube	f)
7) thin film transistor	g)
8) Digital Video Interface	h)
9) shadow mask	i)

**5. Find and learn English equivalents for the following words and expressions:**

1) вертикальная ось	a)
2) пассивная матрица	b)
3) поток электрических зарядов	c)
4) одножильный кабель	d)
5) телевидение высокой чёткости	e)
6) горизонтальный ряд	f)
7) многооконная визуализация; визуализация "картинка в картинке"	g)
8) частота обновления	h)
9) апертурная решётка	i)
10) щелевая маска	j)

## 6. Find and learn the definitions for the following abbreviations.

1) CRT	a)
2) LCD	b)
3) PDP	c)
4) FED	d)
5) LEP	e)
6) DVI	f)
7) HDTV	g)
8) TMDS	h)
9) TFT	i)
10)SM	j)
11)AP	k)

### KEY CONCEPTS

#### Aperture-grill

Monitors based on the Trinitron technology, which was pioneered by Sony, use an **aperture-grill** instead of a shadow-mask type of tube. The aperture grill consists of tiny vertical wires. Electron beams pass through the aperture grill to illuminate the

#### Display History

Displays have come a long way since the blinking green monitors in text-based computer systems of the 1970s. Just look at the advances made by IBM over the course of a decade:

phosphor on the faceplate. Most aperture-grill monitors have a flat faceplate and tend to represent a less distorted image over the entire surface of the display than the curved faceplate of a shadow-mask CRT. However, aperture-grill displays are normally more expensive.

-In 1981, IBM introduced the Color Graphics Adapter (CGA), which was capable of rendering four colors, and had a maximum resolution of 320 pixels horizontally by 200 pixels vertically.

-IBM introduced the Enhanced Graphics Adapter (EGA) display in 1984. EGA allowed up to 16 different colors and increased the resolution to 640x350 pixels, improving the appearance of the display and making it easier to read text.

-In 1987, IBM introduced the Video Graphics Array (VGA) display system. The VGA standard has a resolution of

640x480 pixels and some VGA monitors are still in use.

-IBM introduced the Extended Graphics Array (XGA) display in 1990, offering 800x600 pixel resolution in true color (16.8 million colors) and 1,024x768 resolution in 65,536 colors.

### **DisplayPort Standard**

The Video Electronics Standards Association (VESA) is working on a new digital display interface for LCD, plasma, CRT and projection displays. The new technology, which is called DisplayPort, supports protected digital outputs for high definition and other content along with improved display performance.

According to VESA, the DisplayPort standard will provide a high-quality digital interface for video and audio content with optional secure content protection. The goal is to enable support for a wide range of source and display devices, while combining technologies. For example, the audio and video signals will be available over the same cable -- a smaller video connector will allow for smaller devices such as notebook computers, and the standard will enable streaming high definition (HD) video and audio content.

**Dot pitch** is an indicator of the sharpness of the displayed image. It is measured in millimeters (mm), and a smaller number means a sharper hon the technology used:

-In a shadow-mask CRT monitor, you measure dot pitch as the *diagonal distance* between two like-colored phosphors. Some manufacturers may also cite a horizontal dot pitch, which is the distance between two-like colored phosphors horizontally.

-The dot pitch of an aperture-grill monitor is measured by the *horizontal distance* between two like-colored phosphors. It is also sometimes are called **stripe pitch**. The smaller and closer the dots are to one another, the more realistic and detailed the picture appears. When the dots are farther apart, they become noticeable and make the image look grainier. Unfortunately, manufacturers are not always upfront about dot pitch measurements, and you cannot necessarily compare shadow-mask and aperture-grill CRT types, due to the difference in horizontal and vertical measurements.

The dot pitch translates directly to the resolution on the screen. If you were to put a ruler up to the glass and measure an inch, you would see a certain number of dots, depending on the dot pitch.

### **Organic Light-Emitting Diode**

OLEDs are thin-film LED (Light-Emitting Diode) displays that don't require a backlight to function. The material emits light when stimulated by an electrical current, which is known as **electroluminescence**. OLEDs consist of red, green and blue elements, which combine to create the desired colors. Advantages of OLEDs include lower power requirements, a less-expensive manufacturing process, improvements in contrast and color, and the ability to bend.

### **Refresh Rate**

In monitors based on CRT technology, the refresh rate is the number of times that the image on the display is drawn each second. If your CRT monitor has a refresh rate of 72 Hertz (Hz), then it cycles through all the pixels from top to bottom 72 times a second. Refresh rates are very important because they control

flicker, and you want the refresh rate as high as possible. Too few cycles per second and you will notice a **flickering**, which can lead to headaches and eye strain. Because your monitor's refresh rate depends on the number of rows it has to scan, it limits the maximum possible resolution. Most monitors support multiple refresh rates. Keep in mind that there is a tradeoff between flicker and resolution, and then pick what works best for you. This is especially important with larger monitors where flicker is more noticeable. Recommendations for refresh rate and resolution include 1280x1024 at 85 Hertz or 1600x1200 at 75 Hertz.

### **Shadow-mask**

It is a thin metal screen filled with very small holes. Three electron beams pass through the holes to focus on a single point on a CRT displays' phosphor surface. The shadow mask helps to control the electron beams so that the beams strike the correct phosphor at just the right intensity to create the desired colors and image on the display. The unwanted beams are blocked or "shadowed."

### **Slot-mask**

A less-common type of CRT display, a **slot-mask tube** uses a combination of the shadow-mask and aperture-grill technologies. Rather than the round perforations found in shadow-mask CRT displays, a slot-mask display uses vertically aligned slots. The design creates more brightness through increased electron transmissions combined with the arrangement of the phosphor dots.

## **! Because & because of**

**Because** is a conjunction. It is used at the beginning of a clause, before a subject and a verb. **Because of** is a two-word preposition, used before a noun or a pronoun.

## **Surface-Conduction Electron Emitter Displays**

SED is a new technology developed jointly by Canon and Toshiba. Similar to a CRT, an SED display utilizes electrons and a phosphor-coated screen to create images. The difference is that instead of a deep tube with an electron gun, an SED uses tiny electron emitters and a flat-panel display.

### **VESA**

It is an organization that supports and sets industry-wide interface standards for the PC, workstation and consumer electronics industries. VESA promotes and develops timely, relevant, open standards for the display and display interface industry, ensuring interoperability and encouraging innovation and market growth. In August of 1992, VESA passed the VESA Local Bus (VL-Bus) Standard 1.0. This standard had a significant impact on the industry, because it was the first local bus standard to be developed, which provided a uniform hardware interface for local bus peripherals. The creation of this standard ensured compatibility among a wide variety of graphics boards, monitors, and systems software.

Today, VESA is a worldwide organization that promotes and develops open display and display interface standards for interoperability. VESA is a formative influence in the PC industry and a contributor to the enhancement of flat panel display, monitor, graphics, software and systems technologies including home networking and PC theatre.

**Because** and its clause can go after or before the main clause. *I finished early because I worked fast.*

*Because I worked fast, I finished early.*

**Because**-clauses can stand alone as answers or after hesitations, but not usually in other cases.

NB! After *reason* we usually use **that** and **why**, NOT **because**.

We do not use **because** and **so** together in the same sentence. To link two clauses, we use just one conjunction.

**Reason** (or **reason why**) should be followed by a **that-clause** (NOT **because**). The meaning of **because** is included in the word **reason**.

**7. Find, underline and translate the examples in the text.**

**8. Find the mistakes and correct the sentence. Explain why.**

1. Because I had forgotten the watch, so I didn't know the time.
2. Because the bad connection, we had problem with downloading files. The reason I didn't take the exam was because I didn't think I would pass.
3. Because of I had no access point I had to buy a new one.

**9. Translate the following sentences into Russian.**

1. The LCD consists of a grid of cells with one cell for each of the three colors (red, green, and blue) for each pixel.
2. Determine which *screen surface* you will need for your projector environment.
3. The *active matrix technology* was invented and first demonstrated by Hungarian emigré T Peter Brody and his Thin-Film Devices department at Westinghouse.
4. The electrons from the guns pass through a lattice called a shadow mask, which keeps the beams properly separated and aligned.
5. *High-definition television* provides a resolution that is substantially higher than that of standard-definition television.

**10. Translate the following sentences into English.**

1. Важность частоты обновления в том, что она позволяет изображению двигаться на экране и влияет на качество картинки.
2. Апертурная решетка является одной из двух главных технологий используемых для производства цветных электронно-лучевых трубок и дисплеев ЭВМ.
3. Многоокопная визуализация является характерной особенностью некоторых телевизоров и подобных устройств.
4. Одножильный кабель использует одну TMDS микросхему, в то время как двухжильный использует две.

5. Коэффициент пропорциональности описывает соотношение размеров [сторон] между шириной и высотой.

## SPEAKING SECTION

### 11. Answer the following questions.

1. What is a display?
2. What do most displays use?
3. How much power does a display use?
4. What is the difference between CRT and LCD?
5. What does "aspect ratio" mean?
6. What is dot pitch?
7. What does "refresh rate" mean?
8. What is SED?
9. What is the purpose of VESA?
10. What is VESA working on now?
11. Name some facts from display history.
12. Name the advantages of each monitor.

### 12. Topics for discussion.

1. How a CRT paints the screen.
2. How an LCD screen works.
3. How plasma displays glow.



13. **Prepare a presentation on the topic being discussed.**

## REFERENCES, USEFUL LINKS AND FURTHER READING

1. 'How computer works', by Ron White, 2008, p.262.
2. <http://computer.howstuffworks.com/>
3. Computer Desktop Encyclopedia copyright ©1981-2012 by The Computer Language Company Inc. p.106.

# UNIT 11

## GRAPHIC DESIGN

### VOCABULARY

#### 1. Match the words with their definitions:

1) encompass (v.)	[ɪn'kʌmpəs], [en'kʌmpəs]	a) a developed by Seurat technique of neo-Impressionist painting using tiny dots of various pure colors, which become blended in the viewer's eye
1) alter (v.)	['ɔ:lteɪ]	b) fragment
2) crisp (adj.)	[krɪsp]	c) marked by exactness and accuracy of expression or detail
3) pointillism (n.)	['pɔɪntɪlɪz(ə)m]	d) sharp
4) versatile (adj.)	['vɜ:sətəɪl]	e) change in character or composition, typically in a comparatively small but significant way: [with obj]
5) splinter (n.)	['splɪntə]	f) [with obj.] surround and have or hold within; accomplish (Syn.)
6) precise (adj.)	[prɪ'saɪs]	g) able to adapt or be adapted to many different functions or activities; Syn: many-sided
7) mascot (n.)	['mæskət]	h) a person or thing that is supposed to bring good luck or that is used to symbolize a particular event or organization

### WATCHING

2. Now watch a video 'The Universal Arts of Graphic Design' and say what you have known about Graphic Design.

#### Before you read

3 .Discuss with your partner the following questions.

- What do you know about graphics editors?
- What are the reasons for inventing new graphics programs?

#### 4. Skim the text to check your ideas.

### READING

## GRAPHICS & IMAGE EDITOR

A graphics editor is a type of computer program that allows a user to edit and manipulate graphical images through a variety of methods. A graphics program provides a variety of special features for *altering* bit-mapped images. The difference between image editors and paint programs is not always clear-cut, but in general image editors are specialized for modifying bit-mapped images, such as scanned photographs, whereas paint programs are specialized for creating images. Image editing *encompasses* the processes of altering images, whether they are digital photographs, traditional analog photographs, or illustrations. Traditional analog image editing is known as photo retouching, using tools such as an airbrush to modify photographs, or editing illustrations with any traditional art medium.

It is not unknown for many image editing programs to be also used to render or create computer art from scratch. Graphic software programs are the primary tools with which a user may manipulate, enhance, and transform images. They can be broadly grouped into vector graphics editors, raster graphics editors, and 3D modelers. The differences between these types usually affects what types of applications the program is used for, and how the final images created appear on a screen or when printed. A graphics editor program may be a free-ware download with only a few features or expensive professional graphic arts software with dozens of tools and features.

Graphics editor programs are typically either vector-based or raster-based, depending on how the program allows a person to access and interact with computer graphics. Vector graphics editors are programs that are usually used for commercial illustration and produce clean edges that hold up to zooming and blowing an image up to great sizes. These programs use mathematical calculations and geometric forms such as lines, shapes, angles, and curves to let a user create visual images through a graphical interface. Graphics editor programs that use vector graphics create *crisp*, clean images because the shapes and lines created through them are *precise*.

Raster graphics editor programs, on the other hand, never use mathematical or geometric measurements as the basis for creating artwork. Instead, a raster graphics program is pixel-based and allows an artist to work with an image by manipulating the base elements of the image. A pixel, or picture element, is typically the smallest basic unit of an image. This can often be seen in the artistic style of *pointillism* or in newsprint which is often made up of hundreds of tiny dots rather than solid lines. On a computer screen, an image usually consists of

tiny points of light called pixels, and these individual units of color and value are used to create larger images.

The use of graphics editor programs that are raster-based is usually limited to photograph manipulation and similar fields, since they allow a user to manipulate color and tone quite easily. Lines created in a raster program, however, are not usually crisp and clean if examined closely. Both types of programs are available from a number of different software companies, and can be found as downloadable freeware or expensive professional software that can cost hundreds of American dollars to purchase and use. These types of graphics editor software are usually used by graphic artists, photographers, and other visual arts professionals. There is a range of freeware (for non-commercial use) Free Editors. Some have basic functionality, for an amateur or a learner; others are very powerful and suitable for the professional.



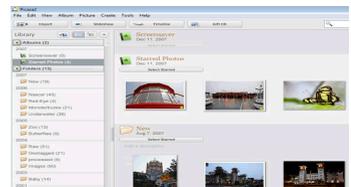
ARTWEAVER is a basic Windows Freeware program to simulate natural brush tools, which is ideal for giving your creativity a free rein. It is available in both English and German. You can create sketches from photos or just experiment with colors. For professional users, Artweaver offers layers, group layers and support for the PSD file format. It is also possible to load and run Photoshop compatible filters.



**IrfanView** is a fundamentally an image viewer, but has a good selection of editing tools that makes this well worth downloading. It is fast, small and compact with a list of supported file extensions that would make any similar commercial software pale in comparison.



This ever popular free application from Google, **PICASA** is basically an application for organizing and editing your digital photos. It offers basic photo editing functions, color enhancement, red eye reduction, cropping, slide shows, printing, optimize for e-mailing and printing (by reducing file size and setting up page layouts).



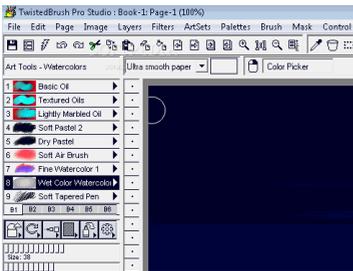


**PIXIA** is a freeware bitmap graphics editor program originally developed for the Japanese Manga but has become very popular with all graphic designers. This Japanese application is also available in English, French, Italian, Spanish, Hungarian, Chinese Simplified, Chinese Traditional, Polish, Korean and

German.

**PROJECT DOGWAFFLE** offers both freeware and commercial versions, the freeware version is fully functional, lacking only the advanced layer and scripting tools of the full version.

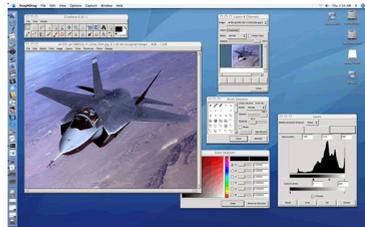
Its features include realistic paint effects, a frame-based animation tool, the standard paint tools common to most modern bitmap paint programs, an alpha channel for transparency effects and support for plug-ins.



**TWISTED BRUSH OPEN STUDIO** edition is a *versatile* and powerful brush engine, it has more than 5,000 brushes available, along with all the features that artists love to use: layers, realistic media, photo cloning, tracing, masks, particles, filters, script recording, scripts to AVI, drawing tablet support, brush shapes, patterns, textures, integrated scanner support, image brushes, drawing

guides, reference image views, dirty brushes, scratch layer, dynamic palettes and a whole lot more. Twisted Brush has been designed for all realms of digital art including natural media fine art, photo retouching, photo composition, photo cloning to turn your photos into the look of hand painted artwork, **mandala** art, patterns, manga, even 3D like painting with the incredible blob modeling tools.

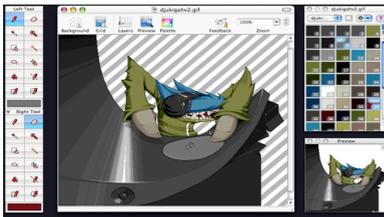
**CINEPAINT** is a professional open-source raster graphics editor. Contrary to belief, it has nothing with a video editor, even though it has been designed for painting and retouching bitmap frames of movie film. It is a *splinter* of GNU Image Manipulation Program, making it very powerful, easy to use and learn.





few limitations. It is very powerful, and is probably easier to use than Photoshop. With its dedicated community you can find all the help, tutorials and plug-ins you would ever need.

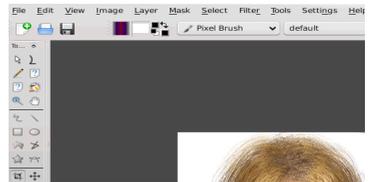
NATHIVE is a free software image editor, similar to Adobe Photoshop, Corel Photo-Paint or GIMP, but focused on usability, logic and providing a smooth learning curve for everyone. The project runs in the GNOME desktop environment and anyone is welcome to collaborate on it with code, translations or ideas. This project is in the alpha phase, so it is an incomplete work, unfit for the end user yet. The intention is to achieve a professional graphic editor progressively without giving up initial usability.



PIXEN is an innovative graphics editor for the Mac. It's designed from top to bottom for pixel artists—people that make low-resolution raster art like the sprites you see in old video games. But it's great for artists of all arenas: Pixen is like a very powerful MSPaint or a simpler, more agile

Photoshop.

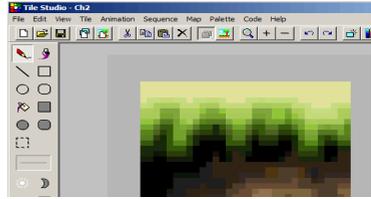
KRITA (Swedish for crayon) is a painting and image editing application for KOffice. Krita contains both ease-of-use and fun features like guided painting and high-end features like support for 16 bit images, CMYK, L\*a\*b and even OpenEXR HDR images.



SEASHORE is an open source image editor for Mac OS X's Cocoa framework. It features gradients, textures and anti-aliasing for both text and brush strokes. It supports multiple layers and alpha channel editing. It is based around the GIMP's technology and uses the same native file format. However,

unlike the GIMP, Seashore only aims to serve the basic image editing needs of most computer users, not to provide a replacement for professional image editing products.

**TILE STUDIO** is a complete development utility for graphics of tile-based games. The application contains a bitmap editor for creating tiles and sprites and a map editor for designing level maps. Tile Studio was originally designed to be used with the Game Library for the language **Clean**. It can be used together with any programming language. You can program Tile Studio to output your maps, animation sequences, bitmaps, palettes, etc. in any format and include the output directly into your source code.



**INKSCAPE** is an open-source vector graphics editor. What sets Inkscape apart is its use of SVG, an open XML-based W3C standard, as the native format. It is not unknown for Inkscape to be used on different projects; it can be successfully used to create web graphics, technical diagrams, icons, creative art, logos and maps.

**TUX PAINT** is a free drawing program for children ages 3 to 12, it combines an easy-to-use interface, fun sound effects, and an encouraging cartoon *mascot* who guides children as they use the program. Kids are presented with a blank canvas and a variety of drawing tools to help them be creative.



**RENDERA** is a free graphics program suitable for artistic painting, photo-retouching, coloring, and seamless tile design. It emulates common art tools and features an easy-to-learn, labeled interface. The edge-wrapping and interactive offset functions make it perfect for Web background tiles.

## LANGUAGE DEVELOPMENT

**5. Find and learn Russian equivalents for the following words and expressions:**

1) bit-mapped images	a)
2) analog image editing	b)
3) from scratch (Idiom )	c)
4) batch conversion	d)
5) blob	e)
6) raster-based	f)
7) PSD file format	g)
8) sprite	h)
9) scene	i)

**6. Find and learn English equivalents for the following words and expressions:**

1) с нуля (идиома)	a)
2) растровая графика	b)
3) раскат тонов	c)
4) штриховка; ретуширование	d)
5) заключение в оболочку, оборка	e)
6) ручная коррекция	f)
7) битовое (растровое) отображение графического объекта	g)
8) мантисса	h)
9) свобода самовыражения	i)

**7. Find and learn the definitions for the following abbreviations.**

1) SVG	a)
2) XML	b)
3) W3C	c)
4) CMYK	d)
5) GNU	l)
6) GIMP	m)
7) PSD	n)

## KEY CONCEPTS

### **Clean Language**

A technique that is used especially in psychotherapy and coaching. It is optimized to have the client discover and develop personal symbols and metaphors, without contamination or distortion of the developing metaphor landscape through the way the questions are put.

It was developed by David Grove in the 1980s as a result of his work on clinical methods for resolving clients' traumatic memories. Clean Language also is the basis of symbolic modeling, a stand-alone method and process for psychotherapy and coaching, which was developed by James Lawley and Penny Tompkins.

### **GIMP**

It means GNU Image manipulation Program. It is a freely distributed piece of software for such tasks as photo retouching, image composition and image authoring. It works on many operating systems and in many languages.

### **ImageForge**

It provides you with a set of powerful tools for painting and editing images, photos or other graphics. Create and edit images, acquire pictures from your scanner, digital camera or other Twain-compliant device, apply special effect filters, produce your own photo albums and simple slide shows, and much more.

### **IrfanView**

Main Features: Many supported file formats, Thumbnail/preview option; Paint option; Create a slide show; Show EXIF/IPTC/Comment text in Slideshow/Fullscreen; Support for Adobe Photoshop Filters; **Batch conversion**; Multipage TIF editing; Support for embedded color profiles in JPG/TIF; Change color depth; Cut/crop; Effects (Sharpen, Blur, Adobe 8BF, Filter Factory, Filters Unlimited, etc.); Screen Capture; Extract icons from EXE/DLL/ICLs.

### **Paint.NET**

It features an intuitive and innovative user interface with support for layers, unlimited undo, special effects, and a wide variety of useful and powerful tools. An active and growing online community provides friendly help, tutorials, and plugins. It was originally intended as a free replacement for Microsoft Paint, it has grown into a powerful yet simple image and photo editor tool, not quite as powerful as Photoshop or Gimp, but a damn such as Adobe Photoshop, Corel Paint Shop Pro, Microsoft Photo Editor, and GIMP.

### **Picasa**

It works hand in hand with Picasa Web Albums (PWA), a photo-sharing web application from Google. If you have a Google account you can store and share 1GB of photos for free.

### **Pixia**

The program supports multiple layers, transparency effects; standard file formats and a number of RGB file formats including .PSD. The main file extension used by this program is .PXA.

### **Project Dogwaffle**

It (current version 1.2) is based on its commercial big brother, Project Dogwaffle 2.0. It's a lighter, yet powerful tool for graphic designers and animators.

### **KolourPaint**

This is a free, raster graphics editor for the K Desktop Environment, similar to Microsoft's Paint. It is a very basic graphic and image editor, built for the amateur designer and photographer.

### **Wrapping (graphics)**

In computer graphics, wrapping is the process of limiting a position to an area. A good example of wrapping is wallpaper, a single pattern repeated indefinitely over a wall. Wrapping is used in 3D computer graphics to repeat a texture over a polygon, eliminating the need for large textures or multiple polygons.

### **! Study the following: *I downloaded the file, but I could find nothing.***

In some languages, a negative word like *nobody*, *nothing*, or *never* has to be used with a negative verb. In Standard English, *nobody*, *nothing*, *never* etc. are themselves enough to give a negative meaning, and *not* is unnecessary.

### **8. Find, underline and translate the other examples in the text.**

**NB!** When making negative sentences you have to remember the word order. We put *not* and *never* in front of the *to* of a *to*-infinitive (not after it). Verbs of thinking and feeling (e.g. *think*, *suppose*, *feel*) are often followed by a *that*-clause. Instead of making the verb in the *that*-clause negative, we normally make *think*, *suppose*, etc., negative. Two common exceptions to this rule are *hope* and *wish*: ***I hope they won't download this file.*** or

***I wish you wouldn't do that.***

**NB!** Only one negative adverb may be used with a verb. Negative adverbs include *not*, *never*, *barely*, *hardly*, *rarely*, *scarcely* and *seldom*. When a sentence begins with a negative meaning (e.g. *hardly*, *not only*), the subject and the verb change places.

### **! Study the following: *It is not unknown for Inkscape to be used on different projects***

A double negative is usually produced by combining the negative form of verb (e.g., *cannot*, *did not*, *have not*) with a negative pronoun (e.g., *nothing*, *nobody*), a negative adverb (e.g., *never*, *hardly*) or a negative conjunction (e.g., *neither/nor*). A statement with a double negative in formal contemporary English usually has an affirmative (= positive) meaning. Often, the positive sense is not what the speaker is trying to say, but a double negative is not always an error. Eg: "It does not unfit for the end user." = "It fits." (Of course, *not unfit* could also mean *average quality*. It depends on context). When used to mean *fit*, the double negative *not unfit* carries a connotation of

the speaker being factual as opposed to complimentary. Double negatives were and are still used as intensifiers of negative statements in informal speech and many dialects of the British Isles and the United States, although they are now considered ungrammatical in formal English.

**9. Find, underline and translate the other examples in the text.**

**10. Find the mistakes and correct the sentence.**

1. Turning on the computer, I was surprised to not hear the faintest sound.
2. I think I can't afford new laptop this year.
3. The box with disks was so heavy that I couldn't hardly lift it.
4. I won't never forget how many problems with that virus I have had.
5. I prefer to not use a mouse mat.
6. Hardly I had sat down when the system was hacked.
7. Not only computers are faster today, but they are also cheaper.

**11. Translate the following sentences into Russian.**

1. It is almost impossible to discuss graphics software without first establishing an understanding of the differences between the two major 2D graphic types: bitmap and vector images.

2. Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs.

3. A straightforward way of wrapping images in Photoshop is to customize presets of the Free Transform tool's Warp option.

4. The OFFSET function returns a reference, based on a row and column number.

5. To create a linear gradient you must define at least two color stops.

**12. Translate the following sentences into English.**

1. Мантисса может означать в математике — дробную часть логарифма числа; в информатике — значащую часть числа плавающей запятой или экспоненциальную запись.

2. Редактор растровых изображений — это компьютерная программа, которая позволяет пользователю рисовать и редактировать картинки на дисплее компьютера и сохранять их.

3. Пакетная обработка — это обработка данных или выполнение программ, при которых элементы пакета обрабатываются или выполняются последовательно без вмешательства оператора.

4. Спрайт, в компьютерной графике это анимация или двумерный образ, интегрированный в более большое фоновое изображение.

5. Большими двоичными объектами обычно являются изображения, аудио или другие объекты мультимедиа, хотя иногда в таком виде хранится исполняемый двоичный код.

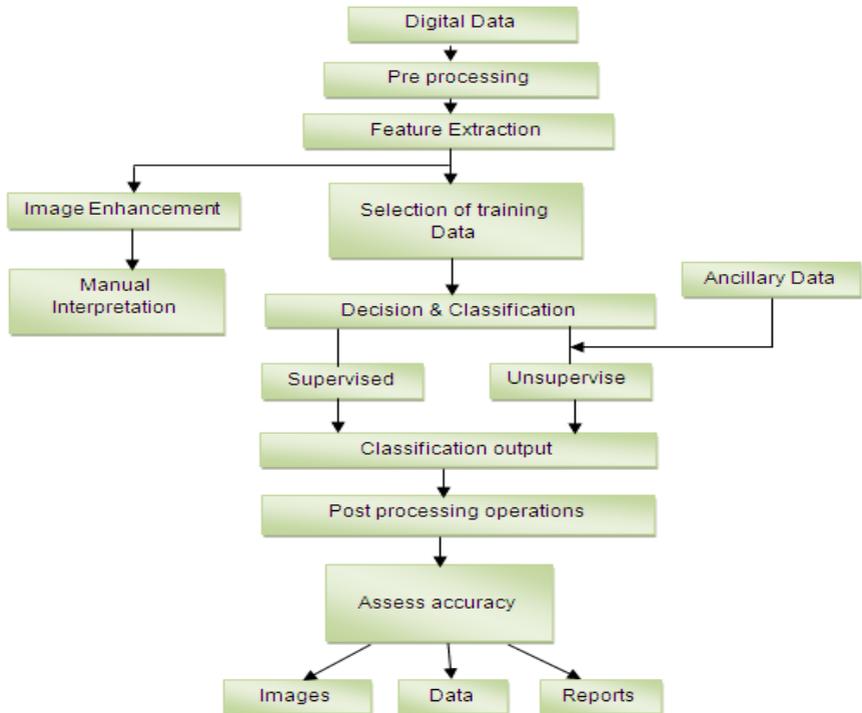
## SPEAKING SECTION

### 13. Answer the following questions.

1. Can you give few examples of computer graphics? Classify applications of computer graphics.
2. What is the meaning of the graphics? Name few types of the graphics.
3. What are the differences between PICT, TIFF, and JPEG images?
4. What is vector graphics? How vector graphics are the rendered over the monitor?
5. What are the advantages of Vector Graphics?
6. What do you understand by image processing? Is image processing a subset of computer graphics?
7. What do you mean by interactive graphics? What are its advantages?
8. What is the conceptual framework for interactive graphics?
9. Give few examples of graphics hardware.
10. Name few graphics libraries you know about.

### 14. Topics for discussion.

1. Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction. Look at the scheme and discuss the following one with your partners. Be ready to describe and explain.



2. History of graphics software applications.
3. Comparison of graphics editors.
4. Compare raster scan display with random scan display.



15. **Prepare a presentation on the topic being discussed.**

## REFERENCES, USEFUL LINKS AND FURTHER READING

1. Computer Graphics: Principles and Practice by James D. Foley, et al.
2. 'How computer works', by Ron White, 2008, p.22.
3. [www.cleanlanguage.co.uk](http://www.cleanlanguage.co.uk)
4. <http://www.grammar-monster.com/glossary/connotation.htm>

## Appendix I

### Standard ASCII Character Set

The first 32 values (0 through 31) are codes for things like carriage return and line feed. The space character is the 33rd value, followed by punctuation, digits, uppercase characters and lowercase characters.

0 NUL	32	64 @	96 `
1 SOH	33 !	65 A	97 a
2 STX	34 "	66 B	98 b
3 ETX	35 #	67 C	99 c
4 EOT	36 \$	68 D	100 d
5 ENQ	37 %	69 E	101 e
6 ACK	38 &	70 F	102 f
7 BEL	39 '	71 G	103 g
8 BS	40 (	72 H	104 h
9 TAB	41 )	73 I	105 i
10 LF	42 *	74 J	106 j
11 VT	43 +	75 K	107 k
12 FF	44 ,	76 L	108 l
13 CR	45 -	77 M	109 m
14 SO	46 .	78 N	110 n
15 SI	47 /	79 O	111 o
16 DLE	48 0	80 P	112 p
17 DC1	49 1	81 Q	113 q
18 DC2	50 2	82 R	114 r
19 DC3	51 3	83 S	115 s
20 DC4	52 4	84 T	116 t
21 NAK	53 5	85 U	117 u
22 SYN	54 6	86 V	118 v
23 ETB	55 7	87 W	119 w
24 CAN	56 8	88 X	120 x
25 EM	57 9	89 Y	121 y
26 SUB	58 :	90 Z	122 z
27 ESC	59 ;	91 [	123 {
28 FS	60 <	92 \	124
29 GS	61 =	93 ]	125 }
30 RS	62 >	94 ^	126 ~
31 US	63 ?	95 _	127 DEL

## APPENDIX II

<p><b>Base names</b></p> <p>1 - <u>unary</u></p> <p>2 - <u>binary</u></p> <p>3 - <u>ternary</u> / trinary</p> <p>4 - <u>quaternary</u></p> <p>5 - <u>quinary</u> / quinary</p> <p>6 - <u>senary</u> / <u>heximal</u> / hexary</p> <p>7 - <u>septenary</u> / septuary</p> <p>8 - <u>octal</u> / octonary / octonal / octimal</p> <p>9 - <u>nonary</u> / novary / noval</p> <p>10 - <u>decimal</u> / <u>denary</u></p> <p>11 - <u>undecimal</u> / undenary / unodecimal</p> <p>12 - <u>dozenal</u> / duodecimal / duodenary</p> <p>13 - <u>tridecimal</u> / <u>tredecimal</u> / triodecimal</p> <p>14 - <u>tetradecimal</u> / quadrodecimal / quattuordecimal</p> <p>15 - <u>pentadecimal</u> / quindecimal</p> <p>16 - <u>hexadecimal</u> / sexadecimal / se-decimal</p> <p>17 - septendecimal / heptadecimal</p> <p>18 - octodecimal / decennoctal</p> <p>19 - nonadecimal / novodecimal / decennoval</p> <p>20 - <u>vigesimal</u> / bigesimal / bidecimal</p> <p>21 - unovigesimal / unobigesimal</p> <p>22 - duovigesimal</p> <p>23 - triovigesimal</p> <p>24 - <u>quadrovigesimal</u> / quadriovigesimal</p> <p>26 - <u>hexavigesimal</u> / sexavigesimal</p> <p>27 - heptovigesimal</p> <p>28 - octovigesimal</p> <p>29 - novovigesimal</p> <p>30 - <u>trigesimal</u> / triogesimal</p> <p>31 - unotrigesimal (...repeat naming pattern...)</p> <p>36 - <u>hexatridecimal</u> / sexatrigesimal (...repeat naming pattern...)</p>	<p>40 - quadragesimal / quadrigesimal</p> <p>41 - unoquadragesimal (...repeat naming pattern...)</p> <p>50 - quinquagesimal / pentagesimal</p> <p>51 - unoquinquagesimal (...repeat naming pattern...)</p> <p>60 - <u>sexagesimal</u> (...repeat naming pattern...)</p> <p>64 - <u>quadrosexagesimal</u> (...repeat naming pattern...)</p> <p>70 - septagesimal / heptagesimal</p> <p>80 - octagesimal / octogesimal</p> <p>90 - nonagesimal / novagesimal</p> <p>100 - centimal / centesimal (...repeat naming pattern...)</p> <p>110 - decacentimal</p> <p>111 - unodecacentimal ( ...repeat naming pattern...)</p> <p>200 - bicentimal / bicesimal (...repeat naming pattern...)</p> <p>210 - decabicesimal</p> <p>211 - unodecabicesimal (...repeat naming pattern...)</p> <p>300 - tercentimal / tricentesimal</p> <p>400 - quattuorcentimal / quadricentesimal</p> <p>500 - quingcentimal / pentacentesimal</p> <p>600 - hexacentimal / hexacentesimal</p> <p>700 - heptacentimal / heptacentesimal</p> <p>800 - octacentimal / octocentimal / octacentesimal / octocentesimal</p> <p>900 - novacentimal / novacentesimal</p> <p>1000 - millesimal</p> <p>2000 - bimillesimal (...repeat naming pattern...)</p> <p>10000 - decamillesimal</p>
--	---

Учебное издание

**Малашенко Елена Александровна**

**ENGLISH FOR IT STUDENTS**

**Part I**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ**

**Редактор**  
**Корректор**

Компьютерная верстка *М. Ю. Мошкова*

Подписано в печать **21.08.2012**. Формат 60×90 <sup>1</sup>/<sub>16</sub>.

Бумага офсетная. Гарнитура Times. Ризография.

Усл. печ. л. 7,25. Уч.-изд. л. 6,97.

Тираж **30** экз. Заказ № **30**.

Издатель и полиграфическое исполнение  
учреждение образования «Международный государственный  
экологический университет имени А.Д.Сахарова»

ЛИ № 02330/993 от 31.08.2011 г.

Республика Беларусь, 220070, г. Минск, ул. Долгобродская, 23

E-mail: [info@iseu.by](mailto:info@iseu.by)

<http://www.iseu.by>