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 II

Environmental Systems & Technologies
In Ecology & Public Health

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 Part II: 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. – 140 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

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.

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

UNIT 1

COMPUTATION

VOCABULARY

1. Match the words with their definitions:

1) relinquish (v.)	[rɪ'lɪŋkwɪʃ]	a) to release
2) raison d'être	[ˌreɪzɔːŋ'detrə]	b) the most important reason or purpose for someone or something's existence
3) polynomial (adj.)	[ˌpɒlɪ'nəʊmɪəl]	c) consisting of several terms
4) hale (v.)	[heɪl]	d) drag or draw forcibly
5) sequential (adj.)	[sɪ'kwɛnʃ(ə)l]	e) forming or following in a logical order or sequence
6) parsing (n.)	['pɑːzɪŋ]	f) the process of analyzing a text, made of a sequence of tokens (for example, words), to determine its grammatical structure with respect to a given (more or less) formal grammar
7) homogeneous (adj.)	[ˌhɔmə'ʤiːnɪəs	g) of the same kind; alike
8) multidimensional (adj.)	[ˌmʌltɪdaɪ'menʃ(ə)n(ə)l]	h) of or involving several dimensions

Before you read

2. Discuss with your partner the following questions.

- **What do you know about algorithms?**
- **What are the reasons for using algorithms?**

3. Skim the text to check your ideas.

READING

ALGORITHMS

When one writes a computer program, one is generally implementing a method of solving a problem, which has been previously devised. This method is often independent of the particular computer to be used: it's likely to be equally appropriate for many computers. In any case, it is the method, not the computer program itself, which must be studied to learn how the problem is being attacked. The term **algorithm** is universally used in computer science to describe problem-solving methods suitable for implementation as computer programs. Algorithms are the "stuff" of computer science: they are central objects of study in many, if not most, areas of the field.

Most algorithms of interest involve complicated methods of organizing the data involved in the computation. Objects created in this way are called data structures, and they are central objects of study in computer science. Thus, algorithms and data structures go hand in hand: data structures exist as the byproducts or endproducts of algorithms, and thus need to be studied in order to understand the algorithms. Simple algorithms can give rise to complicated data structures and, conversely, complicated algorithms can use simple data structures. When a very large computer program is to be developed, a great deal of effort must go into understanding and defining the problem to be solved, managing its complexity, and decomposing it into smaller subtasks which can be easily implemented. It is often true that many of the algorithms required after the **decomposition** are trivial to implement. However, in most cases there are a few algorithms the choice of which is critical since most of the system resources will be spent running those algorithms.

The sharing of programs in computer systems is becoming more widespread, so that while it is true that a serious computer user will use a large fraction of the algorithms, he may need to implement only a somewhat smaller fraction of them. However, implementing simple versions of basic algorithms helps us to understand them better and thus use advanced versions more effectively in the future. Also mechanisms for sharing software on many computer systems often make it difficult to tailor standard programs perform effectively on specific tasks, so that the opportunity to reimplement basic algorithms frequently arises.

Computer programs are often overoptimized. It may be worthwhile to **take pains** to ensure that an implementation is the most efficient possible only if an algorithm is to be used for a very large task or is to be used many times. In most situations, a careful, relatively simple implementation will **branch**: the programmer can have some confidence that it will work, and it is likely to run only five or ten times slower than the best possible version, which means that it may run for perhaps an extra fraction of a second. By contrast, the proper choice of algorithm in the first place can make a difference of a factor of a hundred or a thousand or more, which translates to minutes, hours, days or more in running time.

Often several different algorithms (or implementations) are available to solve the same problem. The choice of the very best algorithm for a particular task can be a very complicated process, often involving sophisticated mathematical analysis. The branch of computer science where such questions are studied is called analysis of algorithms. Many of the algorithms that we will study have been shown to have very good performance through analysis, while others are simply known to work well through experience. We will not dwell on comparative performance issues: our goal is to learn some reasonable algorithms for important tasks. But we will try to be aware of roughly how well these algorithms might be expected to perform. Below are brief descriptions of the basic algorithms.

Algorithms for doing elementary arithmetic operations such as addition, multiplication, and division have a very long history, dating back to the origins of algorithm studies in the work of the Arabic mathematician al-Khowdrizmi, with roots going even further back to the Greeks and the Babylonians. Though the situation is beginning to change, the *raison d'être* of many computer systems is their capability for doing fast, accurate numerical calculations. Computers have built-in capabilities to perform arithmetic on integers and floating-point representations of real numbers; for example, Pascal allows numbers to be of type integer or real, with all of the normal arithmetic operations defined on both types. Algorithms come into play when the operations must be performed on more complicated mathematical objects, such as *polynomials* or matrices. So, mathematical algorithms include fundamental methods from arithmetic and numerical analysis. We study methods for addition and multiplication of integers, polynomials, and matrices as well as algorithms for solving a variety of mathematical problems which arise in many contexts: random number generation, solution of simultaneous equations, data fitting, and integration.

There are several sorting applications in which a relatively simple algorithm may be the method of choice. Sorting programs are often used only once (or only a few times). If the number of items to be sorted is not too large (say, less than five hundred elements), it may well be more efficient just to run a simple method than to implement and debug a complicated method. Elementary methods are always suitable for small files; it is unlikely that a sophisticated algorithm would be justified for a small file, unless a very large number of such files are to be sorted. Other types of files that are relatively easy to sort are ones that are already almost sorted or ones that contain large numbers of equal keys. Simple methods can do much better on such well-structured files than general-purpose methods. So, sorting methods for rearranging files into order are of fundamental importance. A variety of methods are priority queues (e.g. elementary implementations, heap data structure, algorithms on heaps, heap sort, indirect heaps, advanced implementations, selection (finding the kth smallest element (or finding the *k* smallest elements) in a file), and **merging.** Some of these algorithms are used as the basis for other algorithms. Selection and merging are complementary operations in the sense that selection splits a file into two independent files and merging joins two independent files to make one file. The relationship between these operations also becomes evident if one tries to apply the "divide-and-conquer" paradigm to create a sorting method. The file can either be rearranged so that when two parts are sorted the whole file is sorted, or broken into two parts to be sorted and then combined to make the sorted whole file.

Let us name the elementary ones: **selection sort**; **insertion sort** (is slow because it exchanges only adjacent elements); **shell sort**. An elementary sorting method that is often taught in introductory classes is bubble sort: keep passing through the file, exchanging adjacent elements, if necessary; when no exchanges are required on some pass, the file is sorted. **Quicksort** is the sorting algorithm, which is probably more widely used than any other. Quicksort is popular because it's not difficult to implement, it's a good "general-purpose" sort (works well in a variety of situations), and it consumes less resources than any other sorting method in many situations.

The "keys" used to define the order of the records in files for many sorting applications can be very complicated. (For example, consider the ordering function used in the telephone book or a library catalogue.) Because of this, it is reasonable to define sorting methods in terms of the basic operations of "comparing" two keys and "exchanging" two records. Most of the methods we have studied can be described in terms of these two fundamental operations. For many applications, however, it is possible to take advantage of the fact that the keys can be thought of as numbers from some restricted range. Sorting methods, which take advantage of the digital properties of these numbers are called radix sorts. These methods do not just compare keys: they process and compare pieces of keys. Radix sorting algorithms treat the keys as numbers represented in a base-M number system, for different values of M (the radix) and work with individual digits of the numbers. In many applications, records with keys must be processed in order, but not necessarily in full sorted order and not necessarily all at once. Often a set of records must be collected, then the largest processed, then perhaps more records collected, then the next largest processed, and so forth.

An appropriate data structure in such an environment is one, which supports the operations of inserting a new element and deleting the largest element. This can be contrasted with **queues** (delete the oldest) and **stacks** (delete the newest). Such a data structure is called a priority queue. In fact, the priority queue might be thought of as a generalization of the stack and the queue (and other simple data structures), since these data structures can be implemented with priority queues, using appropriate priority assignments. Applications of priority queues include simulation systems (where the keys might correspond to "event times" which must be processed in order), job scheduling in computer systems (where the keys might correspond to "priorities" which indicate which users should be processed first), and numerical computations (where the keys might be computational errors, so the largest can be worked on first).

A fundamental operation intrinsic to a great many computational tasks is **searching**. Searching methods for finding things in files are also of fundamental importance. Normally we think of the information as divided up into records, each record *haling* a key for use in searching. The goal of the search is to find all records with keys matching a given search key. The purpose of the search is usually to access information within the record (not merely the key) for processing. Sequential Searching is the simplest method for searching is simply to store the records in an array, then, look through the array sequentially each time a record is sought. Sequential List Searching is the seqsearch program, which uses purely sequential access to the records, and thus can be naturally adapted to use a linked list representation for the records. Binary Search: if the set of records is large, then the total search time can be significantly reduced by using a search procedure based on applying the "divide-and-conquer" paradigm: divide the set of records into two parts, determine which of the two parts the key being sought belongs to, then, concentrate on that part. Binary tree search is a simple, efficient dynamic searching method, which qualifies as one of the most fundamental algorithms in computer science. A completely different approach to searching from the comparison based tree structures of the last section is provided by **hashing** (Separate Chaining, Open Addressing, Analytic Results): directly referencing records in a table by doing arithmetic transformations on keys into table addresses. If we were to know that the keys are distinct integers from 1 to N, then we could store the record with key *i* in table position *i*, ready for immediate access with the key value. Hashing is a generalization of this trivial method for typical searching application. Several searching methods proceed by examining the search keys one bit at a time (rather than using full comparisons between keys at each step). These methods, called radix searching methods (Digital Search Trees, Radix Search Tries, Multiway Radix Searching, Patricia), work with the bits of the keys themselves, as opposed to the transformed version of the keys used in hashing. As with radix sorting methods, these methods can be useful when the bits of the search keys are easily accessible and the values of the search keys are well distributed. The principal advantages of radix searching methods are that they provide reasonable worst-case performance without the complication of balanced trees; they provide an easy way to handle variable-length keys; some allow some savings in space by storing part of the key within the search structure; and they can provide very fast access to data, competitive with both binary search trees and hashing. The disadvantages are that biased data can lead to degenerate trees with bad performance (and data comprised of characters is biased) and that some of the methods can make very inefficient use of space. These methods are related to each other and similar to sorting methods.

STRING PROCESSING ALGORITHMS include a range of methods for dealing with (long) sequences of characters. Data to be processed often does not decompose logically into independent records with small identifiable pieces. This type of data is characterized only by the fact that it can be written down as a string: a linear (typically very long) sequence of characters.

Strings are obviously central in “word processing” systems, which provide a variety of capabilities for the manipulation of text. Such systems process text strings, which might be loosely defined as sequences of letters, numbers, and special characters. These objects can be quite large, and efficient algorithms play an important role in manipulating them. Another type of string is the binary string, a simple sequence of 0 and 1 values. This is in a sense merely a special type of text string, but it is worth making the distinction not only because different algorithms are appropriate but also binary strings arise naturally in many applications. Several fundamental algorithms have been developed to recognize legal computer programs and to decompose their structure into a form suitable for further processing. This operation, called *parsing* (Context-Free Grammars, Top-Down Parsing, Bottom-Up Parsing, Compilers, Compiler-Compilers) has application beyond computer science, since it is directly related to the study of the structure of language in general. For example, parsing plays an important role in systems which try to “understand” natural (human) languages and in systems for translating from one language to another. One particular case of interest is translating from a “high-level” computer language like Pascal (suitable for human use) to a “low-level” assembly or machine language (suitable for machine execution). A program for doing such a translation is called a compiler. Most files stored on computer systems have a great deal of redundancy and a relatively low “information content”. File compression techniques (Run-length Encoding, Variable-Length Encoding) are often used for text files (in which certain characters appear much more often than others), “raster” files for encoding pictures (which can have large homogeneous areas), and files for the digital representation of sound and other analog signals (which can have large repeated patterns). We looked at methods for encoding strings of characters to save space. Of course, there is another very important reason to encode strings of characters: to keep them secret. Cryptology has many close connections with computer science and algorithms, especially the arithmetic and string-processing algorithms that we have studied. Among the simplest (and among the oldest) methods for encryption is the Caesar cipher; the Vigenere cipher (a small repeated key is used to determine the value of K for each letter); the Vernam cipher, more commonly called the one-time pad.

GEOMETRIC ALGORITHMS (Points, Lines, and Polygons, Line Intersection, Simple Closed Path, Inclusion in a Polygon) comprise a collection of methods for solving problems involving points and lines (and other simple geometric objects) which have only recently come into use. There are algorithms for finding the convex hull of a set of points, for finding intersections among geometric objects, for solving closest point problems, and for *multidimensional* searching. Many of these methods nicely complement more elementary sorting and searching methods.

GRAPH ALGORITHMS are useful for a variety of difficult and important problems. A general strategy for searching in graphs is developed and applied to fundamental connectivity problems, including shortest-path, minimal spanning tree, network flow, and matching.

The study of algorithms is interesting because it is a new field (almost all of the algorithms are less than twenty-five years old) with a rich tradition (a few algorithms have been known for thousands of years).

New discoveries are constantly being made, and few algorithms are completely understood.

LANGUAGE DEVELOPMENT

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

1) general-purpose methods	a)
2) heap data structure	b)
3)divide-and-conquer algorithm	c)
4) insertion sort	d)
5) adjacent elements	e)
6) shell sort	f)
7) bubble sort	g)
8) quicksort	h)
9) string processing	i)
10) variable-length coding	j)
11) a message digest	k)

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

1) структура данных в программировании	a)
2) очередь по приоритету	b)
3) древовидная сортировка, пирамидальная сортировка	c)
4) косвенная/нелинейная динамически распределяемая память	d)
5) сортировка методом выбора (наименьшего или наибольшего	e)

элемента)	
6) поразрядная сортировка	f)
7) продольное кодирование	g)
8) одноразовый (шифровальный) блокнот	h)
9) выпуклая оболочка	i)
10) связующее дерево (алгоритм, позволяющий установить множество параллельных независимых маршрутов между несколькими локальными сетями или сегментами таких сетей)	j)
11) значение хеш-функции	k)

KEY CONCEPTS

algorithm
A procedure, rule, or formula for solving a problem. A computer program is essentially an elaborate algorithm that goes through a series of steps to arrive at a specific outcome.

branch
A sequence of program instructions to which the normal sequence of instructions *relinquishes* control, depending on the value of certain variables.

decomposition
Also known as factoring, refers to the process by which a complex problem or system is broken down into parts that are easier to conceive, understand, program, and maintain.

hash function
It is any well-defined procedure that converts a large, possibly variable-sized amount of data into a small datum, usually a single integer that may serve as an index to an array (cf. associative array). The values returned by a hash function are called *hash values, hash codes, hash sums, checksums* or *simply hashes*.
Hash functions are mostly used to speed up table lookup or data comparison tasks—such as finding items in a database, detecting duplicated or similar records in a large file, finding similar stretches in DNA sequences, and so on. A hash function may map two or more keys to the same hash value. In many applications, it is desirable to minimize the occurrence of such collisions, which means that the hash function must map the keys to the hash values as evenly as possible. Depending on the application, other properties may be required as well.

insertion sort
Inserting element in its proper place among those already considered (keeping them sorted).

merging
Combining two sorted files to make one larger

sorted file.

searching
Retrieving some particular information from a large amount of previously stored information.

selection sort
It works by repeatedly "selecting" the smallest remaining element.

shell sort
A simple extension of insertion sort which gets around this problem by allowing exchanges of elements that are far apart.

stack
A section of memory and its associated registers used for temporary storage of information in which the item most recently stored is the first to be retrieved.

time-staggered queue
a list of data items, commands, etc., stored so as to be retrievable in a definite order, usually the order of insertion regulated by time.

queue
A sequence of stored data or programs awaiting processing. A data structure from which the first item that can be retrieved is the one stored earliest.

quicksort
The basic algorithm was invented in 1960 by C. A. R. Hoare, and it has been studied by many people since that time. It is a "divide-and-conquer" method for sorting. It works by partitioning a file into two parts, then sorting the parts independently.

IDIOMS

take (great) pains = be at pains (synonym) (всячески) стараться, не жалеть сил, прилагать все усилия

6. Translate the following sentences into Russian.

1. The utilities use different algorithms that emphasize storage efficiency at the expense of speed.

2. The compression program uses some variation of a scheme generally called LZ (after its creators, Lempel and Ziv) adaptive dictionary-based algorithm.

3. In 1830, Charles Babbage invented-on paper-the Analytical Engine, which was different from its predecessors because, based on the results of its own computations, it could make decisions such as sequential control, branching, and looping.

4. The software simultaneously repositions the transported pixels and the polygons they form to warp the emerging image so the pixels move steadily toward the positions they occupied in the picture from which they came.

7. Translate the following sentences into Russian.

1. The utilities use different algorithms that emphasize storage efficiency at the expense of speed.

2. The compression program uses some variation of a scheme generally called LZ (after its creators, Lempel and Ziv) adaptive dictionary-based algorithm.

3. In 1830, Charles Babbage invented-on paper-the Analytical Engine, which was different from its predecessors because, based on the results of its own computations, it could make decisions such as sequential control, branching, and looping.

4. The software simultaneously repositions the transported pixels and the polygons they form to warp the emerging image so the pixels move steadily toward the positions they occupied in the picture from which they came.

8. Translate the following sentences into English.

1. Чип хранит характеристики разных музыкальных инструментов в виде набора математических описаний, называемых алгоритмами.

2. Центральный процессор все ставит в режим ожидания, что позволяет записать адрес возврата в стек.

3. Операционная система ставит эту операцию в очередь для последующего выполнения в указанное время (см. регулируемая временем очередь выполнения задач) в соответствии с запросами, которые передаются на запоминающее устройство.

4. Цифровая подпись обычно создается путем вычисления свёртки сообщения или значения хеш-функции.

SPEAKING SECTION

9. Answer the following questions.

1. What is an algorithm?

2. When people refer to a "computer algorithm," what exactly are they talking about?

3. What kinds of problems can be solved by algorithm?

4. How implementation of simple versions of basic algorithms can help?

5. Name sorting algorithms and explain where you can use them.

6. What have you learnt about searching methods?

10.

Prepare a presentation on the topic being discussed.

TEXT FEATURES

A. Organization Of The Text

11. Explain the following terms in italic. Where can we apply them?

a) Introduction > Main Body > Conclusion

b) Abstract > Contents > Introduction > Main Body > Case Study > Discussion > Findings > Conclusion > Acknowledgements > Bibliography/References > Appendices

c) Dedication > Foreword > Preface > Index

12. Text Features are parts of the text that draw your attention to the important information. Look through the table and add other text features and their examples.

Feature	***Function***	***Example***
Abbreviation	to save space	Modern PC graphics are not just about creating pretty pictures.
Italic	to show titles and words from other languages	This process is called *full pivoting;* for forward elimination we only do part of this work hence the process is called *partial pivoting* .
Footnote	to indicate references at the bottom of the page	Gauss-Jordan reduction[1] can be implemented with full pivoting to replace a matrix by its inverse in one sweep through it.
Endnote	to show references at the end of the article or chapter	These effects make interpolating polynomials inappropriate for many curve-fitting applications [1].
Quotation marks	used to draw attention to a phrase, perhaps because it is being used in an unusual or new way	Gaussian elimination with partial pivoting using the largest available pivot is “guaranteed” to produce results with very small computational errors.
Title	To give the reader an idea about what will be read in the book or its section	‘Web-resources’ by M. Jones.

[1]

B. Paragraph

! **A paragraph** is a group of related sentences that discuss one main idea. A paragraph can be as short as one sentence or as long as ten sentences. The number of sentences is unimportant; however, the paragraph should be long enough to develop the main idea clearly.

A paragraph may stand by itself and may also be one part of a longer piece of writing such as an essay or a book. We mark a paragraph by indenting the first word about a half inch (five spaces on a typewriter or computer) from the left margin.

13. The following model contains all the elements of a good paragraph. Read it carefully two or three times. Then answer the Writing Technique questions that follow, which will help you analyze its structure.

We think of the Mona Lisa as a brilliant example of Renaissance art. First of all, the Mona Lisa is a mysterious image. Secondly, we think of it as a thoughtful study in composition, light, and shadow. Therefore, we don't think of it as a mathematical formula. However, in the computer world, all art, graphics, shapes, colors, and lines involve some type of mathematical algorithm. That statement isn't meant to belittle the works of Da Vinci and other great artists.

Mathematical algorithms cannot create art; that still takes a true artist, whether the artist's tools are brush, oils, and canvas or a computerized stylus. But math embedded[2] *in specific file formats can describe any piece of existing art. For example, a graphics-file image of the Mona Lisa that you can display on your PC is the result of mathematical calculations on the bytes of data saved in that file. In conclusion, all the capabilities of a darkroom and an artist's studio are available on a personal computer.*

Writing Technique Questions

1. What is the topic of the paragraph?
2. What two main points does the writer make about the topic?
3. In which two sentences does the writer say that there are two main points?
4. What examples does the writer use to support his idea?

! All paragraphs have **a topic sentence** and **supporting sentences,** and some paragraphs also have **a concluding sentence.**

The **topic sentence** states the main idea of the paragraph, limits the topic to one specific area that can be discussed completely in the space of a single paragraph. The part of the topic sentence that announces the specific area to be dis-

[2] fix (an object) firmly and deeply in a surrounding mass

cussed is called the **controlling idea**. The topic sentence is often, but not always, the first. Notice how the topic sentence of the model states both the topic and the controlling idea:

We think of ***the Mona Lisa (topic)*** *as* ***a brilliant example of Renaissance art. (controlling idea)***

Supporting sentences explain, prove or develop the topic sentence. That is, they explain or prove the topic sentence by giving more information about it. Following are some of the supporting sentences that explain the topic sentence about *the Mona Lisa.*

First of all, the Mona Lisa is a mysterious image.

Secondly, we think of it as a thoughtful study in composition, light, and shadow.

Therefore, we don't think of it as a mathematical formula.

For example, a graphics-file image of the Mona Lisa that you can display on your PC is the result of mathematical calculations on the bytes of data saved in that file.

There are several kinds of specific supporting details: Quotations (e.g. direct quotes, paraphrases, summaries); Examples (e.g. illustrations of your points); Statistics (e.g. facts, figures, diagrams).

Examples make writing lively and interesting, and a reader is more likely to remember your point if you support it with a memorable example. Words and phrases that introduce examples include*for example, for instance,* and *such as.*

! For more information on using **statistics** in academic writing, the handouts and resources may be helpful on the following site: http://writingcenter.waldenu.edu/902.htm.

When you use a direct **quote**, you copy and reference the exact word/s of the author into your writing. A direct quote may be: one word; a phrase or part of a sentence; a sentence; a group of sentences.

! When you decide to use the EXACT words of an author in your writing, you will need to consider whether you want to use only a few words (**short quote** from **one word to about 40 words**) or a longer chunk of text (**long quote - more than 40 words OR three typed lines**).

There are different rules for using quotes according to the length of the quote.

Short direct quotes

1) use double quotation marks "..."
2) include the quote in the text by using reporting words

Long quotes

1) leave no space above and below the long quote
2) make the text size the same as the essay text size
3) indent approximately one centimetre to the right
4) do NOT use quotation marks

The **concluding** sentence signals the end of the paragraph and leaves the reader with important points to remember. You can do this by summarizing the main points of the paragraph or by repeating the topic sentence in different words.

In conclusion, all the capabilities of a darkroom and an artist's studio are available on a personal computer.

Concluding sentences are customary for stand-alone paragraphs. However, a concluding sentence is not needed for every paragraph in a multiparagraph essay. You can use the following concluding signals:

End-of-paragraph Signals with a Comma		End-of-paragraph Signals without a Comma
In conclusion, In brief, In short, In summary,	Finally, Lastly, To sum up, Indeed	We can see that … The example shows that … No doubt that …

NB! DO NOT use the phrase *At last* as an end-of-paragraph signal. *At last* means "at the end of a long period of time."

! Well-organised paragraphs not only help readers understand the argument; they also help writers to structure their ideas effectively. Introductory paragraphs often contain **definitions**, and descriptive paragraphs include a lot of **detail**. Other sentences give **examples** and offer **reasons** and **restatements**.

The structure of the paragraph is:

1. Topic sentence
2. Reason
3. Example
4. Details
5. Further details
6. Reason

14. The sentences below make up a paragraph, but have been mixed up. Use the table to re-write the sentences in the correct order.

a) An algorithm is a fixed set of operations that change data in a way that makes the original document incomprehensible.

b) The key to decrypting it would be "shift one letter to left."

c) A person who wants to send another person a confidential document encrypts the file using the public key as a variable in the algorithm used by the software.

d) A simple example of an algorithm is "shift one letter to the right," so that HAL becomes IBM.

Topic	
Detail	
Example	
Reason	

15. You are writing an essay on 'Algorithms'. Using the notes below, complete the introductory paragraph, following the structure provided.

The term algorithm is universally used in computer science.

Algorithms are the "stuff" of computer science.

There are many 'fundamental' algorithms which are important to know and interesting to study.

There are a few algorithms the choice of which is critical.

Essay aims to consider a large number of the most important algorithms used on computers today.

detail	The term algorithm
reason	Algorithms are the "stuff"
detail	There are many 'fundamental' algorithms
detail	However, in most cases there are a few algorithms
topic	The essay attempts to cover

C. Introduction

! **Introduction** is an important part of your work. Unless you can introduce the subject clearly, the reader may not wish to continue.

16. What is the purpose of the introduction to an essay? Choose from the items below:

a) to define some of the terms in the title
b) to give your opinion of the subject
c) to show that you have read some research on the subject
d) to show that the subject is worth writing about
e) to explain which areas of the subject you will deal with
f) to get the reader's attention with a provocative idea
g) to show how you intend to organise your essay

! It may be necessary to clarify some of the words in the title. This is to make it clear that you understand the title. *Discuss the most important algorithms in use on computers today.*

Algorithm is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a modern computer.

! In longer assignments it is important to show that you are familiar with current research.

This can be demonstrated using phrases such as:

A number of researchers have examined this issue, notably ...

Various investigations have explored the subject, especially ...

! You must show the importance of the topic. This can be either in the academic word or as a contemporary issue of wider relevance.

As algorithm is increasingly important, it is worth spending enough time on each algorithm to understand its essential characteristics and to respect its subtleties.

! There is no such thing as a standard introduction, and much depends on the nature of the research and the length of the essay. However, for a relatively short essay the following are worth including, in this order.

a) Definitions of any terms in the title that are unclear
b) Some background information
c) Reference to other writers who have discussed this topic
d) Your purpose in writing and the importance of the subject
e) The points you are going to make/areas you are going to cover

17. Write an introduction (about 100 words) to an essay on a subject from your own discipline.

C. Organizing the Main Body

! The structure of the main body depends on the length of the essay and the subject of study.

Longer essays may include the following sections:

Experimental set-up – a technical description of the organization of an experiment.

Methods – how the research was carried out.

Findings/results – what was discovered by the research/experiment.

Case study – a description of an example of the topic being researched.

Discussion – an examination of the issues and the writer’s verdict.

! Inside the main body, ideas need to be presented in the most logical fashion, linked together to form a coherent argument. It is useful to mark the beginning of new paragraphs or the introduction of new subjects with special phrases.

To introduce a new paragraph/topic:	To add information inside a paragraph:
The main/chief factor/issue is ... *Turning to the subject of ...* *Moving on to the question of ...* *Another important area is ...* *………… must also be examined* *Even if…* *It is wise to think…*	a) Firstly, ... The first point ... In the first place ... b) Secondly, ... Next, ... Then, ... In addition … Moreover …, On the other hand,… Also,… Similarly,… c) Finally, ... Lastly,...

18. Complete with suitable phrases the following extract from an essay on 'Data structure'.

_________ there are no terms with zero coefficients in a polynomial or no zero elements in a matrix, an advantage of the linked list representation is that we don't need to know in advance how big the objects that we'll be processing are. This is a significant advantage that makes linked structures preferable in many situations. _________, the links themselves can consume a significant part of the available space, a disadvantage in some situations_________, access to individual elements in linked structures is much more restricted than in arrays.

We will see examples of the use of these data structures in various algorithms, and we'll see more complicated data structures that involve more constraints on the elements in an array or more pointers in a linked representation. _________, multidimensional arrays can be defined which use multiple indices to access individual items. _________, we will encounter many "multidimensional" linked structures with more than one pointer per node. The tradeoffs between competing structures are usually complicated, and different structures turn out to be appropriate for different situations.

When possible _________of the data and the specific operations to be performed on it as an abstract data structure which can be realized in several ways. _________, the abstract data structure for polynomials in the examples above is the set of coefficients: a user providing input to one of the programs above need not know whether a linked list or an array is being used. Modern programming systems have sophisticated mechanisms, which make it possible to change representations easily, even in large, tightly integrated systems.

19. Write the main body (about 100 words) to an essay on a subject from your own discipline.

D. Conclusion

! There is usually a link between the starting point, i.e. the title, and the conclusion. If the title is asking a question, the conclusion should contain the answer. The reader may look at the conclusion first to get a quick idea of the main arguments or points. In most cases it is helpful for the reader to have a section that (quite briefly) looks back at what has been said and makes some comments about the main part.

20. The following may be found in conclusions. Decide on the most suitable order for them (1-5).

Implications of the findings
Proposals for further research
Limitations of the research
Reference to how these findings compare with other studies
Summary of main findings

! Conclusion paragraphs are about **5% of your essay word count** (e.g. about 50 or so words per 1000 word essay). In clearly-written sentences, you **restate the thesis** from your introduction (but do not repeat the introduction too closely), make a **brief summary** of your evidence and **finish** with some sort of judgment about the topic. It's a good idea to start your conclusion with transitional words (e.g. 'In summary', 'To conclude', 'In conclusion', 'Finally',) to help you to get the feel of wrapping up what you have said. The conclusion is not the place to present new facts (should be in the body of your essay), so conclusions don't usually have references unless you come up with a 'punchy' quote from someone special as a final word.

21. Read the following extracts from the conclusion and match them with the list of functions in the box. Decide on the most suitable order for them.

a) As always, this investigation has a number of limitations to be considered in evaluating its findings.

b) This might defy the very principle of GAs that it is ignorant of the problem domain when used to solve problem.

c) But we would realize that this practice could make GAs even more powerful.

d) These applications of the algorithms, be they commercial, educational and scientific, are increasingly dependent on these algorithms, the Genetic Algorithms (GAs).

e) These results of the Genetic Algorithms study reported here are consistent with other similar studies conducted in other countries (Dan Sperber & Deirdre Wilson, 1988).

f) The discussion is far from conclusive and, whether artificial life will become real life, will remain to be seen.

g) We believe that, through working out these interesting examples, one could grasp the idea of GAs with greater ease.

h) Its usefulness and gracefulness of solving problems has made it the a more favourite choice among the traditional methods, namely gradient search, random search and others.

i) GAs are very helpful when the developer does not have precise domain expertise, because GAs possess the ability to explore and learn from their domain.

j) In this report, we have placed more emphasis in explaining the use of GAs in many areas of engineering and commerce.

k) We have also discussed the uncertainties about whether computer generated life could exist as real life form.

l) In future, we would witness some developments of variants of GAs to tailor for some very specific tasks.

m) To be more precise, this might defy the very principle of GAs that it is ignorant of the problem domain when used to solve problem

Summary of main body	
Reference to how these findings compare with other studies	
Limitations of research	
Suggestions for further research	
Practical implications and proposals	

22. Write a conclusion (about 100 words) to an essay on a subject from your own discipline.

REFERENCES, USEFUL LINKS AND FURTHER READING

1. ‘Academic Writing’, by Malashenko, Elena, 2007.

2. ‘Algorithms’, by Sedgewick, Robert, 1984. Library of Congress Cataloging in Publication Data, p.91-560.

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

4. ‘Context Generation in Information Retrieval, by Ian Ruthven and C J van Rijsbergen, Department of Computing Science, University of Glasgow, Glasgow, G12 8QQ.

5. How computer works’, by Ron White, 2008, pp.94.

UNIT 2

COMPUTER MODELLING

VOCABULARY

1. Match the words with their definitions:

1) mesh (n.)	[meʃ]	a) increase rapidly in number; multiply
2) thrust (n.)	[θrʌst]	b) make determined efforts to deal with (a problem or difficult task
3) immerse (v.)	[ɪ'mɜːs]	c) a limitation or restriction
4) tackle (v.)	['tækl]	d) a wire frame)
5) constraint (n.)	[kən'streɪnt]	e) susceptibility to injury or attack
6) proliferate (v.)	[prə'lɪf(ə)reɪt]	f) the principal purpose or theme of a course of action or line of reasoning
7) vulnerability (n.)	[ˌvʌln(ə)rə'bɪlətɪ]	g) to involve in smth

Before you read

2. Discuss with your partner the following questions.

- **What do you know about computer modeling?**
- **What are the reasons for using computer modeling?**

3. Skim the text to check your ideas.

READING

THE COMPUTER MODELING PROCESS

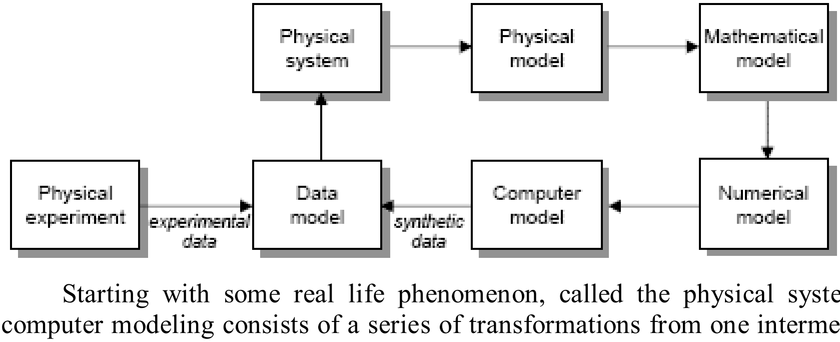
Modeling, in the technical use of the term, refers to the translation of objects or phenomena from the real world into mathematical equations. Computer modeling is the representation of three-dimensional objects on a computer, using some form of software designed for the purpose. Among the uses of computer modeling are war games and disaster simulations, situations in which computers offer a safe, relatively inexpensive means of creating or re-creating events without the attendant loss of life or property.

Mathematical modeling dates to advances in geometry and other disciplines during the late eighteenth century. Among these was the descriptive geometry of French mathematician Gaspard Monge, whose technique was so valuable to Napoleon's artillery that it remained a classified defense secret for many years. Nearly one and a half centuries later, at the end of World War II, mathematicians and scientists working for the United States war effort developed a machine for readily translating mathematical models into forms easily grasped by non-mathematicians.

That machine was the computer, and during the last two decades of the twentieth century, varieties of three-dimensional modeling software *proliferat*-*e*d. These included any number of computer animation and gaming packages, as well as varieties of CAD/CAM systems. CAD allowed engineers and architects, for instance, to create elaborate models that allowed them to "see into" unbuilt structures, and to test the *vulnerabilities* of those structures without risking lives or dollars.

One notable variety of three-dimensional software is virtual reality modeling language, abbreviated VRML and pronounced "ver-mal." Necessary for representing three-dimensional objects on the World Wide Web (that portion of the Internet to which general users are most accustomed), VRML creates a virtual world, or hyperspace, that can be viewed through the two-dimensional computer screen. By pressing designated keys, the user is able to move not only up, down, right, and left, but forward and backward, within this virtual world.

Scientists seek to understand nature by using a mix of theory, experimentation, and **computer modeling**. Theorists explain things using mathematical models such as partial differential equations. Experimentalists measure natural phenomena and collect and analyze the resulting experimental data. Computer modelers develop computer programs that produce synthetic data, which can then be collected and analyzed.

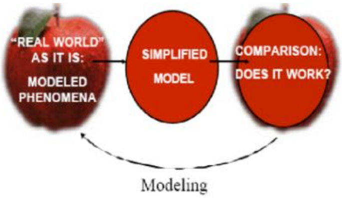


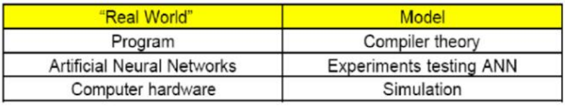
Starting with some real life phenomenon, called the physical system, computer modeling consists of a series of transformations from one intermediate model to another (see on the left). The physical system is first abstracted into a physical model that identifies the scale and scope of the phenomenon of interest. The physical model is then transformed into some mathematical model, typically continuous partial differential equations, that often do not have closed form solutions. In these cases, the mathematical model must be transformed into some discrete numerical model (such as a *mesh*) that can be solved on a computer. The numerical model is then transformed into a computer model, which is a program (often called a **solver** or a simulation) that produces synthetic data that approximates the behavior of the physical system. Finally, the raw synthetic data from the solver is validated against experimental data, reduced, and transformed into a more compact representation (such as a graph or an image) that hopefully, provides the modeler with some insight into the original physical system.

Scientific computing is the research area that *tackles* problems associated with the activities along the bottom row of figure above: (a) building numerical models, (b) building computer models, and (c) querying and analyzing experimental and synthetic data. By nature, scientific computing is broad and multidisciplinary, drawing on aspects of physics, applied mathematics, algorithms, computational geometry, parallel and distributed computing, database systems, graphics, signal processing, graphics, and visualization.

Modeling is a process that always occurs in science, in a sense that the phenomenon of interest must be simplified, in order to be studied. That is the first step of abstraction. A model has to take into account the relevant features of a phenomenon. It obviously means that we are supposed to know which features are relevant. That is possible because there is always some theoretical ground that we start from when doing science.

A simplified model of a phenomenon means that we have a sort of description in some symbolic language, which enables us to predict observable/measurable consequences of given changes in a system. Theory, experiment and simulation are all about (more or less detailed) models of phenomena. Sometimes there are some special *constraints* put on models such as e.g. required conservatism (a consequence of general **Precautionary principle**). Conservative models are made in safety related systems. It means that it must be assumed that uncontrolled parameters (those not explicitly modelled, or those outside the modeling system) have their worst (most unfavorable) credible value.

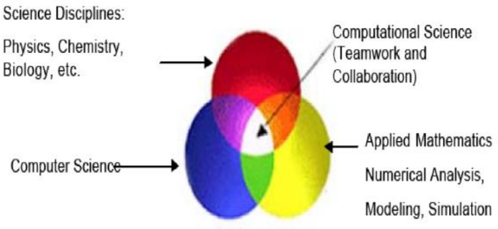




"Real World"	Model
Program	Compiler theory
Artificial Neural Networks	Experiments testing ANN
Computer hardware	Simulation

It is always necessary to "benchmark" new models against old models in known specific cases and analyze their relative strengths/ weaknesses. It is the part of Comparison: Does it work?

In recent years computation which comprises computer-based modeling and simulation has become the third research methodology within Computational Science (CS), complementing theory and experiment.


Science Disciplines:
Physics, Chemistry, Biology, etc.
Computational Science (Teamwork and Collaboration)
Applied Mathematics
Numerical Analysis, Modeling, Simulation
Computer Science

Computational Science

CS has emerged, at the intersection of Computer Science, applied mathematics, and science disciplines in both theoretical investigation and experimentation.

Mastery of CS tools, such as modeling with 3D visualization and computer simulation, efficient handling of large data sets, ability to access a variety of distributed resources and collaborate with other experts over the Internet, etc. are now expected of university graduates, not necessarily CS majors. Those skills are becoming a part of scientific culture.

Today, computing environments and methods for using them have become powerful enough to tackle problems of great complexity. With the dramatic changes in computing, the need for dynamic and flexible Computational Science becomes ever more obvious.

Computer simulation makes it possible to investigate regimes that are beyond current experimental capabilities and to study phenomena that cannot be replicated in laboratories, such as the evolution of the universe. In the realm of science, computer simulations are guided by theory as well as experimental results, while the computational results often suggest new experiments and theoretical models. In engineering, many more design options can be explored through computer models than by building physical ones, usually at a small fraction of the cost and elapsed time. Simulations such as the galaxy formation studies on the left can only be conducted on very powerful computers. Science often proceeds with bursts of intense research activity. Even though the term "simulation" is old, it reflects the way in which a good deal of science will be done in the next century. Scientists will perform computer experiments in addition to testing scientific hypotheses by performing experiments on actual physical objects of investigation. One can also say that simulation represents a fundamental discipline in its own right regardless of the specific application.

Computational science involves the use of computers ("supercomputers") for visualization and simulation of complex and large-scale phenomena. Studies involving N body simulations, molecular dynamics, weather prediction and finite element analysis are within the *thrust* of computational science. If Computer Science has its basis in computability theory, then computational science has its basis in computer simulation. In the Key Concepts section you can read about some of the key focus areas of the past we have taken to shed light on the potential or existing role that simulation plays in each of them: **Chaos and Complex Sys-**

tems; Virtual Reality; Artificial Life; Physically Based Modeling and Computer Animation.

The computing power of present day machines enables us to simulate an increasing number of phenomena and processes; especially the non-linear ones. Modern graphic capabilities makes this method a very attractive and user friendly. For example, researchers from A*STAR's Institute of Materials Research and Engineering (IMRE) have developed an innovative method for creating sharp, full-spectrum colour images at 100,000 dots per inch (dpi), using metal-laced nanometer-sized structures, without the need for inks or dyes. The inspiration for the research was derived from stained glass, which is traditionally made by mixing tiny fragments of metal into the glass. It was found that nanoparticles from these metal fragments scattered light passing through the glass to give stained glass its colours. Using a similar concept with the help of modern nanotechnology tools, the researchers precisely patterned metal nanostructures, and designed the surface to reflect the light to achieve the colour images. "Instead of using different dyes for different colours, we encoded colour information into the size and position of tiny metal disks. These disks then interacted with light through the phenomenon of plasmon resonances," said Dr Joel Yang, the project leader of the research. "The team built a database of colour that corresponded to a specific nanostructure pattern, size and spacing. These nanostructures were then positioned accordingly. Similar to a child's 'colouring-by-numbers' image, the sizes and positions of these nanostructures defined the 'numbers'. But instead of sequentially colouring each area with a different ink, an ultrathin and uniform metal film was deposited across the entire image causing the 'encoded' colours to appear all at once, almost like magic!" added Dr Joel Yang. In comparison, current industrial printers such as inkjet and laserjet printers can only achieve up to 10,000 dpi while research grade methods are able to dispense dyes for only single colour images. The researchers from IMRE had also collaborated with A*STAR's Institute of High Performance Computing (IHPC) to design the pattern using computer simulation and modeling. Dr Ravi Hegde of IHPC said, "The computer simulations were vital in understanding how the structures gave rise to such rich colours. This knowledge is currently being used to predict the behaviour of more complicated nanostructure arrays." The researchers are currently working with Exploit Technologies Pte Ltd (ETPL), A*STAR's technology transfer arm, to engage potential collaborators and to explore licensing the technology. The research was published online on August 12, 2012 in *Nature Nanotechnology*.

LANGUAGE DEVELOPMENT

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

1) elapsed time	a)
2) non-linear process	b)
3) nanometer-sized structures	c)
4) plasmon resonances	d)
5) uniform metal film	e)
6) descriptive geometry	f)
7) computer-aided manufacturing	g)
8) designated keys	h)

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

1) металло-решетчатые структуры	a)
2) витражное стекло	b)
3) монохромное изображение	c)
4) наноструктурная решетка	d)
5) трёхмерные объекты	e)
6) автоматизированное проектирование	f)
7) язык моделирования виртуальной реальности	g)
8) вычислительные науки	h)

KEY CONCEPTS

Artificial Life
AL is an outcome of computational science that challenges our definition of the term experiment. An experiment in artificial life is one where a computer program is written to simulate artificial life forms, often carrying along metaphors such as genetic reproduction and mutation.

Chaos and Complex Systems
The idea that one can observe complexity within a structurally simple deterministic model is of fundamental interest. Qualitative topological phase space features of linear systems may be determined statically, but for non-linear systems simulation must be used.

computer modeling
The use of a computer to develop a mathematical model of a complex system or process and to provide conditions for testing it.

computer simulation
The technique of representing the real world by a computer program; "a simulation should imitate the internal processes and not merely the results of the thing being simulated".

solver
A generic term indicating a piece of mathematical software, possibly in the form of a stand-alone computer program or as a software library, that 'solves' a mathematical problem. A solver takes problem descriptions in some sort of generic form and calculate their solution. In a solver, the emphasis is on creating a program or library that can easily be applied to other problems of similar type.

Physically Based Modeling and Computer Animation
Within computer graphics, there has been a noticeable move forward in the direction of physically based modeling (constraint based models derived from physical laws).

Precautionary principle
It states that if an action or policy has a suspected risk of causing harm to the public or to the environment, in the absence of scientific consensus that the action or policy is harmful, the burden of proof that it is not harmful falls on those taking the action.

Virtual Reality
It means to *immerse* the analyst within the simulated world. Although, it is often seen as being synonymous with man-machine hardware interfaces, the technology must incorporate methods for building dynamic digital (virtual) worlds, which is a typical problem of computer simulation.

6. Translate the following sentences into Russian.

1. After the space shuttle Columbia crashed on February 1, 2003, analysts at the National Aeronautics and Space Administration (NASA) used modeling software applied by the National Transportation Safety Board (NTSB) for studying crashes.

2. The purpose of simulations produced by ICT and others involved in computer modeling goes far beyond mere entertainment: at a fraction of the expense and risk involved in war games involving real troops and equipment, commanders and their subordinates can study and learn from battle.

3. This article describes a method for estimating parameters of a reflection model from a single color image taken by a CCD camera.

4. There is another type of computer keyboard, which is designed to make things go a bit "smoother" on your fingers and makes typing easier, which is suggested for people who are having a hard time learning how to type or those who use the designated fingers on the designated keys.

5. One way to represent an elapsed time is with a simple arithmetic data type, as with the following function to compute the elapsed time between two calendar times.

7. Translate the following sentences into English.

1. Монокристалические и одномерные наноструктуры диоксида титана были бы намного предпочтительны так как обладают изумительными свойсвами и характеристиками, такими как высокая подвижность электронов, максимальную удельную поверхность и даже высокую механическую прочность.

2. Любой объект, который вы видите или трогаете, имеет трехмерную структуру и может быть измерен: длина, ширина и высота.

3. Группа исследователей из Сингапура, используя металло-решетчатую наноструктуру, сумела получить в несколько раз отчетливое цветное изображение высокого разрешения, чем при обычных методах.

4. Система автоматизированного проектирования, известная также как автоматизированное проектирование и черчение, это использование компьютерных систем для содействия в создании, трансформации, анализа или оптимизации дизайна.

5. Язык моделирования виртуальной реальности позволяет создать «виртуальные миры» объединённые в сеть посредством Интернета и соединенные гиперссылками со всемирной паутиной.

SPEAKING SECTION

8. Answer the following questions.

1. What are the advantages and disadvantages of computer modeling?

2. What is the difference between computer modeling and computer simulations?

3. What are the advantages and disadvantages of computer simulations?

4. How to model? What to take into account? What to neglect?

5. Do we need to describe neglected features indirectly? Why/Why not?

6. What formalism to use in modeling?

7. Is the model appropriate? Does it serve its purpose? Why/Why not?

8. A model is always done with a certain “resolution”. Do we have right level of abstraction? Why/Why not?

9. In what aspects does the features/behavior of the model differ from what is expected?

10. In what way does model differ from “reality” (other existing models or experimental results)?

11. Validation: are the results valid (for the presumptive purpose)? Why/Why not?

9. Prepare a presentation on the topic being discussed.

BACKGROUND TO WRITING

ELEMENTS OF WRITING (1)

A. Cause and Effect

! The relationship between two situations can be shown in a variety of ways:

CAUSE: heat ----------------------------EFFECT: errors
The heat *causes* errors.
The heat *leads to* errors.
The heat *results in* errors.
The heat *produces* errors.
EFFECT: errors----------------------------CAUSE: heat
Errors *are caused by* heat (NB! Passive).
Errors *are produced by* heat.
Errors *results from* heat.

! It is also possible to use conjunctions that demonstrate cause and effect.

Cause	**Effect**
because (of)	so
since	therefore
as	consequently
owing to	which is why
due to	

Because the heat increased, the errors were severe. (*because* + verb)
The errors occurred *because of* increase of heat. (*because* + noun)
Owing to the heat the errors were severe.

The collisions of electrons moving through the traces and wires of microchips and circuit boards generated heat, *therefore* the errors were severe. (used in midsentence)

NB! It is more common to use conjunctions to illustrate particular situations.

10. Complete the following sentences with a suitable verb or conjunction.

1. Lotus 1-2-3 was the first big business hit on the IBM PC __________ it combined the functions of an electronic spreadsheet, a database, and graphing.

2. The number of sectors and tracks and, __________, the number of clusters that a drive can create on a disk's surface, determine the capacity of the disk.

3. In Europe, an A-Iaw codec uses a different coding scheme, __________ U.S. 56K modems only reach their full potential communicating with servers in the U.S.

4. In the early days of personal computers, when people thought it was simply a revolution rather than a fundamental change in our existence, someone came up with the idea that all this computerized data would __________ the " paperless office.

5. Leaving the University of Pennsylvania __________ disagreements about patent ownership, J. Presper Eckert and John Mauchly-the two men who headed the ENIAC project-launch the first commercial computer company, Electronic Control Company.

6. They were simply too crude __________ more subtle flows of electrons.

7. All microchips are basically vast collections of transistors-switches- arranged in different patterns __________ that they accomplish different tasks.

11. Write three more sentences from your own subject area.

B. Cohesion

! Cohesion means linking phrases together so that the whole text is clear and readable. It is achieved by several methods, such as the use of conjunctions and reference words and phrases, which refer back to something mentioned before.

pronouns	he/she/it/they
objective pronouns	her/him/them
other phrases	the former/the latter/the first/the second
possessive pronouns	his/her/hers/their/theirs
demonstrative pronouns	this/that/these/those

12. Read the following paragraph and complete the table.

Ken Thompson (1943) was the principal inventor of UNIX. *It* is a multiuser, multitasking operating system. *He* received *his* Bachelor's and Master's degrees, both in electrical engineering, from the University of California at Berkeley (UCB). Soon thereafter, in 1966, he was hired by Bell Labs. The latter appears to be a research and development arm of AT&T. In 1973, Thompson made his first public presentation about UNIX. It was noticed by the right people at UCB, and this led to the first copy of the operating system being shipped to that university.

The UNIX source code is distributed freely throughout the 70s, and *it* soon becomes popular at universities and research labs.

Reference	Reference word/phrase
Ken Thompson s	
UNIX	
The UNIX source code	
first public presentation	
the latter…	
noticed	

C. Definitions

! Definitions are normally needed in two situations:

a) In introductions, to clarify a word or phrase in the title.

b) More generally, to explain a word or phrase that may be either very technical (and so not in normal dictionaries), or very recent, or with no widely agreed meaning.

Word	Category	Detail	Use
3D modeling	is the process	of developing a mathematical representation of any three-dimensional surface of object,	used in a computer simulation of physical phenomena

13. Insert suitable category words in the following definitions.

1. *Computer animation* is the______ used for generating animated images by using computer graphics.

2. *Descriptive geometry* is the ______of geometry, which allows the representation of three-dimensional objects in two dimensions.

3. *Virtual reality* (VR) is a _______ that applies to computer-simulated environments.

4. *VRML* is a standard __________ for representing 3-dimensional (3D) interactive vector graphics.

5. *Computer-aided manufacturing* (CAM) is the __________ of computer software.

6. *Computational science* (or scientific computing) is the __________ of study concerned with constructing mathematical models and quantitative analysis techniques.

7. *A Non-Linear Processor (NLP) algorithm* is __________ to eliminate residual echo.

! More complete definitions may be written by adding examples or extra information:

Virtual reality (VR) is a term that applies to computer-simulated environments that can simulate physical presence in places in the real world, as well as in imaginary worlds.

14. Complete and extend the following definitions.

1. Descriptive geometry is_________ which allows _________by using _________.

2. VRML is_________ for representing _________, designed _________.

3. Computational science (or scientific computing) is _________concerned with _________ and using computers _________.

4. A complex system is _________ composed of _________that as a whole exhibit _________.

5. Computer-aided manufacturing (CAM) is _________ to control _________.

6. CAD is _________ to design _________, such as _________.

7. AL is _________ through the use of _________.

D. Discussion

! Many essay titles require the writer to examine both sides of a case and to conclude by coming down in favour of one side. These may be called discussion, for and against or argument essays. For example: **Computer simulation– pos & cons? – Discuss.**

Discussion vocabulary:

+	-
benefit	drawback
advantage	disadvantage
a positive aspect	a negative feature
pro (informal)	con (informal)
plus (informal)	minus (informal)
one minor benefit of computer simulation … is	a serious drawback to computer simulation is...

15. Discuss the advantages and disadvantages of simulation: *Simulation Pros and Cons*

! Presenting your case it is better to use impersonal phrases rather than *I think*:

It is widely believed that …

Most people consider that ...
It is generally agreed that ...
It is probable/possible that ...
This evidence suggests that ...

! However, if you want to defend a minority point of view, you can use the following:

It can be argued that....
It has been suggested that ....
Some people believe that ....

It is important to show that you are aware of counter-arguments, which can be presented first.

16. Study the example and write similar sentences about simulation using ideas above from (15).

Counter-argument	Your position
Although it has been suggested that ***any incorrect key stroke alters the results of the simulation and give you the wrong results***,	it is generally agreed that ***simulation can give you results that are not experimentally measurable with our current level of technology.***

! Before giving your own opinion, it is necessary to show that you have read the relevant sources and have studied the evidence. Opinions without evidence have little value. The following paragraph discusses a complete computer model of an organism to cure diseases.

Scientists *claim* to have developed the world's first complete computer model of an organism, which can use computer-aided design for better diagnosis and treatment of diseases. A team of Stanford researchers, including an Indian, used data from more than 900 scientific papers to account for every molecular interaction that takes place in the life cycle of Mycoplasma genitalium, the world's smallest free-living bacterium. The model represents a stepping-stone toward the use of computer-aided design in bioengineering and medicine, *according to* the Journal 'Cell'. James M Anderson, director National Institutes of Health Division of Program Coordination, Planning and Strategic Initiatives *points out* that ‘This achievement demonstrates a transforming approach to answering questions about fundamental biological processes’.

It seems that biology over the past two decades has been marked by the rise of high-throughout studies producing enormous troves of cellular information. A lack of experimental data is no longer the primary limiting factor for researchers. Instead, it's how to make sense of what they already know. Scientists *feel* that the model will help to demonstrate a number approaches, including detailed investigations of DNA-binding protein dynamics and the identification of new gene functions.

Press Trust of India / Washington July 22, 2012, 13:06

The paragraph has the following structure:

Scientists – paraphrase + quotation
the Journal 'Cell' – paraphrase
James M Anderson– paraphrase + quotation
Writer's comment on scientists' + opinion

17. Write an essay (about 100 words) on a subject from your own discipline, making use of phrases from exercises 1, 2 above.

E. Examples

! When writing essays/articles it is often better to support statements by giving examples. Phrases for introducing examples include: *for instance/for example/ such as*/e.g. / *particularly/especially /A case in point* (for single examples).

18. Use suitable example phrases to complete the following sentences.

1. ______, when simulating a single server queue, the following variables may be identified and characterized accordingly.

2. ______, a hydraulic system can be used as an analogue of electrical, traffic and economic systems.

3. Symbolic models represent the properties of the real-life system through the means of symbols, ______ mathematical equations and computer programs.

4. Deterministic models are models which do not contain the element of probability. ______ are: linear programming, non-linear programming and dynamic programming.

5. _______is Newtonian physics, which is highly useful except for the very small, the very fast, and the very massive phenomena of the universe.

6. Computer modeling, _______in maintaining budgets and schedules on today's complex facilities, is very important.

7. His principal efforts entail physics based *computer modeling*, ***particularly*** in the areas of atmospheric radiative scattering in application to remote aerosol properties retrieval… .

F. Generalisations

! In written work, generalisations are very useful because they can be used to present complex ideas or data in a simple form. There are two ways of making a generalisation:

a) Using the plural: Lambert's model for body reflection is useful in computer graphics.

b) Using definite article and the singular: The model for body reflection is a useful model. (less common/more formal)

It is better to avoid absolute phrases such as *women are cleverer than men*.

Instead use more cautious phrases such as *women tend to be cleverer than men or most women are more intelligent than men.*

NB! When making generalisations it is easy to over-generalise, using inadequate data.

19. Write generalisations on the following topics.

1. computer/ simulation ____

2. computer animation/generate images___________________________

3. Precautionary principle/states _________________________________

REFERENCES, USEFUL LINKS AND FURTHER READING

1. Academic Writing', by Malashenko, Elena, 2007.

2. Computer Simulation Techniques: The definitive introduction! by Harry Perros Computer Science Department NC State University Raleigh, NC, 2009.pp.4-11.

3. Joslyn, C. and Rocha, L. (2000). Towards semiotic agent-based models of socio-technical organizations, Proc. AI, Simulation and Planning in High Autonomy Systems (AIS 2000) Conference, Tucson, Arizona, pp. 70-79.

4. http://www.csd.cs.cmu.edu/research/areas/scicomp/

5. http://www.sciencedaily.com/releases/2012/08/120812223917.htm

http://www.business-standard.com/generalnews/news/nowcomplete-computer-modelan-organism-to-cure-diseases/35290/

UNIT 3

PROGRAMMING LANGUAGES & PARADIGMS

VOCABULARY

1. Match the words with their definitions:

1) epiphany (n.)	[ɪ'pɪfənɪ]	a) existing, happening, or done at the same time
2) constraint (n.)	[kənstreɪnt]	b) having a tendency to be affected by it or to do it
3) encumber (v.)	[ɪn'kʌmbə]	c) a sudden realization of great truth
4) concurrent (adj.)	[kən'kʌr(ə)nt]	d) a limitation or restriction
5) prone (adj.)	[proʊn]	e) restrict or impede (someone or something) in such a way that free action or movement is difficult
6) compile	[kəm'paɪl]	f) include or contain (something) as a constituent part
7) linker	['liŋkər]	g) convert (a program) into a machine-code or lower-level form in which the program can be executed
1) embody	[ɪm'bɔdɪ], [em'bɔdɪ]	h) a program used with a compiler or assembler to provide links to the libraries needed for an executable program

Before you read

2. Discuss with your partner the following questions.

- **What do you know about programming languages and paradigms?**
- **Is there any difference? Which one if any?**
- **What are the reasons for using programming languages and paradigms?**

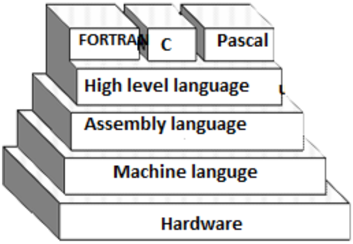
3. Skim the text to check your ideas.

READING

WHAT IS WHAT?

In this article, we will discuss programming languages and paradigms so that you have a complete understanding. Let us first inspect if there any difference is.

The difference between programming paradigms and programming languages is that **programming language** is an artificial language that has vocabulary and sets of grammatical rules to instruct a computer to perform specific tasks. **Programing paradigm** is a particular way (i.e., a 'school of thought') of looking at a programming problem.



The term programming language usually refers to high-level languages, such as BASIC, C, C++, COBOL, FORTRAN, Ada, and Pascal. Each language has a unique set of keywords (words that it understands) and a special syntax for organizing program instructions. High-level programming languages, while simple compared to human languages, are more complex than the languages the computer actually understands, called machine languages. Each different type of CPU has its own unique machine language. **Assembly languages** are lying between machine languages and high-level languages. Assembly languages are similar to machine languages, but they are much easier to program in because they allow a programmer to substitute names for numbers. Machine languages consist of numbers only. Lying above high-level languages are languages called fourth-generation languages (usually abbreviated *4GL*). 4GLs are far removed from machine languages and represent the class of computer languages closest to human languages. Regardless of what language you use, you eventually need to convert your program into machine language so that the computer can understand it. There are two ways to do this: *compile* the program and interpret the program. A program that executes instructions **is** written in a high-level language. There are two ways to run programs written in a high-level language. The most common is to compile the program. To transform a program written in a high-level programming language from source code into object code. Programmers write programs in a form called source code. Source code must go through several steps before it becomes an executable program. The first step is to pass the source code through a compiler, which translates the high-level language instructions into object code. The final step in producing an executable program -- after the compiler has produced object code -- is to pass the object code through a *linker*. The linker combines modules and gives real values to all symbolic addresses, thereby producing machine code.

The other method is to pass the program through an interpreter. An interpreter translates high-level instructions into an intermediate form, which it then executes. In contrast, a compiler translates high-level instructions directly into machine language. Compiled programs generally run faster than interpreted programs. The advantage of an interpreter, however, is that it does not need to go through the compilation stage during which machine instructions are generated. This process can be time-consuming if the program is long. The interpreter, on the other hand, can immediately execute high-level programs. For this reason, interpreters are sometimes used during the development of a program, when a programmer wants to add small sections at a time and test them quickly. In addition, interpreters are often used in education because they allow students to program interactively. Both interpreters and compilers are available for most high-level languages. However, BASIC and LISP are especially designed to be executed by an interpreter. In addition, page description languages, such as PostScript, use an interpreter. Every PostScript printer, for example, has a built-in interpreter that executes PostScript instructions. The question of which language is best is one that consumes a lot of time and energy among computer professionals. Every language has its strengths and weaknesses. For example, FORTRAN is a particularly good language for processing numerical data, but it does not lend itself very well to organizing large programs. Pascal is very good for writing well-structured and readable programs, but it is not as flexible as the C programming language. C++ *embodies* powerful object-oriented features, but it is complex and difficult to learn. The choice of which language to use depends on the type of computer the program is to run on, what sort of program it is, and the expertise of the programmer. Computer programmers have evolved from the early days of the bit processing first generation languages into sophisticated logical designers of complex software applications. Programming is a rich discipline and practical programming languages are usually quite complicated. Fortunately, the important ideas of programming languages are simple (See Appendix IV).

Usually, the word "paradigm" is used to describe a thought pattern or methodology that exists during a certain period of time. When scientists refer to a scientific paradigm, they are talking about the prevailing system of ideas that was dominant in a scientific field at a point in time. When a person or field has a **paradigm shift**, it means that they are no longer using the old methods of thought and approach, but have decided on a new approach, often reached through an *epiphany*.

Programming paradigm is a framework that defines how the user conceptualized and interprets complex problems. It is also is a fundamental style or the logical approach to programming a computer based on a mathematical theory or a coherent set of principles used in software engineering to implement a programming language. There are currently 27 paradigms (see the chart above) exist in the world. Most of them are of similar concepts extending from the 4 main programming paradigms.

Programming languages should support many paradigms. Let us name 4 main programming paradigms: **the imperative paradigm**, **the functional paradigm**, **the logical paradigm**, **the object-oriented paradigm**. Other possible programming paradigms are: the visual paradigm, one of the **parallel/ concurrent paradigms** and the *constraint* based paradigm. The paradigms are not exclusive, but reflect the different emphasis of language designers. Most practical languages embody features of more than one paradigm.

Each paradigm supports a set of concepts that makes it the best for a certain kind of problem. For example, object-oriented programming is best for problems with a large number of related data abstractions organized in a hierarchy. Logic programming is best for transforming or navigating complex symbolic structures according to logical rules. Discrete synchronous programming is best for reactive problems, i.e., problems that consist of reactions to sequences of external events. Programming paradigms are unique to each language within the computer programming domain, and many programming languages utilize multiple paradigms. The term paradigm is best described as a "pattern or model." Therefore, a programming paradigm can be defined as a pattern or model used within a software programming language to create software applications. Languages that support these three paradigms are given in a classification table below.

Imperative/ Algorithmic	Declarative		Object-Oriented
	Functional Programming	Logic Programming	
Algol Cobol PL/1 Ada C Modula-3 Esterel	Lisp Haskell ML Miranda APL	Prolog	Smalltalk Simula C++ Java

Popular mainstream languages such as Java or C++ support just one or two separate paradigms. This is unfortunate, since different programming problems need different programming concepts to solve them cleanly, and those one or two paradigms often do not contain the right concepts. A language should ideally support many concepts in a well-factored way, so that the programmer can choose the right concepts whenever they are needed without being *encumbered* by the others. This style of programming is sometimes called multiparadigm programming, implying that it is something exotic and out of the ordinary.

Programming languages are extremely logical and follow standard rules of mathematics. Each language has a unique method for applying these rules, especially around the areas of functions, variables, methods, and objects. For example, programs written in C++ or Object Pascal can be purely procedural, or purely object-oriented, or contain elements of both paradigms. Software designers and programmers decide how to use those paradigm elements. In object-oriented programming, programmers can think of a program as a collection of interacting objects, while in functional programming a program can be thought of as a sequence of stateless function evaluations. When programming computers or systems with many processors, process-oriented programming allows programmers to think about applications as sets of *concurrent* processes acting upon logically shared data structures. Just as different groups in software engineering advocate different methodologies, different programming languages advocate different programming paradigms. Some languages are designed to support one particular paradigm (Smalltalk supports object-oriented programming, Haskell supports functional programming), while other programming languages support multiple paradigms (such as Object Pascal, C++, C#, Visual Basic, Common Lisp, Scheme, Perl, Python, Ruby, Oz and F Sharp).

It is helpful to understand the history of the programming language and software in general to better grasp the concept of the programming paradigm. In the early days of software development, software engineering was completed by creating binary code or machine code, represented by 1s and 0s. These binary manipulations caused programs to react in a specified manner. This early computer programming is commonly referred to as the "low-level" programming paradigm. This was a tedious and error prone method for creating programs. Programming languages quickly evolved into the "procedural" paradigm or third generation languages including COBOL, Fortran, and BASIC. These procedural programming languages define programs in a step-by-step approach.

The next evolution of programming languages was to create a more logical approach to software development, the "object oriented" programming paradigm. This approach is used by the programming languages of Java™, Smalltalk, and Eiffel. This paradigm attempts to abstract modules of a program into reusable objects.

In addition to these programming paradigms, there is also the "declarative" paradigm and the "functional" paradigm. While some programming languages strictly enforce the use of a single paradigm, many support multiple paradigms. Some examples of these types include C++, C#, and Visual Basic®.

Each paradigm has unique requirements on the usage and abstractions of processes within the programming language. Nevertheless, Peter Van Roy says that understanding the right concepts can help improve programming style even in languages that do not directly support them, just as object-oriented programming is possible in C with the right programmer attitude.

By allowing developers flexibility within programming languages, a programming paradigm can be utilized that best meets the business problem to be solved. As the art of computer programming has evolved, so too has the creation of the programming paradigm. By creating a framework of a pattern or model for system development, programmers can create computer programs to be the most efficiency within the selected paradigm.

LANGUAGE DEVELOPMENT

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

1) constraint programming	a)
2) discrete synchronous programming	b)
3) software applications	c)
4) concurrent processes	d)
5) step-by-step approach	e)
6) external event	f)
7) assembly language	g)
8) source code	h)

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

1)программа, работающая с абстрактными типами данных	a)
2)реагирующие проблемы (изменяющие своё поведение в ответ на конкретные ситуации)	b)
3)оценочная функция без запоминания состояния о сетевых протоколах	c)
4)низкоуровневое программирование	d)
5)уязвимый для ошибок	e)
6)преобразовывать исходные тексты программы в объектные модули	f)
7)объектный код, объектная программа	g)
8)стадия компиляции	h)

KEY CONCEPTS

Concurrent programming

It improves performance. Multiprogramming systems attempt to utilize resources that would otherwise be wasted, by running two or more jobs concurrently. Multiaccess systems extend this principle, allowing many jobs to be run, each on behalf of a user at an interactive terminal. Concurrency can be classified into: Apparent concurrency: single processor (interleaved execution of concurrent tasks); Real concurrency: multiprocessor environment. There are some issues: How to synchronize the interactions among concurrently executing processes to maintain the internal data integrity. Another problem is to schedule the racing processes for a limited set of shared resources.

Imperative paradigm

It is based on commands that update variables in storage. The Latin word *imperare* means "to command". The language provides statements, such as *assignment statements*, which explicitly change the *state* of the memory of the computer.

This model closely matches the actual executions of computer and usually has high execution efficiency. Many people also find the imperative paradigm to be a more natural way of expressing themselves.

Functional programming paradigms

In this paradigm we express computations as the evaluation of mathematical functions. These paradigms treat values as single entities. Unlike variables, values are never modified. Instead, values are transformed into new values. • Computations of functional languages are performed largely through applying functions to values, i.e., (+ 4 5).

Logic programming paradigms

In this paradigm we express computation in exclusively in terms of *mathematical logic.*

While the functional paradigm emphasizes the idea of a mathematical function, the logic paradigm focuses on predicate logic, in which the basic concept is a *relation*. Logic languages are useful for expressing problems where it is not obvious what the functions should be.

paradigm shift

A fundamental change in approach or underlying assumptions Origin: 1970s: term used in the writings of Thomas S. Kuhn (1922-96), philosopher of science**.**

Programming culture

The totality of programming behavior, which often is tightly related to a family of programming languages**.** The sum of a main paradigm, programming styles, and certain programming techniques.

Programming languages

It is a notational system for describing tasks/computations in a machine and human readable form. Most computer languages are designed to facilitate certain operations and not others: numerical computation, or text manipulation, or I/O. More broadly, a computer language typically embodies a particular *programming paradigm*.

Programming paradigm

A pattern that serves as a *school of thoughts* for programming of computers

Programming style

The way we express ourselves in a computer program. Related to elegance or lack of elegance.

Programming technique

Related to an algorithmic idea for solving a particular class of problems

The Object-Oriented Paradigm

The Object Oriented paradigm focuses on the *objects* that a program is representing, and on allowing them to exhibit "behavior". Unlike imperative paradigm, where data are passive and procedures are active, in the O-O paradigm data is combined with procedures to give *objects*, which are thereby rendered active. Alan Kay characterized the fundamental of OOP as follows: Everything is modeled as object. Computation is performed by message passing: objects communicate with one another via message passing. Every object is an instance of a class where a class represents a grouping of similar objects. Inheritance: defines the relationships between classes.

7. Translate the following sentences into Russian.

1. However, it does not mean that constraint programming is restricted to CLP.

2. Data abstraction is a programming (and design) technique that relies on the separation of interface and implementation.

3. In computer science, a low-level programming language is a programming language that provides little or no abstraction from a computer's instruction set architecture.

4. Early systems were frequently error-prone and difficult to modify because they made widespread use of global data.

5. Functional programming a program can be thought of as a combination of stateless function evaluations.

8. Translate the following sentences into English.

1. Программное приложение это определенный подкласс компьютерного программного обеспечения, который применяет возможности компьютера непосредственно к выполнению желаемой задачи пользователя.

2. Поэтапный метод используется многими хакерами.

3. Параллельные процессы, исполняемые в операционной системе могут быть как независимыми так и взаимодействующими процессами.

4. Дискретное синхронное программирование прекрасно подходит для решения реагирующих проблем, т.е. проблем, которые изменяют своё поведение в ответ на внешние события.

SPEAKING SECTION

9. Answer the following questions.

1. What do we mean by "programming paradigms"?

2. Discuss how the following object-oriented concepts help a programmer design and implement an application. Illustrate your answer with appropriate examples:

a) objects and classes
b) encapsulation
c) specialisation and inheritance
d) polymorphism

e) aggregation.

3. When one first encounters a new programming language, the first question is usually: What can this language 'do'?"(Ben-Ari 1996). Choose two different programming languages, which are based on different paradigms (such as data-oriented, imperative, object-oriented, or scripting) and: a) compare and contrast what each language can do; b) discuss why you should use each one, highlighting what type of applications they are most suitable for.

4. For many real-world applications a programming language that supports concurrency is desirable. In particular, solutions to the problems of process synchronisation and inter-process communication are required. Explain how these problems arise, and describe the range of solutions that are available. In your answer discuss the relative strengths and weaknesses of each solution, and where possible, illustrate your answer with the aid of real concurrent language examples.

5. There are numerous books and papers that address why software projects fail. Common software issues include: inability to cope with changing requirements; hard to maintain software; modules that do not fit together; poor software quality; unacceptable performance and untrustworthy build processes. Choose ONE programming paradigm (such as data-oriented, imperative, object-oriented, event driven or scripting), and discuss what characteristics it has that would either help, or hinder, software project development.

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

BACKGROUND TO WRITING

ELEMENTS OF WRITING (2)

A. Numbers

! Discussing statistical data is a necessary part of much academic writing:

Of the total world population of 6,767,805,208, the latest data of *Internet Users* is numbered at 1,733,993,741.

The four-day event has 373 entries from 15 states -- the highest number of competitors since 2002.

Figures and numbers are both used to talk about statistical data in a general sense. The figures in the report need to be read critically. Digits are individual numbers. Both fractions (1⁄2) and decimals (0.975) may be used.

7,673 – a four *digit* number

$ 234,400 – a six *figure* salary (a number)

Figure (*Fig*) 2 – Methodologies to estimate data for 1970 to 1984 period and 1985 to present emissions differ (a diagram)

NB! No final -s on hundred/thousand/million:

In 1992 104, 425 thousand people voted for Clinton. but: *Thousands of processors form a large cluster.*

! When presenting data, the writer must attempt to be accurate without confusing the reader with too much detail. In some cases, where the actual number is unimportant, words or phrases may replace numbers to simplify the text:

Deciding between the *47 of programs* that transfer iPods to computers can be maddening.

Deciding between the *dozens of programs* that transfer iPods to computers can be maddening.

! The following words or phrases can be used to describe quantity.

Few	less than expected
Several	3–4
Various	3–6
Dozens of	30–60
Scores of	50–100

10. Rewrite the following sentences using one of the words or phrases above.

1. Only Four People Showed Up to Protest Apple at Grand Central

2. It is his achievement to have *created* five or six useful patterns.

3. The exhibition will consist of some forty to forty-five works from the Met's collection

4. Scholarships are awarded for *three, or four* years.

5. The Kearney Park and Recreation Department, with the assistance of nearly 100 volunteers, put on the Games.

! Percentages are commonly used for expressing rates of change:

Approximately 48% of U.S. adults are using smartphones and the penetration exceeds 60% in some markets this year.

Study the following expressions, which are also used to simplify statistics.

One in three	a third/a quarter
twice/three times as many	the majority/the minority
a tenfold increase	fifty per cent, a percentage
to double/to halve	on average/the average number
the most/the least	a small/large proportion

11. Rewrite each sentence in a simpler way, using one of the expressions above.

1) More than 400,000 people use computers for a social purpose such as social networking, or communicating with or emailing family and friends.
2) As of December 2008, there were 75,750,000 mobile phones in use in the *UK*.
3) Belarusian is an Eastern Slavonic language with about 7.5 million speakers in Belarus.
4) The LinkedIn Web site was launched in 2003 and is now the largest professional networking site in the world with more than 65 million members, representing 200 countries and executives from every Fortune 500 company.
5) 217 million people worldwide play online games, according to new figures released by comScore. (28%)
6) A recent study shows that 40 percent of computer users at the D.C. Public Library use the computers to research and apply for jobs and 20 percent of them report that they found jobs as a result.
7) Over the past four years, as part of its ongoing transformation, the library increased the number of computers available to the public from 100 in 2006 to 600 this year.

B. References and Quotations

! A reference is an acknowledgement that you are making use of another writer's ideas or data in your writing:

Robert S. Bauer, an executive director at Willow Garage, *pointed out* that *computers* were once seen as exotic machines.

There are three main reasons for giving references:

a) To avoid plagiarism.

b) To give more authority to your writing.

c) To show you are familiar with other research on the topic.

c) To help reader to find the original source by using the reference section.

Colles W. M., Hardy H. Playright and Copyright in all Countries. — L., 1906.

! In order to give references accurately it is important to follow the following procedure:

a) Keep a careful record of the details of your sources.

b) Study journals and departmental guidelines to find out which system of referencing is used in your subject area.

c) Use:

- summary of a writer's ideas:

Robert S. Bauer, an executive director at Willow Garage, pointed out that computers were once seen as exotic machines. In the early 1970s, he said, Xerox Parc developed a series of sophisticated computers that cost several hundred thousand dollars. However, these innovative machines paved the way for today's personal computers.

-quotation of a writer's words:

Dr. Bauer predicted that the first wave of robots would most likely become "the body for people with physical disabilities." [Nick Bilton, Disruptions: Dining With Robots in Silicon Valley, 2012.]

-a mixture of summary and quotation:

As Mr. Cousins noted, these gadgets will be given more functional bodies, including arms, so they can interact in a physical space. He considered that "Today's telepresence robots let you be somewhere" [Bilton N., 2012: pp.44-56].

12. Write a summary of the author's ideas, including a suitable reference.

1. 'The revolution taking place in information and communication technologies have been the central and driving force for the globalization process' [Ajayi, G. O. (2000). *Challenge to Nigeria of Globalization and the Information Age.* Keynote address at the Workshop on National Information Communication Infrastructure (NICI) Policy Plans and Strategies for Implementation, Maitama, Abuja].

! Referring verbs use both the present and the past tenses. If you feel that the ideas or data are still valid it is better to use the present tense for recent sources:

In addition, as Trostinikov (1970) has pointed out, rapid expansion of a mass of diversified information is occurring, which has received the name "information explosion".

Suggests that If the source is older and the ideas perhaps out-of-date use the past tense:

According to Daniel (2000) Nancy Schiller was one of the first writers to use the expression "virtual library" which she defined in 1992, simply as "libraries in which computer and telecommunications technologies make access to a wide range of information resources possible."

! There are three main systems of reference in use in academic writing:

Capron (2000) revealed … (date of publication in brackets when referring verb is used)

This surface model was used to describe incoherent directional component of surface reflection from rough surfaces. [K. Torrance and E. Sparrow, 1967] (authors and date in brackets after summary).

NB! For quotations page numbers should also be given after the date.

'Addressing a different cause for non-Lambertian reflectance from the one discussed here, we used linear transport theory to analyze subsurface scattering from a multi-layered surface'[P. Hanrahan andW. Krueger, 1993, pp.165-174].

13. Insert numbers in brackets in the text for each source, and at the end of the chapter or article list the references in number order:

This function was designed to obey Helmholtz's reciprocity principle [2] but is not based on any theoretical foundation. The studies cited above were attempts to design reflectance models based on measured reflectance data. In contrast, Smith [1] and Buhl et al.

1. B. G. Smith. Lunar surface roughness: Shadowing and thermal emission. Journal of Geophysical Research, vol.72(16):, August 1967 ,pp.4059–4067.

2. P. Beckmann and A. Spizzichino. *The Scattering of Electromagnetic-Waves from Rough Surfaces*. Pergamon, New York, 1963, pp.436-450.

14. Use footnotes. The references are listed at the bottom of the page.

This phenomenon is referred to as backscattering[3].

NB! A full reference section is required at the end of the article or book.

C. Style

! Modern academic writing has a semi/formal, impersonal and objective style.

The focus is on presenting information as clearly and accurately as possible. Be as precise as possible when dealing with facts or figures. If it is necessary to estimate numbers use *approximately* rather than *about.*

Following are some of the small but specific mistakes in style that are made (mainly unconsciously) in formal written work.

Avoid using:

- personal language
- language that is emotional
- words that express your opinion too strongly
- unnecessary words
- brackets and dashes to add information

[3] B. W. Hapke and Huge van Horn. Journal of Geophysical Research, 68(15), August 1963, pp.4545–4570.

- *like* (use *such as* or *for instance*), *thing* (use *factor*, *issue* or *topic)*, *lots of* (use *a significant/considerable number*), *little/big* (use *small/large*), *get* phrases (use *improve* and *deteriorate*)
- *etc*. or *and so on* when writing lists,. Insert *and* before the last item.
- phrases such as *about a hundred* or *hundreds of years ago*.

Do not use:

- colloquial language or slang
- shortened forms of words and phrases incorrectly
- dot/bullet point lists unless you are instructed to
- shift verb tense unnecessarily
- exclamation marks (!!!) in your essay
- question forms. Use statements instead.
- questions and commands
- misuse font and font styles (mainly italics & underlining)

15. In the following, first underline the examples of poor style and then re-write them in a more suitable way:

1) Lots of people build computers so you need to look at component shipments.
2) Yes, we know this is a sadly long list of rules, but please read it.
3) You can't always trust those download links that you find.
4) The second thing is, anyone who uses the Internet today is agreeing to one form of TOS or another, it's the way everything is going.
5) Alternatively, perhaps the varying meanings of the word "program" in everyday language ("TV program", "study program" *etc.*) confuse things.
6) Lots of studies looked at adults rather than kids.
7) I don't think anyone understood how important the Steam gaming service would become when it was released in 2003.
8) I've heard mixed things about this game, but the mysterious Ada Wong character looks like the most exciting thing about it to me!
9) Why people insist computers can't think?

D. Synonyms

! When writing it is necessary to find synonyms in order to provide variety and interest for the reader: Synonyms are not always exactly the same in meaning, but it is important not to change the register. There are some common academic synonyms you need to remember:

Nouns		Verbs	
goal	target	reduce	decrease
study	research	achieve	reach
results	findings	alter	change
area	field	evaluate	examine
authority	source	claim	suggest
benefit	advantage	assist	help
category	type	attach	join
component	part	challenge	question
concept	idea	clarify	explain
behaviour	conduct	collapse	fall down
controversy	argument	concentrate	focus
feeling	emotion	confine	limit
beliefs	ethics	show	demonstrate
expansion	increase	eliminate	remove
interpretation	explanation	found	Establish
issue	topic	develop	evolve
method	system	maintain	insist
option	possibility	predict	forecast
statistics	figures	prohibit	ban
framework	structure	retain	keep
trend	tendency	strengthen	reinforce
		accelerate	speed up

16. Replace all the words or phrases in italic with suitable synonyms.

1) Symfony *Components* implement common features needed to develop websites.

2) FedStats provides easy access to *statistics* and information produced by more than 100 U.S. Federal Government agencies.

3) Of course, the spree killer was a CounterStrike gamer and a lot of politicans are now calling for a total *prohibition* of 'killer' *computer* games.

4) These games aren't just for fun. They're designed to teach kids how *to concentrate*.

5) Some *components* may *retain high voltages* even after the printer is turned off.

E. Tables and Figures

! At times, it may be permissible and appropriate to insert tables, figures and other graphics in your essay. These graphics may have been copied, adapted from sources of information or may be from your own research. They need to be relevant, correctly labelled and referenced—unless they are entirely your own work. It is important that tables and figures are used purposefully (i.e. with good reason) and referenced correctly.

NB! **DO NOT:**

- restructure data from an information source into another format (e.g. a graph, a flowchart) without referencing the author of your information. You may structure the graph, but the author still 'owns' the research!
- just 'plonk' a table or figure into your writing. You need to refer to its existence and relevance to your argument in the preceding text.
- give extensive descriptions in your writing of the contents of a table or diagram. The information in a table or diagram tells its own story—it's your job to point out its significance to your argument.

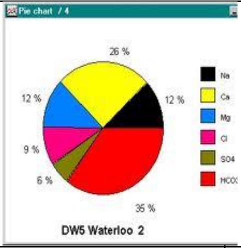
17. Below are illustrations of some of the main types of visuals used in academic texts. Match the uses (a-f) to the types (1-6) and the examples (A-F) in the box below.

Uses

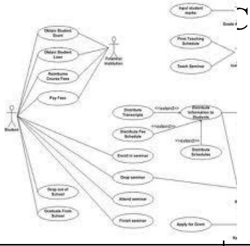
a) location b) comparison c) proportion d) function e) changes in time f) statistical display

Types	Uses	Example
1. diagram (scheme)		
2. table		
3. map		
4. pie chart		
5. bar chart		
6. line graph		

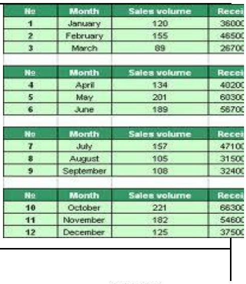
A.



B.



C.



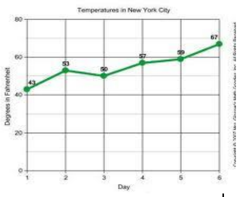
№	Month	Sales volume	Recei
1	January	120	3600
2	February	155	4650
3	March	89	2670

№	Month	Sales volume	Recei
4	April	134	4020
5	May	201	6030
6	June	189	5670

№	Month	Sales volume	Recei
7	July	157	4710
8	August	105	3150
9	September	108	3240

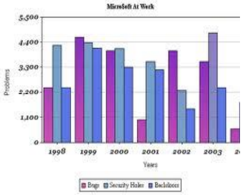
№	Month	Sales volume	Recei
10	October	221	6630
11	November	182	5460
12	December	125	3750

D.



E.

F.

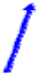


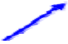
! Although visuals do largely speak for themselves, it is usual to help the reader interpret them by briefly commenting on their main features using the language of change.

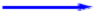
Verb	Verb	Adjective + Noun
grow	fall (to)/ decline (to)	a slight drop
raise/rise (to)	drop (to)	a gradual fall
increase (to)	decrease (to)/reduce (to)	a sharp decrease
climb (to)	collapse	a dramatic growth
step up	cut	a huge boom
expand	go down (to)/fall (to)	a steep increase
improve	break	a substantial climb
shoot up/soar	push down	a considerable change
peak (at)	bottom out	a significant reduction
boom	dip (to)	a marked change
Adverbs		a moderate reduction
substantially	slightly	a minimal growth
considerably	gradually	**Verb**
significantly	steadily	maintain the same level
markedly	sharply	stay at
moderately	dramatically	keep
minimally	hugely	reach a peak
swiftly	rapidly	remain stable
quickly	slowly	be constant
suddenly		level off/out

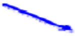
18. Place the correct letter in the right box.

	slump		rise		recover		plunge		pick up		hold steady
	drop		soar		climb		fall		rally		bounce back
	crash		slide		stabilize		escalate		decline		take off
	rocket		dip		fluctuate		plummet				flatten out

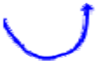

A

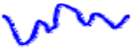

B


C


D

E

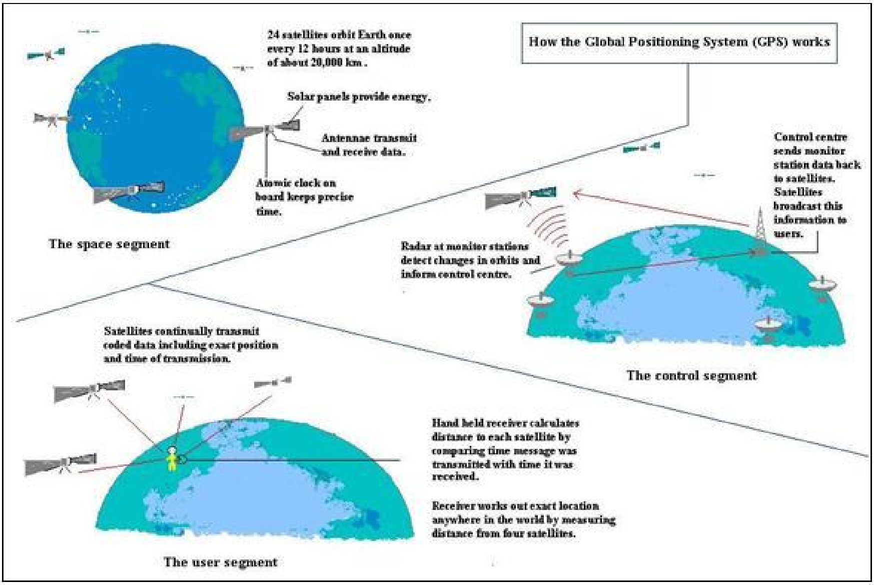

F


G

! To introduce visuals and name the parts of a diagram you can use the following phrases:

Introducing visuals	Naming the parts of diagrams
I'd like you to look at this graph...	The vertical axis represents total annual …
Let me show you this pie chart...	The horizontal axis shows our …
Let's have a look at this model...	The curve,
Let's turn to this map...	The solid line,
To illustrate my point let's look at some diagrams...	The dotted line,
As you can see from these figures...	The broken line,
If you look at these photographs you'll see...	The shaded area,
If you look at this bar chart you'll notice...	The unshaded section,
If you look at this histogram you'll appreciate...	The dotted column,
If you look at this flow chart you'll understand ..	The coloured segment,
If you look at this matrix...	The red bar…

19. The diagram shows how the Global Positioning System (GPS) works in order to help people find their location anywhere on Earth. Write a report for a university lecturer describing the information shown below. You should write at least 150 words.


How the Global Positioning System (GPS) works
24 satellites orbit Earth once every 12 hours at an altitude of about 20,000 km .
Solar panels provide energy.
Antennae transmit and receive data.
Atomic clock on board keeps precise time.
The space segment
Control centre sends monitor station data back to satellites. Satellites broadcast this information to users.
Radar at monitor stations detect changes in orbits and inform control centre.
The control segment
Satellites continually transmit coded data including exact position and time of transmission.
Hand held receiver calculates distance to each satellite by comparing time message was transmitted with time it was received.
Receiver works out exact location anywhere in the world by measuring distance from four satellites.
The user segment

a) Look at the diagram and answer the following questions.

- How many segments are there in the system?
- What are the main components of each segment?
- What part does each segment play in the whole system?

b) Now write a description following this plan:

1. Describe briefly what a GPS is and what the three segments of the system are.
2. Describe the space segment: what it consists of and what it does.
3. Describe the control segment: what it consists of and what it does.
4. Describe the user segment: what it consists of and what it does.

REFERENCES , USEFUL LINKS AND FURTHER READING

1. Academic Writing’, by Malashenko, Elena, 2007.
2. BCS the Chartered Institute for IT , 2012. /http://www.bcs.org/content/
3. Charts, graphs and diagrams. Business English, 2008.pp.4-9./www.thelanguagemenu.com/

UNIT 4

DATA TYPE IN A PROGRAMMING LANGUAGE

VOCABULARY

1. Match the words with their definitions:

1) constraint (n.)	[kən'streɪnt]	a) the length of time for which a person or animal lives or a thing functions
2) lifespan (n.)	['laɪfspæn]	b) to encase in or as if in a capsule.
3) encapsulation (n.)	[ɪn'kæpsjuleɪʃ(ə)n]	c) a limitation or restriction

Before you read

2. Discuss with your partner the following questions.

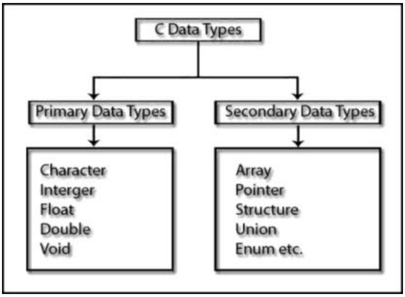
- **What do you know about data types?**
- **Why are they essential?**

3. Skim the text to check your ideas.

READING

IS DATA TYPE ESSENTIAL?

by Guy Lecky-Thompson



A data type in a programming language is a set of data with values having predefined characteristics. Data types are essential to any computer programming language. It is easy for humans to distinguish between different types of data. We can usually tell at a glance whether a number is a percentage, a time, or an amount of money. We do this through special symbols -- %, :, and $ -- that indicate the data's type. Similarly, a computer uses special internal codes to keep track of the different types of data it processes. Most programming languages require the programmer to declare the data type of every data object, and most database systems require the user to specify the type of each data field. Examples of data types are: integer, floating point unit number, character, string, and pointer. The available data types vary from one programming language to another, and from one database application to another, but the following usually exist in one form or another:

Integer: In more common parlance, whole number; a number that has no fractional part.

Floating-point: A number with a decimal point. For example, 3 is an integer, but 3.5 is a floating-point number.

Character (text): Readable text

Usually, a limited number of such data types come built into a language. Without them, it becomes very difficult to maintain information within a computer program. Since the main principle behind computer programming is to take information, process it, and deliver the information in a different form to the user, data types obviously play a large part in determining how this is achieved. With object-oriented programming, a programmer can create new data types to meet application needs. Such an exercise as known as "data abstraction" and the result is a new **class** of data. Such a class can draw upon the "built-in" data types such as number integers and characters. For example, a class could be created that would abstract the characteristics of a purchase order. The purchase order data type would contain the more basic data types of numbers and characters and could also include other object defined by another class. The purchase order data type would have all of the inherent services that a programming language provided to its built-in data types. The language usually specifies the range of values for a given data type, how the values are processed by the computer, and how they are stored. Different languages have different *constraints* upon the data types that they provide. Languages that leave little room for programmers to define their own data types are said to be strongly-typed languages. Weakly typed languages, on the other hand (like many BASIC variants) do not require that a variable's data type is declared before use, but it should always maintain the same data type throughout its *lifespan*. Languages usually allow the possibility to cast (convert) between compatible types.

In most languages, numbers are either integers or floating points. The number 320 is an integer, as is -125. The number 27.1 is a floating point number, as is -34.2.

The larger the range of numbers needing to be represented, the larger the (fixed) data storage requirement will be. Subsequently, and because memory and other storage has traditionally been limited, computer programming languages also provide different sizes of numbers:

- Short: Small integer ranges
- Long: Large integer ranges
- Float: Small floating point ranges

- Double: Large floating point ranges

The names may change (Modula-2 uses REAL, INTEGER and LONGREAL, LONGINT for example), but the general principles remain the same. The exact ranges may also change, but can be calculated if one knows the byte size of the data type - we can only represent the numbers 0 to 255 (or -127 to 128) in a single byte, for example.

Support for complex numbers (with a real and imaginary part) is not always assured, and matrix calculations will usually have to be performed manually (programmatically).

A character value is usually one byte of ASCII, but other character sets (EBCDIC, for example) may change this. It is not usual for a programming language to deal directly with strings (sequences of characters), and usually a scalar variable (array), or pointer to a memory block of a known size will need to be declared.

Where the string type is available, it should usually be an abstract data type (or class) which hides the internals from the programmer (*encapsulation*), and must be manipulated using the exposed functions (methods) for doing so. For example, in C++ we might have a statement 'MyString-GetLength()'.

A pointer is a reference to a piece of memory, whether that be 'raw' memory, or formatted according to another data type, or a piece of executable code, such as a function (procedure, method...).

Pointers can be used to store arrays (like strings of characters), or as a reference to s static single value. Care needs to be taken when using pointers, as they can have some strange properties under certain conditions.

Let us have a look at other data types. This link can either help to find more information about data types http://www.cplusplus.com/doc/tutorial/ other_data_types/.

Defined data types (typedef)

C++ allows the definition of our own types based on other existing data types. We can do this using the keyword typedef, whose format is: typedef existing_type new_type_name; where existing_type is a C++ fundamental or compound type and new_type_name is the name for the new type we are defining.

Unions

Unions allow one same portion of memory to be accessed as different data types, since all of them are in fact the same location in memory. Its declaration and use is similar to the one of structures but its functionality is totally different:

```
union union_name {
  member_type1 member_name1;
  member_type2 member_name2;
  member_type3 member_name3;
  .
  .
} object_names;
```

All the elements of the union declaration occupy the same physical space in memory. Its size is the one of the greatest element of the declaration. One of the uses a union may have is to unite an elementary type with an array or structures of smaller elements.

Anonymous unions

In C++ we have the option to declare anonymous unions. If we declare a union without any name, the union will be anonymous and we will be able to access its members directly by their member names.

Enumerations (enum)

Enumerations create new data types to contain something different that is not limited to the values fundamental data types may take. Its form is the following:

```
enum enumeration_name {
  value1,
  value2,
  value3,
  .
  .
} object_names;
```

Enumerations are type compatible with numeric variables, so their constants are always assigned an integer numerical value internally. If it is not specified, the integer value equivalent to the first possible value is equivalent to 0 and the following ones follow a +1 progression.

LANGUAGE DEVELOPMENT

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

1) internal code	a)
2) inherent services	b)
3) purchase order	c)
4) weakly-typed language	d)

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

1) абстракция данных (определение типа данных вместе с определенными на нем операциями, а также аксиомами, задающими формальные свойства этих операций)	a)

2) встроенный тип данных (тип данных, исходно имеющийся в самом языке программирования, в отличие от типов данных, определённых программистом в исходном тексте программы)	b)
3)язык со строгим контролем типов	c)
4) возможность сокрытия некоторых аспектов представления класса	d)
5) скалярная переменная переменная, принимающая значения скалярного типа	e)

KEY CONCEPTS

C++

The C++ language was designed by Bjarne Stroustrup at AT&T's Bell Labs in Murray Hill, New Jersey, starting in 1979. By that time the C language had become well established as a powerful tool for systems programming (see C (Unit 5). However Stroustrup (and others) believed that C's limited data structures and function mechanism were proving inadequate to express the relationships found in increasingly large software packages involving many objects with complex relationships.

class

A class is a data type that combines both a data structure and methods for manipulating the data. For example, a string class might consist of an array to hold the characters in the string and methods to compare strings, combine strings, or extract portions of a string.

data types

As far as the circuitry of a computer is concerned, there's only one kind of data—a series of bits (binary digits) filling a series of memory locations. How those bits are to be interpreted by the people using the computer is entirely arbitrary. The purpose of data types is to define useful concepts such as integer, floating-point number, or character in terms of how they are stored in computer memory.

Thus, most computer languages have a data type called integer, which represents a whole number that can be stored in 16 bits (two bytes) of memory. When a programmer writes a declaration such as: in the C language, the compiler will create machine instructions that set aside two bytes of memory to hold the contents of the variable Counter. If a later statement says: Counter = Counter + 1; (or its equivalent, Counter++) the program's instructions are set up to fetch two bytes of memory to the processor's accumulator, add 1, and store the result back into the two memory bytes. Similarly, the data type long represents four bytes (32 bits) worth of binary digits, while the data type float stores a floating-point number that can have a whole part and a decimal fraction part (see numeric data). The char (character) type typically uses only a single byte (8 bits), which is enough to hold the basic ASCII character codes up to 255 (see characters and strings). The Bool (Boolean) data type represents a simple true or false (usually 1 or 0) value (see Boolean operators).

encapsulation

In the earliest programming languages, any part of a program could access any other part simply by executing an instruction such as "jump" or "goto." Later, the concept of the subroutine helped impose some order by creating relatively
self-contained routines that could be "called" from the main program. At the time the subroutine is called, it is provided with necessary data in the form of global variables
or (preferably) parameters, which are variable references or values passed explicitly when the subroutine is called. When the subroutine finishes processing, it may return values
by changing global variables or changing the values of variables that were passed as parameters. Encapsulation thus both protects code from uncontrolled modification or access and hides information (details) that programmers who simply want to use functionality don't need to know about. Thus, high-quality classes can be designed by experts and marketed to other developers who can take advantage of their functionality without having to "reinvent the wheel."

characters and strings

While the attention of the first computer designers focused mainly on numeric calculations, it was clear that much of the data that business people and others would want to
manipulate with the new machines would be textual in nature. Billing records, for example, would have to include customer names and addresses, not just balance totals.

The "natural" representation of data in a computer is as a series of two-state (binary) values, interpreted as binary numbers. The solution for representing text (letters of the alphabet, punctuation marks, and other special symbols) is to assign a numeric value to each text symbol. The result is a character code, such as ASCII (American Standard Code for Information Interchange), which is the scheme used most widely today. (Another system, EBCDIC (Extended Binary-Coded Decimal Interchange Code) was used during the heyday of IBM mainframes, but is seldom used today.) The seven-bit ASCII system is compact (using one byte of memory to store each character), and was quite suitable for early microcomputers that required only the basic English alphabet, punctuation, and a few control characters (such as carriage return). In an attempt to use characters to provide simple graphics capabilities, an "extended ASCII" was developed for use on IBM-compatible PCs.

This used eight bits, increasing the number of characters available from 128 to 256. However, the use of bit-mapped graphics in Windows and other operating systems made this version of ASCII unnecessary. Instead, the ANSI (American National Standards Institute) eight-bit character set used the additional character positions to store a variety of special symbols (such as fractions and the copyright symbol) and various accent marks used in European languages.

numeric data

Text characters and strings can be stored rather simply in computer memory, such as by devoting 8 bits (one byte) or 16 bits to each character. The storage of numbers is more complex because there are both different formats and different sizes of numbers recognized by most programming languages.

Integers (whole numbers) have the simplest representation, but there are two important considerations: the total number of bits available and whether one bit is used to hold the sign. Since all numbers are stored as binary digits, an *unsigned* integer has a range from 0 to 2bits where "bits" is the total number of bits available. Thus if there are 16 bits available, the maximum value for an integer is 65535. If negative numbers are to be handled, a signed integer must be used (in most languages such as C, C++, and Java, an integer is signed unless unsigned is specified). Since one bit is used to hold the sign and each bit doubles the maximum size, it follows that a signed integer can have only half the range above or below zero. Thus, a 16-bit signed integer can range from -32,768 to 32,767. One complication is that the available sizes of integers depend on whether the computer system's native data size is 16, 32, or 64 bits. In most cases the native size is 32 bits,

so the declaration "int" in a C program on such a machine implies a signed 32-bit integer that can range from -231 or -2,147,483,647 to 231-1, or 2,147,483,647. However, if one is using large numbers in a program, it is important to check that the chosen type is large enough. The *long* specifier is often used to indicate an integer twice the normal size, or 64 bits in this case.

6. Translate the following sentences into Russian.

1. In existing languages, there is no inherent connection between data and the procedures that operate upon that data.
2. In a programming language, encapsulation is used to refer to one of two related but distinct notions, and sometimes to the combination thereof.
3. Scalar variables hold both strings and numbers, and are remarkable in that strings and numbers are completely interchangable.
4. In a strongly-typed languages, things like Multiple Inheritance are necessary if you want your type system to be complete.
5. Further, it turned out that other companies could reverse-engineer the internal code that ran the system hardware (see bios) without infringing IBM's legal rights.

7. Translate the following sentences into English.

1) Майкрософт обвинили в использовании неопубликованного внутреннего кода Виндоуз для выпуска таких программных продуктов как офисный пакет корпорации Майкрософт как преимущество над конкурентами.

2) Во многих программных языках все исходные типы данных являются встроенными типами данных.

3) «Объединение данных» содержит в себе объекты различных типов и размеров и, по выбору, управляющий код с названием «Объединение».

4) Вообще, инкапсуляция, это механизм, объединяющий данные и обрабатывающий их код как единое целое.

5) Целый тип (чисел), их диапазон и размер приведены в таблице (3.2).

SPEAKING SECTION

8. Answer the following questions.

1) Why do programming languages have data types?
2) How can data types be classified?
3) What kinds of identifiers are used to name types?
4) How are data types implemented in programming languages?
5) What is a strongly-typed programming language?
6) What is the difference between a strongly typed language and a statically typed language?
7) What’s the opposite of "strongly typed" and its meaning?
8) Why do some programming languages have user-defined pointers?
9) Why do some programming languages not have user-defined pointers?

9.

Prepare a presentation on the topic being discussed.

ACCURACY IN WRITING

ACCURACY (1)

A. Abbreviations

! An **abbreviation** is a short way of writing a word or a phrase that could also be written out in full. Abbreviations are an important and expanding feature of contemporary English. They are used for convenience, and familiarity with abbreviations makes both academic reading and writing easier. Abbreviations must be clearly distinguished from contractions. The key difference is that an abbreviation does not normally have a distinctive pronunciation of its own. So, for instance, the abbreviation *oz* is pronounced just like *ounce(s)* and the abbreviation *e.g.* is pronounced just like *for example*. There are a number of Latin abbreviations, which are sometimes used in English texts. Here are the commonest ones with their English equivalents:

ca.	circa	approximately/about (is properly used only in citing a date which is not known exactly, and then usually only if the date is given in parentheses)
cf.	confer	compare
etc.	et cetera	and so forth (calls for special comment. It should **never** be used in careful writing: it is vague and sloppy and, when applied to people, rather offensive)
e.g.	exempli gratia	for example
et al.	et alii	and other people
i.e.	id est	in other words, that is
ibid.	ibidem	in the same place (to refer to source mentioned immediately before)
K	kilo	thousand
op. cit.	opus citatum /opere citato	in the source mentioned previously
p.a.	per annum	yearly
sc.	scilicet	which means
v.	vide	consult/see
viz.	videlicet	namely

Fig. = figure (for labelling charts and graphs)
pp. = pages
re = with reference to

The rule about using these Latin abbreviations is very simple: **DON'T use them.** Their use is only appropriate in special circumstances in which brevity is at a premium, such as in footnotes. It is very poor style to spatter your page with these things, and it could be disastrous to use them without being quite sure what they mean. If you do use one, make sure you punctuate it correctly.

!Three main types can be found:

a) **shortened words** – photo (photograph)

b) **acronyms** (a word composed of the first letters of the words in a phrase, especially when this is used as a name, e.g. laser, AIDS (acquired immune deficiency syndrome), NATO (North Atlantic Treaty Organisation). The more official acronyms are written in capitals as NATO, which is a real body, but others use lower case (yuppie – young upwardly mobile professional, which is a concept.

c) **others** are read as sets of individual letters –. They include names of countries, organizations and companies (*USA* / *BBC* / *IBM*), and also abbreviations that are found only in written English (*NB= nota bene / PTO = please turn over / Rd = road*).

!All academic subjects employ abbreviations to save time.

Examples from computing include:

CE=Computer Engineer(ing)
SMS=Short Messaging Service
3DA=Three Dimension Array
AAUI=Apple Attachment Unit Interface

For more examples visit http://www.abbreviations.com/. It could also be helpful for doing exercises later.

!There are many standard abbreviations found in some types of writing which have a full stop after them to indicate a shortened form. For example, *co.* (*company*) and *Oct.* (*October*). With type (b) and (c) abbreviations there is no standard pattern for using full stops, so both *BBC* and *B.B.C.* are used. There is, however, a trend to use full stops less. The important thing is to employ a consistent style in your work.

!Abbreviations can be confusing.

PC, for example, can mean *Police Constable* (in Britain), *personal computer* and also p*olitically correct*. *CD* may stand for *compact disc* or *corps diplomatique*. It is useful to be aware of these potential confusions.

!Other abbreviations are very subject specific and may be special to one article: *In that case they need explaining: is a model transformation language and toolkit. In the field of Model-Driven Engineering (MDE), ATL (ATL Transformation Language) provides ways to produce a set of target models from a set of source models.*

Summary of abbreviations:

- Do not use an abbreviation that can easily be avoided.
- In an abbreviation, use full stops and capital letters in the conventional way.
- Do not forget to punctuate the rest of the sentence normally.

10. Explain the abbreviations in the following sentences.

1) NB. CVs must be no longer than 3 sides of A4.

2) SMS text messaging is the most widely used data application in the world, with 3.6 billion active users.

3) Windows Vista is an operating system released in several variations by Microsoft for use on personal computers, including home and business desktops, laptops, tablet PCs, and media center PCs.

4) Fig. *9* shows a computer game platform on the w.w.w. for the EU, US and China.

5) The *ASCII* is a character-encoding scheme originally based on the English alphabet.

6) The *BCG* format is a computer representation for labelled transition systems, Kripke structures, and, more generally, state/transition models.

7) C++ is one of the most popular programming languages and is implemented on a wide variety of hardware.

8) Op. cit. is an abbreviation used in an endnote or footnote to refer the reader to a previously cited work.

9) Cf. is used in texts to point the reader to another location in the text.

10) *AAUI* is the 14- or 15-pin port or connection interface on earlier models of Macintosh computers that allowed it to be connected by a short interface cable (or "transceiver") to an Ethernet cable.

B. Adverbs

! Adverbs are used in academic texts in a variety of ways. Among the most important are:

a) to provide more detail, with verbs and adjectives:

All a person will have to do is give ***a reasonably*** *coherent description of the desired results and the required program will be coded by some form of artificial intelligence.*

With the development of what ***eventually*** *became TCP/IP (Transmission Control Protocol/Internet Protocol) Vint Cerf and Bob Kahn essentially became the fathers of the Internet we know today (see TCP /IP).*

b) individually, often at the beginning of sentences, to introduce new points:

Currently, the most widely accepted paradigm treats a program as a collection of objects with

defined capabilities that respond to “messages” asking for services.

Alternatively, the user can type with stylus or fingertips on a “virtual keyboard” displayed on the screen.

NB! These can be similar in function to conjunctions.

! Adverbs linked to verbs and adjectives usually fall into three groups.

Time	when?	*previously* entered; *retrospectively* revised
Degree	how much?	declined *considerably;* increased *substantially; strongly* advanced
Manner	in what way?	*remotely* controlled; *carefully* developed; *neatly* dovetailed

For more information see an Alphabetical list of common adverbs on:
http://www.englishclub.com/vocabulary/adverbs-time.htm
http://www.englishclub.com/vocabulary/adverbs-degree.htm
http://www.englishclub.com/vocabulary/adverbs-manner.htm.

! Adverbs used individually need to be employed with care. DO NOT overuse them.

However, adverbs can be useful for opening paragraphs or linking ideas. The following examples are often followed by a comma.

Time	**Relating ideas**
recently	Clearly
increasingly	Obviously
originally	(not) surprisingly
presently	Alternatively
currently	Similarly
traditionally	(more) importantly
lately	

11. Insert a suitable adverb from the table into the gaps in the sentences.

1) _____, if one wants to set a particular bit to zero, one simply ANDs the byte with a byte that has a zero in that position and ones in the rest of the byte.
2) _____,the unlikely platform of a Sony PlayStation 3 and its powerful new processor has been harnessed.
3) While the data ____ takes up more space, the advantage is that the data remains intact and recoverable if any one drive fail.
4) Real-time systems are _____ important because of the importance of the activities (such as air traffic control and power grids) entrusted to them.
5) _____,Webbased readers or aggregators such as NewsGator Online can allow feeds to be read using any Web browser.
6) _____,each time a CGI request is passed to the URL for a script, the appropriate language interpreter must be loaded and initialized.
7) ______, the most widely accepted paradigm treats a program as a collection of objects with defined capabilities that respond to “messages” asking for services.

8) ______ the concerns of computer science overlap a number of related fields.

9) Programming language support for concurrent programming ______ came through devising new dialects of existing languages.

10)______, information technology plays a part in every phase of this effort—and sometimes even becomes part of the battlefield.

C. Articles

! All countable nouns need an article when used in the singular. The article can be either *a/an* or *the.*

Indefinite article *a/an* is used:

To talk about indefinite things;

After the verbs be and have;

With money (a/one pound), fractions (a/one quarter), weight/measures (a/one inch), whole numbers (a/one thousand), price/weight (€1/a liter), frequency/time (twice a day), distance/fuel (30 miles a gallon), distance/speed (20km an hour) and illnesses (a headache, a cold).

In general, definite article *the* is used with:

a) superlatives (*most interactive*)

b) time periods (*nineteenth century/1960s*)

c) unique things (*government/moon/earth*)

d) specified things (*knowledge of most employees*)

e) regions and rivers (east*/River Amur*)

f) very well-known people and things (*English novelist*)

g) institutions and bodies (*World Health Organisation*)

h) positions (*middle*)

It is *not* used with:

i) names of countries, except for the UK, the USA and a few others

j) abstract nouns (*poverty*)

k) companies/bodies named after people/places (*Sainsbury's*, *Sheffield University*)

12. Explain:

1) Research is *an* important activity in universities.

2) *The* research begun by Dr. Mathews was continued by Professor Brankovic

3) *A* survey was conducted among 200 patients in the clinic.

13. In the following sentences, decide if the words in italic are specific or not. Insert 'the' if specific.

1) ____ *Registry Editor* enables you to edit these values, but won't let you create them.

2) ____*database managers* let you define ____ *data type* for ____ *information* you want to store.
3) ____*record* is ____collection of data about ____ *particular person*, place, or thing.
4) ____*beginning* of ____ *fixed-length field* file contains information that defines ____*file's record structure*.

14. Complete the following text by inserting a/an/the (or nothing) in each gap.

1)____most database information is stored in 2) ____fixed-length fields, so called because 3)____ number of characters-spaces-that can be used for each field is determined when 4) ____ database is created. 5) ____ beginning of 6) ____ fixed-length field file contains 7) ____ information that defines 8) ____ file's record structure each field 's name, data type (usually numeric or alphanumeric), and length . In addition, 9) ____ structure might include information on 10) ____ format of 11) ____ data held in 12) ____ field; for example, 13) ____ field used to record dates might require 14) ____ MM-DD-YY (month-day-year) format. 15) ____ field can also be required to validate 16) ____ information entered into it; for example, if 17) ____ data entered into 18) ____ validating state field is not one of 19) ____ 50 postal service abbreviations, it will be rejected.

20) ____ rest of 21) ____ file is data, laid down in one continuous stream. 22) ____ locations at which specific pieces of data are recorded are determined by 23) ____ lengths allotted to each field.

To find any given record, 24) ____ database software calculates 25) ____ location's offset through 26) ____ simple formula: 27) ____ number of 28) ____ particular record multiplied by 29) ____ total length of each record equals 30) ____ starting point of that record. With 31) ____ starting point calculated, 32) ____ program reads 33) ____ bytes beginning at that point in 34) ____ file.

REFERENCES, USEFUL LINKS AND FURTHER READING

1. 'Academic Writing', by Malashenko, Elena, 2007.

2. Berard, Edward V. "Abstraction, Encapsulation, and Information Hiding." Available online. URL: http://www.itmweb.com/ essay550.htm. Accessed July 23, 2007.

3. Booch, G. Object-Oriented Analysis and Design with Applications. 3rd ed. Upper Saddle River, N.J.: Addison-Wesley, 2007.

4. Muller, Peter. "Introduction to Object-Oriented Programming Using C++." Available online. URL: http://www.gnacademy. org/uugna/text/cc/Tutorial/tutorial.html. Accessed August 14, 2007.

5. Poo, Danny, and Derek Kiong. Object-Oriented Programming and Java. 2nd ed. New York: Springer-Verlag, 2007.

6. “IEEE Standard for Floating Point Arithmetic.” Available online. URL: http://www.psc.edu/general/software/packages/ieee/ieee.html. Accessed August 16, 2007.

7. “Numeric Data Types and Expression Evaluation [in C].” Available online. URL: http://www.psc.edu/general/software/packages/ieee/ieee.html. Accessed August 16, 2007.

8. Sebesta, Robert W. Concepts of Programming Languages. 8th ed. Boston: Addison-Wesley, 2007.

9. “Type Conversion and Conversion Operators in C#.” Available online. http://www.psc.edu/general/software/packages/ieee/ieee.html. Accessed August 16, 2007.

10. “XM L Schema Numeric Data Types.” Available online. URL: http://www.psc.edu/general/software/packages/ieee/ieee.html

11. Lary Trask. Abbreviations. Available online. URL:http://www.informatics.sussex.ac.uk/department/docs/punctuation/node28.html. Accessed January 23, 1998.

12. Harry Henderson. Encyclopedia of Computer science and technology.—Rev. Ed.—2008.

UNIT 5

VISUAL PROGRAMMING ENVIRONMENT

VOCABULARY

1. Match the words with their definitions:

1) raster (n.)	['rastə]	a) a component of a user interface which operates in a particular way
2) handling (n.)	['hændlɪŋ]	b) the time it takes for a specific block of data on a data track to rotate around to the read/write head
3) latency (n.)	['leɪt(ə)n(t)sɪ]	c) a programming structure or process formed by linking a number of separate elements
4) thread (n.)	[θred]	d) a rectangular pattern of parallel scanning lines followed by the electron beam on a television screen or computer monitor
5) widget (n.)	['wɪʤɪt]	e) the length of time a program takes to run
6) run-time	['rʌntaɪm]	f) processing

Before you read

2. Discuss with your partner the following questions.

- **What do you know about visuals?**
- **Are they essential? Why?**

3. Skim the text to check your ideas.

READING

WHAT IS VPE?

The first programmers used pencil and paper to sketch out a series of commands, or punched them directly on cards for input into the machine.However, as more computer resources became available, it was a natural thought that programs could be used to help programmers create other programs. The availability of Teletype or early CRT terminals on timesharing systems by the 1960s encouraged programmers to write simple text editing programs that could be used to create the computer language source code file, which in turn would be fed to the compiler to be turned into an executable program (see terminal and text editor). The assemblers and BASIC language implementations on the first personal computers also included simple editing facilities. More powerful programming editors soon evolved, particularly in academic settings. One of the best known is EMACS, an editor that contains its own LISP-like language that can be used to write macros to automatically generate program elements. With the many other utilities available in the UNIX operating system, programmers could now be said to have a programming environment— a set of tools that can be used to write, compile, run, debug, and analyze programs. More tightly integrated programming environments also appeared. The UCSD "p-system" brought together a program editor, compiler, and other tools for developing Pascal programs.

While this system was somewhat cumbersome, in the mid-1980s Borland International released (and steadily improved) Turbo Pascal. This product offered what became known as an "integrated development interface" or IDE. Using a single system of menus and windows, the programmer could edit, compile, run, and debug programs without leaving the main window. The release of Visual Basic by Microsoft a few years later brought a full graphical user interface (GUI). Visual Basic not only ran in Windows, it also gave programmers the ability to design programs by arranging user interface elements (such as menus and dialog boxes) on the screen and then attaching code and setting properties to control the behavior of each interface object. This approach was soon extended by Microsoft to development environments for C and C++ (and later, Java) while Borland released Delphi, a visual Pascal development system. So what is visual programming environment?

VPE is software, which allows the use of visual expressions (such as graphics, drawings, animation or icons) in the process of programming. These visual expressions may be used as graphical interfaces for textual programming languages. They may be used to form the syntax of new visual programming languages leading to new paradigms such as programming by demonstration or they may be used in graphical presentations of the behaviour or structure of a program. Today visual programming environments are available for most languages. Indeed, many programming environments can host many different languages and target environments. Examples include Microsoft's Visual Studio .NET and the open-source Eclipse, which can be extended to new languages via plug-ins.

In short, VPE is a visual programming environment, which supports general purpose visual programming. Theoretically, it can be used to create any application or system. It is based on both structural (vector graphics, graphs, capable of event handling) and nonstructural graphics (*raster* graphics without event *handling*), therefore graphs in schematics can be used to define inputs, outputs, bodies, groups, wires, and connectors of elements as well as event handling related to them. VPE is very suitable for designing complex large-scale schematics.

VPE is visual development environment, or integrated development environment, for rich visual programming, with the following features:

- Context-free programming environment: no strictly specialized
- Multi-platform: Linux, *BSD, Mac OS X, Solaris, Windows
- Multi-language: **Python, C, C++, C#,** Java, JavaScript, Tcl, Lua
- Multi GUI toolkit: Gtk, Qt, Tk (Tkinter), Fltk, Swing, Forms
- Decentralized and distributed systems
- Real-time systems
- Low-*latency* execution with very fast feedback
- Continuous execution and document modification
- With 100% uptime of application
- Running existing protocol stacks/suits in different *thread*s, or even processes, and communicating with them.
- No need to restart application after document (e.g. SCADA) is modified to accept local or global changes.

In the VPE, visual elements can be graphically designed, programmed in one of supported programming languages (such as Python or C), saved and arranged in libraries. After basic elements are built and stored into the libraries, they can be used for the construction of schematics, which are basically documents. It is possible to save a document for later use. Full object serialization is supported by default, so after closing and reopening of the application, and loading the previously saved document in the application, it is theoretically possible to repeat the same conditions under which the document was previously operating. The initial code is located in the document, but event handlers are located in the elements.

Elements can be programmed in various languages, such as: Python, Tcl, C, etc. Primarily in the application itself, but it is advisable to use the Python programming language because it can directly affect the state and behavior of the program/document and all its parts. For instance, it is possible to change the visual appearance of main application, or even operating system settings.

Whole environment is dynamic, meaning that it can be changed at *runtime*, and adapted to the specific needs. For instance, when designing, it is necessary to see all the options in the VPE, but after deployment, it is need to prohibit the possibility of drawing and programming to avoid errors and unexpected situations, and reduce the chance for error.

VPE is a multi-platform application that runs on all modern operating systems. It is written for use on the Linux operating system, but can also be used in other operating systems.

VPE is primarily designed to create SCADA applications. SCADA applications require high precision in numerical operations, which is one of the reasons why Python programming language is used. In addition to accuracy, SCADA systems also require highly reliability. After designing the general solution using the Python programming language as proof of concept, it is suggested to optimize the solution for the greater execution speed and lower memory usage in the C programming language.

VPE comes with all the necessary standard libraries for most of basic operations, and these are mainly operating system libraries, such as: threading, processing, socket, http, ftp, RPC client/server, etc.

Main reference implementation of VPE is written in Python, Tcl, and C programming languages. GUI toolkits used are Tk (Tkinter) and Gtk (PyGtk), but in parallel the application can use other GUI toolkits, such as: Qt, Wx, Forms, Swing, SWT, etc, and their wrappers for different programming languages.

Different *widgets* are available only in an independent top-level window that corresponds to that GUI toolkit. The reason for this is inability of a GUI toolkit to recognize widgets from other GUI toolkits. It should have in mind that VPE is not a GUI editor, but abstract visual editor with a programmable document and elements.

VPE is next generation of visual programming languages, similar to AutoCAD & LabVIEW, allowing complex programmable graphics. VPE can create context-free applications allowing for example telecommunications and other electronics equipment working together in same schema/document, in other words, different unrelated devices can friendly work together in a complex system.

LANGUAGE DEVELOPMENT

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

1) raster graphics	a)
2) handling	b)
3) context-free	c)
4) distributed system	d)
5) real-time system	e)
6) low-latency execution	f)
7) visual editor	g)

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

1) векторная графика	a)
2) крупная схема	b)
3) программный пакет (прикладных программ) разработчика	c)
4) последовательное упорядочение объекта	d)
5) исходный код	e)
6) упаковщик	f)

KEY CONCEPTS

C

The C programming language was developed in the early 1970s by Dennis Ritchie, who based it on the earlier languages BCPL and B. C was first used on DEC PDP-11 computers running the newly developed UNIX operating system, where the language provided a high-level alternative to the use of PDP Assembly language for development of the many utilities that give UNIX its flexibility. Since the 1980s, C and its descendent, C++, have become the most widely used programming languages. Like the earlier Algol and the somewhat later Pascal, C is a procedural language that reflects the philosophy of programming that was gradually taking shape during the 1970s. In general, C's approach can be described as providing the necessary features for real world computing in a compact and efficient form. The language provides the basic control structures such as if and switch and while, do, and for. The built-in data types provide for integers (int, short, and long), floating-point numbers (float and double), and characters (char). An array of any type can be declared, and a string is implemented as an array of char. Pointers (references to memory locations) are used for a variety of purposes, such as for storing and retrieving data in an array. While the use of pointers can be a bit difficult for beginners to understand, it reflects C's emphasis as a systems programming language that can "get close to the hardware" in manipulating memory.

C++

Consider the example of a simple object: a stack onto which numbers can be "pushed" or from which they can be "popped". In C, a stack would have to be implemented as a struct to hold the stack data and stack pointer, and a group of separately declared functions that could access the stack data structure in order to, for example "push" a number onto the stack or "pop" the top number from it. In such a scheme there is no direct, enforceable relationship between the object's data and functions. This means, among other things, that parts of a program could be dependent on the internal structure of the object, or could directly access and change such internal data. In a large software project with many programmers working on the code, this invites chaos.

C#

Introduced in 2002, C# (pronounced "C sharp") is a programming language similar to C++ and Java but simplified in several respects and tailored for use with Microsoft's latest programming platform. C# is a general-purpose language and is thoroughly objectoriented— all functions must be declared as members of a class or "struct," and even fundamental data types are derived from the System.Object class.Compared with C++, C# is stricter about the use and conversion of data types, not allowing most implicit conversions (such as from an enumeration type to the corresponding integer—see data structures). Unlike C++, C# does not permit multiple inheritance (where a type can be derived from two or more base types), thereby avoiding an added layer of complexity in class relationships in large software projects. (However, a similar effect can be obtained by declaring multiple "interfaces" or specified ways of accessing the same class.) Unlike Java (but like C++), C# includes pointers (and a safer version called "delegates"), enumerations (enum types), structs (treated as lightweight classes), and overloading (multiple definitions for operators). The latest version of the language, C# 3.0 (introduced in 2007), provides additional features for list processing and functional programming.

programming environment

Modern programming environments help the programmer in a number of ways. While the program is being written, the editor can highlight syntax errors as soon as they'remade. Whether arising during editing or after compilation, an error message can be clicked to bring up an explanation, and an extensive online help system can provide information about language keywords, built-in functions, data types, or other matters. The debugger lets the programmer trace the flow of execution or examine the value of variables at various points in the program. Most large programs today actually consists of dozens or even hundreds of separate files, including header files, source code files for different modules, and resources such as icons or graphics. The process of tracking the connections (or dependencies) between all these files, which used

to require a list called a *makefile* can now be handled automatically, and relationships between classes in object-oriented programs can be shown visually. Researchers are working on a variety of imaginative approaches for future programming environments. For example, an interactive graphical display might be used to allow the programmer to in effect walk through and interact with various representations of the program.

Python

Created by Guido van Rossum and first released in 1990, Python is a relatively simple but powerful scripting language. The name comes from the well-known British comedy group Monty Python.

Python is particularly useful for system administrators, webmasters, and other people who have to link various files, data source, or programs to perform their daily tasks. Python dispenses with much of the traditional syntax used in the C family of languages. For example, the following little program converts a Fahrenheit temperature to its

Celsius equivalent: temp = input(“Farenheit temperature:”) print (temp-32.0) *5.0/9.0

Without the semicolons and braces found in C and

related languages, Python looks rather like BASIC. Also, note that the type of input data does not have to be declared. The runtime mechanism will assume it’s numeric from the expression found in the print statement. Python programs thus tend to be shorter and simpler than C, Java, or even Perl programs. The simple syntax and lack of data typing does not mean that Python is not a “serious” language, however. Python contains full facilities for object-oriented programming, for example. Python programs can be written quickly and easily by trying commands out interactively and then converted the script to bytecode, a machine-independent representation that can be run on an interpreter designed for each machine environment. Alternatively, there are translation programs that can convert a Python script to a C source file that can then be compiled for top speed. Perl is still a popular scripting language for UNIX and Web-related applications. Perl contains a powerful built-in regular expression and pattern-matching mechanism, as well as many other built-in functions likely to be useful for practical scripting. Python, on the other hand, is a more generalized and more cleanly structured language that is likely to be suited for a wider variety of applications, and it is more readily extensible to larger and more complex applications.

Tcl

Developed by John Ousterhout in 1988, Tcl (Tool command language) is used for scripting, prototyping, testing interfaces, and embedding in applications (see scripting languages). Tcl has an unusually simple and consistent syntax. A script is simply a series of commands (either built in or user defined) and their arguments (parameters). A command itself can be an argument to another command, creating the equivalent of a function call in other languages. For example, setting the value of a variable uses the set command:

```
set total 0
```

The value of the variable can now be referenced as $total. Control structures are simply commands that run other commands. A *while* loop, for example, consists of a command or expression that performs a comparison, followed by a series of commands to be executed each time it returns "true":

```
while { MoreInFile } {
GetData
DisplayData
}
```

In practice, many of the commands used are utilities from the operating system, usually UNIX or Linux. Tcl also includes a number of useful data structures such as associative arrays, which consist of pairs of data items such as:

```
set abbr (California)     CA
```

Tcl includes a number of extensions that, for example, provide access to popular database formats such as MySQL and can interface with other programming languages such as C++ and Java. The most widely used extension is Tk, which provides a library for creating user interfaces for a variety of operating systems and languages such as Perl, Python, and Ruby. Tcl has been described as a "glue" to connect existing applications. It is relatively easy to write and test a script interactively (often at the command line), and then insert it into the code of an application. When the application runs, the Tcl interpreter runs the script, whose output can then be used by the main application.

6. Translate the following sentences into Russian.

1) Raster data is easy to work with, but the "coarseness" of the grid means that it does not capture much local variation.

2) The first application was developed in the 1950s for handling radioactive materials.

3) Knuth also did important work in areas such as LR (left-to-right, rightmost) parsing, a context-free parsing approach used in many program language interpreters and compilers or detail.

4) The AWT (Abstract Windowing Toolkit) is a set of classes that provide a graphical user interface.

5) A wrapper is used to compress and encrypt software that is being sold over the Internet.

7. Translate the following sentences into English.

1. Начиная с 1970-х операционная система UNIX предусматривала в себе строчный редактор (ed или ex) и экранный редактор (vi).

2. Операционная система реального времени (ОСРВ) это операционная система, нацеленная на обработку запросов приложений системы реального времени.

3. Распределённые вычисления, это область вычислительных машин и систем вычисления, которые изучают распределенные системы.

4. Последовательное упорядочение объекта означает сохранение значения объекта в любой момент времени.

5. В компьютерном программировании это графический элемент пользовательского интерфейса.

SPEAKING SECTION

8. Answer the following questions.

1. Describe the importance of Visual interface design.

2. Name the feature of VPE.

3. What was VPE primarily designed for?

4. Why it is important to study detailed code or functions of windows / visual programming, rather there are many professional tools like Visual Studio etc. available that do the same in mush less time as compared to write a detailed code?

5. What is GUI? Explain it.

6. What are the reports used in graphics application?

9.

Prepare a presentation on the topic being discussed.

ACCURACY IN WRITING

ACCURACY (2)

CONJUNCTIONS

! Conjunctions are words and phrases, which join parts of a sentence together. They are:

Coordinating (such as *or, and, but, yet, for, nor, so*). When a coordinating conjunction connects two independent clauses, it is often (but not always) accompanied by a comma. When the two independent clauses connected by a coordinating conjunction are nicely balanced or brief, many writers will omit the comma. The comma is always correct when used to separate two independent clauses connected by a coordinating conjunction.

Correlative (such as *not only ...but also; neither ... nor...; whether ... or...;both ...and ...; not ...but...;as...as...; either ...or...* are combined with other words, always travel in pairs, joining various sentence elements that should be treated as grammatically equal).

Conjunctive Adverbs (such as *however, moreover, nevertheless, consequently, as a result* are used to create complex relationships between ideas).

- **Subordinating** (sometimes is called a dependent word which comes at the beginning of a Subordinate (or Dependent) Clause and establishes the relationship between the dependent clause and the rest of the sentence. It also turns the clause into something that depends on the rest of the sentence for its meaning).

after although as as if	as long as as though because before	even if even though if if only in order that	now that once rather than since	so that than that though till	unless until when whenever	where whereas wherever while

! In general, there are six main types of conjunctions:

a) Addition: *Furthermore*, many people began to post things other than event listings—including job openings, for which Newmark soon set up a separate category on the list.

b) Result: *Thus* Rich Text Format (RTF), a format that includes most generic document features, is supported by most modern word processors.

c) Reason: *Owing to* the fact that more than one channel is used for downlink, the GPRS mobile phones make possible greater data transmission speeds.

d) Time: *Thirdly*, it is difficult to implement certain kinds of basic queries using SQL and relational databases, such as the shortest path between two points.

e) Example: *For instance*, if your code contains numbers with many digits, use an underscore character to separate digits in groups of three, like a comma, or a space, as a separator.

f) Opposition: *Although,* there is a boolean in some C++ compilers, boolean is not a basic type in C.

! For more extra information you can visit
http://grammar.ccc.commnet.edu/grammar/conjunctions.htm

10. Insert a suitable conjunction in each gap.

for example	***owing to***	***however***	***thus***
so that	***furthermore***	***although***	***or***

1)_______ you must be careful when mixing different data types, sometimes there are good reasons to do so.

2)_______ a machine with 32 address lines can handle up to 32 bits, 3) ______ 4 gigabytes (billions of bytes) worth of addresses.

4) _______ in the amount of time you spend trying to figure it out, you could already have the program finished in Visual Basic or C#.

5) Bind data to tests ______ you can use data-driven tests (6) ______ to search for keywords in a database).

7) _______ you should avoid using the converted value for computational purposes, 8) ______ the problems associated with the use of floating-point data types with monetary data.

B. Caution

! It is wise to use a cautious tone in your writing, because very often you are discussing issues in which there is no absolutely right answer, or absolutely correct definition, or absolutely perfect solution. If you present something as being the best way, it might easily be shown *not* to be the best way. Therefore, it is usually better to 'suggest', rather than 'state.

Areas where caution is particularly important include:

a) outlining a hypothesis that needs to be tested, (e.g. in an introduction);

b) discussing the results of a study, which may not be conclusive;

c) commenting on the work of other writers.

Here are some phrases that convey a cautious tone.

1.	Introductory verbs:	e.g. seem, tend, look like, appear to be, think, believe, doubt, be sure, indicate, suggest, estimate
2.	Certain lexical verbs	e.g. believe, assume, suggest
3.	Certain modal verbs:	e.g. will, must, would, may, might, could, can
4.	Adverbs of frequency	e.g. often, sometimes, usually, generally, occasionally

5.	Modal adverbs	e.g. probably, possibly, perhaps, conceivably (compare with less tentative adverbs like certainly, definitely, clearly)
6.	Modal adjectives	e.g. probable, possible (compare with less tentative adjectives like certain, definite, clear), likely, unlikely
7.	Modal nouns	e.g. assumption, possibility, probability
8.	That clauses	e.g. It could be the case that e.g. It might be suggested that e.g. It appears that e.g. It may be that e.g. It is likely that e.g. This suggests that
9.	To-clause + adjective	e.g. It may be possible to obtain e.g. It is important to develop e.g. It is useful to study

Another way to express caution is to use *quite*, *rather* or *fairly* before an adjective.

a *fairly* accurate summary

quite a significant correlation

a *rather* inconvenient location

NB. *quite* is often used before the article. It is often used positively, whereas *rather* tends to be

used negatively.

When referring to sources, the verb used indicates the degree of caution appropriate. Compare:

Before the launch of unlocked version, rumors *states* that unlocked version of iPhone … (positive)

Microsoft CEO *suggests* that more hardware would open… (cautious)

Other verbs that imply tentative or cautious findings are: think/consider/hypothesize/believe/claim/presume

11. Find the other examples in the text and underline them.

12. Rewrite the following sentences in a more cautious way.

1. Older students cope better with computer apps than younger ones.

2. Computer manuals are difficult to understand.

3. Application software, also known as an application or an app, is computer software designed to help the user to perform specific tasks.

4. There are cases where this would have been the only possible method of transmission.

5. These apps also make it easy to figure out what to remove if you're running out of storage space.

C. Formality in verbs

! A distinctive feature of academic writing style is choosing the more formal alternative when selecting a verb, noun, or other part of speech. English often has two, or more, choices to express an action or occurrence. The choice is often between a phrasal or prepositional verb (verb + preposition) and a single verb, the latter with Latinate origins. Often in lectures and other instances of everyday spoken English, the verb + preposition is used; however, for written academic style, the preferred choice is a single verb wherever possible. This is one of the most dramatic stylistic shifts from informal to formal style.

LESS FORMAL: Researchers *looked at* the way strain *builds up* around a fault.

ACADEMIC: Researchers *observed* the way strain *accumulates* around a fault.

13. A feature of most academic writing is a tendency to use rather formal verbs to express the writer's meaning accurately. Study the list below and find the meaning in each case.

NB. Some of these verbs, e.g. hold, are used in academic writing with a special meaning.

Verb	**Example of use**
to adapt	The simulation has also *been adapted* extensively for use in Asian ...
to arise	Inverse problems *arise* in many branches of science and mathematics, …
to carry out	The largest study was *carried out* as a sub-project of the Academy of ...
to characterise	Today's era is *characterised by* dynamic, yet focused IT partnerships
to clarify	AT helps to *clarify* any *contradictions* in an AS and…
to concentrate on	The final part of that *study concentrated on* the use of 'video-recall' ...

to be concerned with	The CSDM *Program is concerned primarily with* phenomena in the …
to demonstrate	… and new experimental evidence *has demonstrated* that **...**
to determine	The IDA evidence code was experimentally *determined*...
to discriminate	… a failure to *discriminate* between the GFP devices and SCPDs …
to emphasise	Participants *emphasised* that devices installed in their homes could be of ...
to establish	*Lifelong Learning* Programme 2007–2013 was *established* by Decision ...
to exhibit	Most *patients exhibited* advanced disease …
to focus on	… this term *focused on* integrated resource *management* …
to generate	Clients may *generate* byte-*range* requests without having received this header
to hold	Once you've added items to it, Melon will *hold* the bits and pieces of …
to identify	More than 30 reasons *have been identified* ...
to imply	The *previous research* on the information content *implied* that…
to indicate	The *survey indicates* that opportunities for new developments have ...
to interact	… in simple terms, this is how these *two* essential *systems interact*
to interpret	The criteria used to *interpret* the data gathered in the fieldwork and …
to manifest	Concrete examples of abstraction *as manifested* in students' ...
to overcome	Let us see what exactly is *computer* rage, and how you can *overcome* it
to predict	Mcommerce *study predicts* increase by 300 percent in 3 years ...
to propose	He *proposes* we can prevent overuse and poorly placed *QR codes.*
to prove	this paper *proves* that interactive systems are …
to recognise	IBM is internationally *recognized* as the *leading expert* in Netezza technology
to relate to	Factor analysis is *related* to principal component analysis (PCA), but the …
to supplement	The statement is *supplemented with* INTO itab …
to undergo	Whereas the WPD loop *underwent* significant conformational *changes* in ...
to yield	Such surveys *yield* single-digit percentages at best …

14. Choose a verb from the list that reduces the informality of each sentence. You may need to change the tense to fit the sentence.

assist	establish	reduce	increase	create
determine	investigate	fluctuate	raise	eliminate

1. Expert Systems can *help out* the user in the diagnosis of problems.
2. This program was *set up* to improve access to medical care.
3. Research expenditures have *gone up* to nearly $350 million.
4. Researchers have *found out* that this drug has serious side effects.
5. The use of optical character readers (OCRs) should *cut down* the number of problems with mail service.
6. Building a nuclear power plant will not *get rid of* the energy problem completely.
7. Researchers have been *looking into* this problem for 15 years now.
8. This issue was *brought up* during the investigation.
9. Engineers can *come up with* better designs using CAD.
10. The emission levels have been *going up and down*.

15. Think of a single verb that could be substituted for the prepositional verbs in the following sentences. Pay attention to the tense of the sentence.

1. The implementation of computer-integrated-manufacturing has *brought about* some serious problems.
2. The process should be *done over* until the desired results are achieved.
3. Plans are being made to *come up with* a database containing detailed environmental information for the region.
4. Subtle changes in the earth's crust were *picked up* by these new devices.
5. Proposals to construct new nuclear reactors have *met with* great resistance from environmentalists.

REFERENCES, USEFUL LINKS AND FURTHER READING

1. Academic Writing', by Malashenko, Elena, 2007.
2. 'Computer Desktop Encyclopedia copyright ©1981-2012 by The Computer Language Company Inc.
3. 'Burd, Barry. *Eclipse for Dummies.* Hoboken, N.J.: Wiley, 2004.
4. Beyon, Jeffrey Y. *LabVIEW Programming, Data Acquisition and Analysis.* Upper Saddle River, N.J.: Prentice Hall, 2000.
5. Hladni, Ivan. *Inside Delphi 2006.* Plano, Tex.: Wordware Publishing, 2006.

6. Foster-Johnson, Eric. *Graphical Applications with Tcl & Tk.* 2nd ed. New York: Hungry Minds, 1997.

7. Kernighan, Brian W., and Rob Pike. *The UNIX Programming Environment.* Englewood Cliffs, N.J.: Prentice Hall, 1984.

8. Lutz, Mark. *Learning Python.* 3rd. ed. Sebastapol, Calif.: O'Reilly, 2007.

9. Lutz, Mark. *Programming Python.* 3rd ed. Sebastapol, Calif.: O'Reilly, 2006.

10. Parsons, Andrew, and Nick Randolph. *Professional Visual Studio 2005.* Indianapolis: Wiley, 2006.

11. Python Programming Language—Official Web site. Available online. URL: http://www.python.org/. Accessed August 17, 2007.

12. Raimes, Ann. Grammar Troublespots. New York: Cambridge University Press, 2004, c.75.

13. Swales, John M. and Christine B. Feak. *Academic Writing for Graduate Students: Essential Tasks and Skills*. Ann Arbor: University of Michigan Press, 1994. 15-17.

UNIT 6

DATABASES

VOCABULARY

1. Match the words with their definitions:

1) ledger (n.)	['leʤə]	a) to get something back from the place where you left it
2) entity (n.)	['entɪtɪ]	b) in a database, anything about which information can be stored. Typically refers to a record structure
3) hypertext (n.)	[haɪpə(r)tekst]	c) essentially different in kind; not able to be compared
4) disparate (adj.)	['dɪsp(ə)rət]	d) an official document or speech which praises a person for something brave or special that they have done
5) retrieve (v.)	[rɪ'triːv]	e) a book or other collection of financial accounts
6) citation	[saɪ'teɪʃ(ə)n]	f) a software system that links topics on the screen to related information and graphics, which are typically accessed by a point-and-click method

Before you read

2. Discuss with your partner the following questions.

- **What do you know about databases?**
- **What are the reasons for using databases?**

3. Skim the text to check your ideas.

READING

HOW DATABASES WORK

People have kept track of information and data in many ways. Sometimes it was by using an elaborate file system, *a ledger*, or even a box. As computers become more popular and less expensive, many people find it easier to store information on the computer. The most common way to store large amounts of information with a computer is by using a database (DB) -- a structured collection of records or data that is stored in a computer system. The structure is achieved by organizing the data according to a database model. You can think of a database as an electronic filing system. A collection of information organized in such a way that a computer program can quickly select desired pieces of data.

In the early days of computing, a database generally consisted of a single file that was divided into data blocks that in turn consisted of records and fields within records. The COBOL language was (and is) particularly suited to reading, processing, and writing data in such files. This flat file database model is still used for many simple applications including “home data managers.” However, for more complex applications where there are many files containing interrelated data, the flat file model proves inadequate.

In 1970, computer scientist E. F. Codd proposed a relational model for data organization. In the relational model, data is not viewed as files containing records, but as a set of tables, where the columns represent fields and the rows individual *entities* (such as customers or transactions). Relational databases usually also enforce referential integrity. This means preventing changes to the database from causing inconsistencies. For example, if table A and table B are linked and a record is deleted from table A, any links to that record from records in table B must be removed. Similarly, if a change is made in a linked field in a table, records in a linked table must be updated to reflect the change. During the 1980s, the dBase relational database program became the most popular DBMS on personal computers. Microsoft Access is now popular on Windows systems, and Oracle is prominent in the UNIX world. Beginning in the 1980s, SQL (Structured Query Language) became a widely used standard for querying and manipulating data tables, and most DBMS implement SQL.

All computer databases are made up of tables. These tables contain records and each record is made up of some number of fields, columns and rows which are designed to provide an organized or arranged mechanism for managing, storing and retrieving information. A field is a single piece of information; a record is one complete set of fields; and a file is a collection of records. For example, a telephone book is analogous to a file. It contains a list of records, each of which consists of three fields: name, address, and telephone number. An alternative concept in database design is known as *Hypertext*. In a Hypertext database, any object, whether it is a piece of text, a picture, or a film, can be linked to any other object. Hypertext databases are particularly useful for organizing large amounts of *disparate* information, but they are not designed for numerical analysis.

To access information from a database, you need a database management system (DBMS). This is a collection of programs that enables you to enter, organize, and select data in a database.

Increasingly, the term database is used as shorthand for database management system.

Databases are useful as one can manipulate data, update records in bulk, perform complex calculations and *retrieve* records that match particular criteria.

The collected information could be in any number of formats (electronic, printed, graphic, audio, statistical, combinations). There are physical (paper/print) and electronic databases. A database could be as simple as an alphabetical arrangement of names in an address book or as complex as a database that provides information in a combination of formats.

For instance, bibliographic databases provide a descriptive record of an item, but the item itself is not provided in the database. Information about the item is provided, including such things as author, title, subject, publisher, etc. The information provided is called a *citation*. Sometimes a short summary or abstract of the item is provided as well. Examples of bibliographic databases include the GALILEO database Social Sciences Abstracts, or the Internet Movie Database on the World Wide Web. A full-text database provides the full-text of a publication. For example, Research Library in GALILEO provides not only the citation to a journal article, but often the entire text of the article as well. "College Source Online" offers full-text of 20,000 college catalogs, so rather than having to request a catalog from several colleges to make comparisons, you can gather information from all colleges you're interested in at one time.

Some databases provide numeric information, such as statistics or demographic information. Examples of these are (link will open in a pop-up window) Census Bureau databases and databases containing stock market information. You can also find databases that collect only image information (**EBSCOhost image collection**), audio information (MP3 or wav files), or a combination of any of the above types. CNN's site has a search option that provides access to news articles and the original video and audio files that accompanied them. Meta-databases are databases that allow one to search for content that is indexed by other databases. GOLD is an example of this kind of database. If you find a citation for an article in one of the bibliographic databases and want to determine if the article is available in full-text in another database, you could do a search for the journal in GALILEO in Journals A-Z to get a list of all the databases that index that specific publication.

There are thousands of different types and manufacturers of computer databases. A few common examples are Microsoft's Access, **Oracle** and **MySQL**. Some computer databases are completely free, while others cost tens of thousands of dollars. Databases are used around the world by nearly every business in the twenty-first century, including vast government databases that contain huge amounts of information about citizens.

If you're familiar with spreadsheets like Microsoft Excel, you're probably already accustomed to storing data in tabular form. It's not much of a stretch to make the leap from spreadsheets to databases. Just like Excel tables, database tables consist of columns and rows. Each column contains a different type of attribute and each row corresponds to a single record. For example, imagine that we were building a database table that contained names and telephone numbers. We'd probably set up columns named "FirstName", "LastName" and "TelephoneNumber." Then we'd simply start adding rows underneath those columns that contained the data we're planning to store. If we were building a table of contact information for our business that has 50 employees, we'd wind up with a table that contains 50 rows.

Databases are actually much more powerful than spreadsheets in the way you're able to manipulate data. Here are just a few of the actions that you can perform on a database that would be difficult if not impossible to perform on a spreadsheet:

- Retrieve all records that match certain criteria
- Update records in bulk
- Cross-reference records in different tables
- Perform complex aggregate calculations

You can correlate information from multiple tables in a database by creating foreign key relationships between the tables.

LANGUAGE DEVELOPMENT

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

1) flat file database	a)
2) interrelated data	b)
3) relational model	c)
4) manipulating data table	d)
5) access information	e)
6) data bulk	f)
7) physical database	g)
8) numeric information=numerical information	h)
9) pop-up window	i)
10) candidate key	j)

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

1) целостность ссылочных данных, ссылочная целостность	a)
2) отразить изменения	b)
3) обращаться с запросом к базе данных	c)
4) система управления базами данных, СУБД	d)
5) библиографическая база данных	e)
6) полнотекстовая база данных	f)
7) графическая информация	g)
8) суммарные вычисления	h)
9) внешний ключ	i)
10) таблица	j)

KEY CONCEPTS

EBSCO *host*

A powerful online reference system accessible via the Internet or direct connection. It offers a variety of proprietary full text databases and popular databases from leading information providers. The comprehensive databases range from general reference collections to specially-designed, subject-specific databases for public, academic, school, medical, corporate, and government libraries.

Oracle Database

An object-relational database management system (ORDBMS) produced and marketed by Oracle Corporation. Larry Ellison and his friends and former co-workers Bob Miner and Ed Oates started the consultancy Software Development Laboratories (SDL) in 1977. SDL developed the original version of the Oracle software. The name *Oracle* comes from the code-name of a CIA-funded project Ellison had worked on while previously employed by Ampex. The Oracle RDBMS stores data logically in the form of tablespaces and physically in the form of data files ("datafiles"). Tablespaces can contain various types of memory segments, such as Data Segments, Index Segments, etc. Segments in turn comprise one or more extents. Extents comprise groups of contiguous data blocks. Data blocks form the basic units of data storage. Oracle database management tracks its computer data storage with the help of information stored in the SYSTEM tablespace. The SYSTEM tablespace contains the data dictionary—and often (by default) indexes and clusters. A data dictionary consists of a special collection of tables that contains information about all user-objects in the database. Since version 8*i*, the Oracle RDBMS also supports "locally managed" tablespaces, which can store space management information in bitmaps in their own headers rather than in the SYSTEM tablespace (as happens with the default "dictionary-managed" tablespaces).

MySQL

A relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. It is named after developer Michael Widenius' daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. Members of the MySQL community have created several forks (variations) such as Drizzle, OurDelta, Percona Server, and MariaDB. All of these forks were in progress before the Oracle acquisition; Drizzle was announced eight months before the Sun acquisition. MySQL is a popular choice of database for use in web applications, and is a central component of the widely-used LAMP web application software stack — LAMP is an acronym for "Linux, Apache, MySQL, PHP". Its popularity is closely tied to the popularity of PHP. MySQL is used in some of the most frequently visited web sites on the Internet, including Flickr, Facebook, Wikipedia, Google— though not for searches, Nokia.com and YouTube.

6. Translate the following sentences into Russian.

1. Some can manipulate only one collection of data-a table---at a time; these database programs are called flat-file database managers.

2. When a firewall detects suspicious activity, it sends an alert in the form of a pop-up window or email to notify the computer's user or the network manager that someone might have tried to break in.

3. When a calculation uses an aggregate function, it's called an aggregate calculation which you create by defining a new calculated field.

4. In the context of relational databases, a foreign key is a referential constraint between two tables.

5. Referential integrity is a relational database concept in which multiple tables share a relationship based on the data stored in the tables, and that relationship must remain consistent.

7. Translate the following sentences into English.

1) Передача больших массивов данных это программный механизм, разработанный для перемещения большой объем данных путем методов блокирования, компрессии и буферизации для того, чтобы оптимизировать время переноса.

2) Запрос базы данных это часть программы (запрос), который/ая запрашивается в базе данных для того, чтобы получить оттуда информацию.

3) Данная работа исследует алгоритмы эффективного разделения множественных взаимосвязанных объектов данных параллельно узлам в компьютерной системе.

4) Внешний ключ, это поле в реляционной таблице, которое совпадает с потенциальным ключом другой таблицы.

5) Ссылочная целостность означает, что внешний ключ в любой таблице должен всегда соотноситься с соответствующей строкой в справочной таблице.

SPEAKING SECTION

8. Answer the following questions.

1. What is a database?

2. What is a research database?

3. What is the difference between a database and the Internet?

4. If a database is so much like a spreadsheet, why can’t we just use a spreadsheet?

5. What is DBMS?

6. What is a Database system?

7. The DBMS acts as an interface between what two components of an enterprise-class database system?

8. What are the advantages of DBMS?

9. What are the disadvantages in File Processing System?

10. Describe the three levels of data abstraction?

11. Who is MySQL named after? What does it stand for?

12. Why are there so many database models?

13. Are there undiscovered new models?

14. Is there an ultimate data model?

9. Prepare a presentation on the topic being discussed.

ACCURACY IN WRITING

ACCURACY (3)

A.Modals

! Modal verbs used in academic writing tend to have three main meanings:

a) **Ability**

May and *can* are similar but *can* is more common:

Study the following: *The assessment ... **may** be made in a variety of ways.*

b) **Degrees of certainty**

Will and *should* are used for predictions of near certainty (*will* is stronger):

Study the following: *Improved soft **should** lead to lower cost.*

May and *might* both suggest possibility:

Study the following: *Landfill carbon sequestration **might** supplement fossil fuel combustion.*

Would and *could* are used in conditional situations (not always with *if*):

Study the following: ... *estimates of the model's parameters **could** conceivably be computed*

Scale of Certainty

could/might	weak inference/ low possibility
may	stronger/perhaps, quite possibly
should	strong/ moderate possibility, probable
must	very strong/ certainly
will	strongest/ very certainly

Lower End of Scale of Probability

might may won't/ wouldn't can't/couldn't	low possibility ↓ Impossibility

c) Degrees of obligation

Must suggests strong obligation; *should* is for recommendations:

Study the following:

... to obtain a total estimate ... several approximations ***must*** *be used.*

...a primary research emphasis ... ***should*** *then be on identifying ...*

10. Complete the following sentences with a suitable modal of ability.

1.The question is whether democracy _______ survive in such difficult conditions.

2. Fifty years ago a new house ________ be bought for £1500.

3. Students _______be expected to write more than one long essay a week.

4. The mistakes of past historians _______now be clearly seen.

5. Jenkins (1976) argued that aluminum _______be used in place of steel.

11. Complete the following with a suitable modal of certainty.

1. It _______________ not be surprising if the company were bought by Microsoft.

2. Various situations _______________ lead to a user's loss of confidence.

3. Other studies confirm that a permanent shift in system use _______________ occur.

4. By 2020 most children _______________ have internet access by the age of five.

5. If the pressure is lowered, the reaction _____________ take place more quickly.

12. Use a suitable modal of obligation to complete the following.

1. Students studying abroad _______________ take some of their favourite gadgets with them.

2. All reports _______________ be returned to the main administrator by June 6th.

3. First-year undergraduates _______________ take at least three modules from the list below.

13. Use a modal verb in place of the underlined words and phrases. You might need to reconstruct the sentence rather than substitute words.

1. The scientist found it impossible to present her analysis because she had not collected all her data.

2. There is a low possibility that the shipment arrived yesterday afternoon.

3. It is quite likely that such software is available at the market.

4. Partners are likely to see benefits from his simulation program.

14. Write sentences about some of the people in the box, giving their nationality and answering what they are famous for.

Bill Gates	Hiroshi Yamauchi	Steven Paul Jobs	Linus Torvalds
Matias Duarte	Andy Rubin	Charles Babbage	Mark E. Dean

B. Nouns and adjectives

! Academic texts depend heavily on **adjectives** (which modify nouns and noun phrases), **adverbs** (see Accuracy 1) (which modify verbs, adjectives and other adverbs) and **nouns** in order to communicate meaning effectively.

Study the following and compare:

The *efficiency* of computer fans depends on the *precision* of their construction.

Precise construction results in an *efficient* computer fans.

The first sentence uses the nouns *efficiency* and *precision.* The second uses adjectives: *precise* and *efficient.* Although the meaning is similar, the first sentence is more formal. Effective academic writing requires accurate use of both forms, which can be easily confused.

15. Underline and correct the mistakes in the following:

1. Vigil is a very safety programming language, and an entry in the January 2013 PLT Games competition.

2. The data is typically organized to model relevancy aspects of reality.

3. Computer modeling has become an importance industry, generating hundreds of millions of dollars of revenues annually.

4. The Connecticut Better Business Bureau is warning residents about a potentially danger computer virus.

5. Note that it is enough to set the wide as the height will be resolved from the wide.

16. Complete the gaps in the table below.

Noun	Adjective	Noun	Adjective	Noun	Adjective
frequent		reliability		significance	
	important		reasonable		approximate
width		probability		necessity	
	particular		dangerous		relevant

The following adjectives are commonly used in academic texts. Try using them in your assignments in place of the more informal, colloquial expressions you use in conversation.

	Common Academic Adjectives	**Examples of use with nouns* most commonly found in academic texts**
Relating to **importance**	important / salient necessary significant	*salient* feature *significant* subsidy
Relating to **size/ amount / intensity / frequency**	high / increasing low / declining adequate / sufficient prime / main / primary / major only / sole annual / hourly indiscriminate	*sole* indicator *primary* consideration *only* restriction *indiscriminate* consumption *declining* interest *increasing* birthrate *annual* evaluation
Relating to **quality**	new / innovative economical consistent sustainable abstract hierarchical	*innovative* strategy *economical* estimate *consistent* representation *sustainable* solution *abstract* concept *hierarchical* organization
Relating to **variation**	different / alternative variable	*alternative* options *variable* dimensions
Relating to **probability**	likely / possible sure / definite / inevitable impossible	*likely* paradigm *possible* scenario *inevitable* outcome

* Note that these nouns are very common in academic texts. If you do not know their meanings, and how to use them, refer to the Examples Bank in the Longman Dictionary of Contemporary English software. Then try using them in your writing where appropriate.

C. Nouns: Countable and Uncountable

! Countable nouns are nouns that have both singular and plural forms. The following nouns, which are often used in academic writing, are countable:

idea	chapter	method	effect	issue	task
machine	thought	problem	report	scheme	article

Study the following:

*I have read two recent **articles** investigating the **issue** of security and its **effects**.*

Uncountable nouns have no plural form and therefore take a singular verb. The following nouns are usually uncountable:

information	knowledge	machinery	advice	research	nature
literature	environment	behaviour	labour	news	technology
progress	equipment	furniture	traffic	baggage	money

Study the following:

*A great deal of **research** on the **nature** of new computer virus has been conducted and it is encouraging to know that we have made some good **progress**.*

When you use a collective noun such as *data* you can choose either a singular verb or a plural verb depending on whether you want to emphasise the noun as a single unit or as a number of individuals or items.

Study the following:

*Data for the research **has** been gathered from a questionnaire survey and personal interviews.*

*Data from our study **provide** a firm basis to evaluate the effectiveness of the current model.*

17. Edit the incorrect statements.

1. We need to do more researches to find out the causes of the problem.

2. The department has purchased three new machineries for the main laboratory.

3. The Hong Kong Polytechnic University are planning to offer more training programmes next year.

4. The CEO has commissioned a study to gather people’s views on the new computer model.

REFERENCES, USEFUL LINKS AND FURTHER READING

1. Academic Writing’, by Malashenko, Elena, 2007.

2. About.com. Database Administration [links]. Available online. URL: http://databases.about.com/od/administration/Database_ Administration.htm. Accessed July 8, 2007.

3. Alapati, Sam R. Expert Oracle Database 10g Administration. Berkeley, Calif.: Apress, 2005.

4. Biber, D., Johansson, S., Leech, G., Conrad, S., & Finegan, E. (1999). Longman grammar of spoken and written English. Harlow, Essex: Pearson.

5. Coxhead, A. (2000). The new academic word list. TESOL Quarterly, 34 (2), 213-238.

6. Mannino, Michael A. Database Design, Application Development, and Administration. 3rd ed. New York: McGraw-Hill, 2005.

7. MySQL AB. MySQL Administrator's Guide and Language Reference. Indianapolis: MySQL Press, 2005.

8. http://www.uefap.com/writing/feature/complex_lexcomp.htm.

UNIT 7

NETWORK

VOCABULARY

1. Match the words with their definitions:

1) peer (n.)	[pɪə]	a) nearness in space or time
2) hub (n.)	[hʌb]	b) extend over so as to cover partly
3) deteriorate (v.)	[dɪ'tɪərɪəreɪt]	c)network node , router
4) overlap (v.)	[ˌəuvə'læp]	d) secretly listen to a conversation
5) thorough-fare(n.)	['θʌrəfɛə]	e) a device for connecting computers in a network
6) eavesdrop (v.)	['iːvzdrɔp]	f)become progressively worse
7) proximity (n.)	[prɔk'sɪmətɪ]	g)path forming a route between two places

Before you read

2. Discuss with your partner the following questions.

- **What do you know about network?**
- **What are the reasons for using networks?**

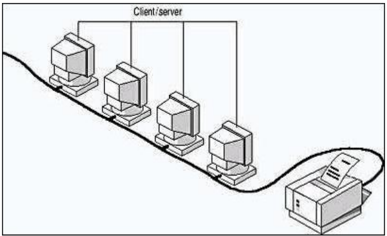
3. Skim the text to check your ideas.

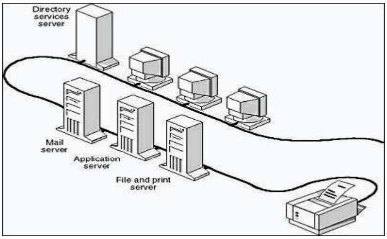
READING

NETWORKING

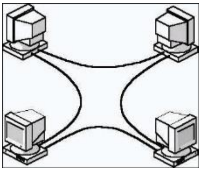
A network is a group of computers and other devices, such as printers and modems, connected to each other. This enables the computers to effectively share data and resources. The concept of sharing resources over a network is called networking. The computers in a network can share data, messages, graphics, printers, fax machines, modems, and other hardware and software resources.

In a **Peer-to-Peer Network**, there are no dedicated servers, and there is no hierarchy among the computers. All the computers are equal and therefore are known as *peers*. Each computer functions as both a **client** and a **server**. And there is no administrator responsible for the entire network. The user at each computer determines what **data** on that computer is **shared** on the network. In a peer-to-peer network, all computers are considered equal; they all have the same abilities to use the **resources** available on the network. Computers are not dedicated to function as servers. They use the network to share resources among the independent peers. The computer whose applications are required by the other networked computers functions as a server. The other computers function as clients. Therefore, a dedicated administrator is not assigned for network management. A peer-to-peer network is a small group of people using a network and performing similar tasks, which necessitates the sharing of resources. The users of each computer plan and control the security of their resources. The users determine the resources on their computers, which can be shared on the network. The shared network resources, such as disk space, printers or faxes, can be used by anyone who has access to the network. This is possible only if the shared network resources are not password protected. A peer-to-peer network does not support a central login process. Peer-to-peer networks are relatively simple. The users handle administration.





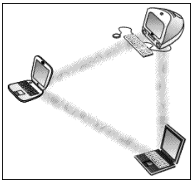
A dedicated server is one that functions only as a server and is not used as a client or workstation. **Server** Based **Networks** have become the standard models for networking. In a server-based network, clients rely on the services that the server provides, such as file storing and printing. Client computers are generally less powerful than server computers. A server-based network using network operating system is that the networks are organized into **domains**. In server-based networks, a network administrator centrally manages the resource security. The administrator defines and manages user access to network resources.



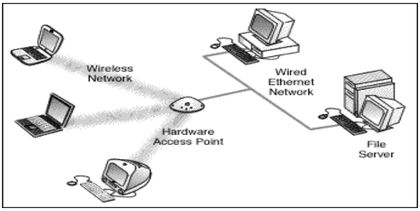
Local Area Network (LAN) is a network with two or more computers connected to each other in a single location.

Wired Local Area Network is the simplest type of network in which computers are connected to each other by cables. *Each of the computers on the LAN is also called a node.* A LAN is characterized by three primary attributes: **topology, medium, protocols**. Wireless local-area networks have the advantage of expandability**.** The topology and the medium used on a particular network are specified by the protocol.

There are a number of ways in which nodes can communicate over a network. The simplest is to establish a dedicated link between the transmitting and receiving stations. This technique is known as circuit switching. A better way of communicating is to use a technique known as **packet switching**, in which a dedicated path is not reserved between the source and the destination.

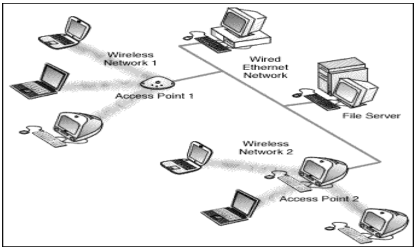


Wireless Local Area Network refers to technology that enables two or more computers to communicate using standard network protocols, but without network cabling. Each computer can communicate directly with all of the other wireless enabled computers. A wireless network can also use an access point, or base station. In this type of network the access point acts like a *hub*, providing connectivity for the wireless computers. It can connect the wireless LAN to a wired LAN, allowing wireless computer access to LAN resources, such as file servers or existing Internet Connectivity.



There are two types of access points: **Hardware access points** (HAP) offer complete support of most wireless features, but check your requirements carefully.

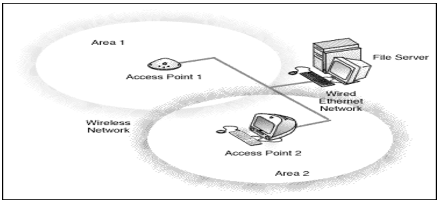
Software Access Points which run on a computer equipped with a wireless network interface card as used in peer-to-peer wireless network. The software routers that can be used as a basic Software Access Point, and include features not commonly found in hardware solutions.



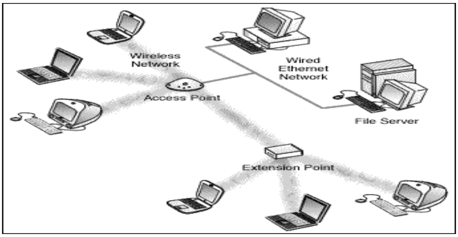
Wireless networking offers a cost-effective solution to users with difficult physical installations such as campuses, hospitals or businesses with more than one location in immediate *proximity* but separated by public *thoroughfare*. This type of installation requires two access points. Each access point acts as a bridge or router connecting its own LAN to the wireless connection. The wireless connection allows the two access points to communicate with each other, and therefore interconnect the two LAN's. Each access point has a finite range within which a wireless connection can be maintained between the client computer and the access point. The actual distance varies depending upon the environment. Also it should be noted that when operating at the limits of range the performance may drop, as the quality of connection *deteriorates* and the system compensates. Typical indoor ranges are 150-300 feet, but can be shorter if the building construction interferes with radio transmissions. Longer ranges are possible, but performance will degrade with distance. Outdoor ranges are quoted up to 1000 feet, but again this depends upon the environment.

In most cases, separate access points are interconnected via a wired LAN, providing wireless connectivity in specific areas such as offices or rooms, but connected to a main wired LAN for access to network resources, such as file servers. If a single area is too large to be covered by a single access point, then multiple access points or extension points can be used.

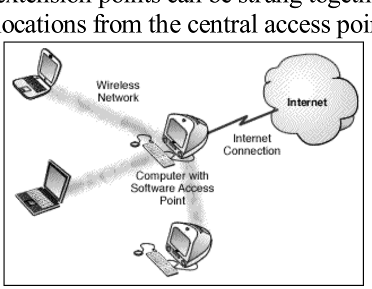
When using multiple access points, each access point wireless area should overlap its neighbors. This provides a seamless area for users to move around in using a feature called "roaming." A wireless computer can "roam" from one access point to another, with the software and hardware maintaining a steady network connection by monitoring the signal strength from in-range access points and locking on to the one with the best quality.



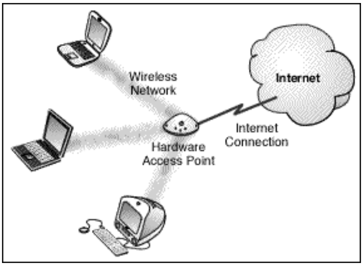
Some access point configurations require security authentication when swapping access points, usually in the form of a password dialog box. Access points are required to have overlapping wireless areas to achieve this. *A user can move from Area 1 to Area 2 transparently. The Wireless networking hardware automatically swaps to the Access Point with the best signal.*



Some manufacturers produce extension points, which act as wireless relays, extending the range of a single access point. Multiple extension points can be strung together to provide wireless access to far away locations from the central access point.



To share an Internet connection across a LAN you need two things: an Internet sharing hardware device or software program and a LAN. Any computer equipped with a wireless network card running suitable Internet sharing software can be used as a software access point. A number of

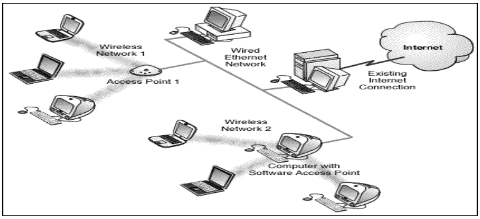


vendors offer hardware access points.

A hardware access point may provide Internet Sharing capabilities to Wired LAN computers, but does not usually provide much flexibility beyond very simple configurations.

If an existing wired LAN already has an Internet connection, then the hardware access points simply connect to LAN and allow wireless computers to access the existing Internet connection in the same way as wired LAN computers.

Wireless communications obviously provide potential security issues, as an intruder does not need physical access to the traditional wired network in order to gain access to data communications. However, 802.11 wireless communications cannot be received much less decoded by simple scanners, short wave receivers etc. This has led to the common misconception that wireless communications cannot be *eavesdropped* at all. However, eavesdropping is possible using specialist equipment.



To protect against any potential security issues, 802.11 wireless communications have a function called **WEP**. Wireless networking hardware requires the use of underlying technology that deals with radio frequencies as well as data transmission. The most widely used standard is 802.11 produced by the Institute of Electrical and Electronic Engineers (IEEE). This is a standard defining all aspects of Radio Frequency Wireless networking.

It should also be noted that traditional Virtual Private Networking (VPN) techniques will work over wireless networks in the same way as traditional wired networks.

http://f1tutorials.com/

LANGUAGE DEVELOPMENT

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

1) dedicated server	a)
2) peer-to-peer network	b)
3) circuit switching	c)
4) packet switching	d)
5) node	e)

6) receiving station	f)
7) dedicated path	g)
8) wireless network access	h)
9) intruder	i)

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

1) сетевой компьютер, использующий ресурсы сервера)	a)
2) серверная сеть	b)
3) передающая станция	c)
4) точка доступа	d)
5) беспроводная передача данных	e)
6) экономичное (техническое) решение	f)
7) прямое обращение	g)
8) режим поддержания постоянной связи при перемещениях абонента;	h)
9) коротковолновый приемник	i)

KEY CONCEPTS

Clients

Computers that can access the shared network resources provided by a server.

Domain

A collection of networks and clients that share security information. Domain security and logon permissions are controlled by special servers called domain controllers. Users cannot access the resources of servers in a domain until a domain controller has authenticated them.

LAN

Local Area Network is a group of personal computers and associated equipment that are linked by cable, for example in an office building, and that share a communications line. LAN is an abbreviation for `

Media

The hardware components, such as cables with the help of which computers in a network are connected to each other. Four basic types of media are used in local-area networks; coaxial cable, twisted-pair wires, fiber-optic cable, and wireless. Each medium has its advantages and disadvantages relative to cost, speed, and expandability. Coaxial cables provide high speed and low error rates. Twisted-pair wires are cheaper than coaxial cables, can sustain the speeds common to most personal computers, and are easy to install. Fiber-optic cable is the medium of choice for high-speed local-area networks.

Peer-to-Peer Network

The peer-to-peer networks support 10 computers. The users in a peer-to-peer network are located in the same geographical area. Operating systems, such as Microsoft Windows 98 or Microsoft Windows XP, can be used to set up a peer-to-peer network. Additional software is not required because peer-to-peer networking is built into the systems. Peer-to-peer networks have weak and intrusive security because a central server is not used to administer and secure the network. This implies that a user who logs on to one peer can access any shared network resource, which is not controlled by a specific password. Peer-to-peer networks are appropriate for environments where all the users are located in the same geographical area and the network security is not an important factor. In addition, these networks are useful when the network expansion is limited.

Packet switching

Paul Baran of RAND develops the idea of distributed, packet-switching networks. Networks such as the Internet use packet-switching: Data is sent as individual packets that contain a “chunk” of data, an address, and an indication of where the data fits within the message as a whole. The packets are routed at the routers using software that tries to find the fastest link to the destination. When the packets arrive at the destination, they are reassembled into the original message. A typical packet is divided into preamble, address, control, data, and error-check fields. The computers in a LAN are connected by using cables. This method cannot be used to connect computers that are in different locations, for example, in buildings across a town or city. Therefore, a LAN is not suitable for large businesses with offices in several locations.

Protocol

A set of rules governing the exchange or transmission of data between devices.

Resources

Files, printers or other items that can be used by network users. These resources can be either hardware or software resources.

Servers

Computers that provide the shared resources to network users. There is usually only one server in a small network, but bigger networks may have more than one server.

Server Based Networks

The beneficial of server-based networks is central file storage. Server-based networks provide easy backup of critical data. Data backup is another useful characteristic of server based networks. Server based networks can support a larger number of users than peer-to-peer networks. To support a large number of users, server-based networks use monitoring and network management tools. Servers must perform varied and complex tasks.

Security is often the primary reason for choosing a server-based approach to networking. In a server-based environment, one administrator who sets the policy and applies it to every user on the network can manage security.

Shared data

A server provides shared resources and data over a network. The files (a document, a worksheet or a folder) that are provided by the server over the network are called shared data.

Software Access Points

To connect wireless LAN to wired LAN you will need some sort of bridge between the wireless and wired network. This can be accomplished either with a hardware access point or a software access point. Hardware access points are available with various types of network interfaces, such as Ethernet or Token Ring, but typically require extra hardware to be purchased if you're networking requirements change. If networking requirements go beyond just interconnecting a wired network to a small wireless network, a software access point may be the best solution. A software access point does not limit the type or number of network interfaces you use. It may also allow considerable flexibility in providing access to different network types, such as different types of Ethernet, Wireless and Token Ring networks. Such connections are only limited by the number of slots or interfaces in the computer used for this task. Further to this the software access point may include significant additional features such as shared Internet access, web caching or content filtering, providing significant benefits to users and administrators.

Topology

The pattern used to connect the computers together. With a bus topology, a network cable connects each computer to the next one, forming a chain. With a star topology, each of the computers is connected to a central nexus called a hub/Switch. A ring topology is essentially a bus network with the two ends joined together.

Wired Local Area Network

A WLAN computer network is two or more computers that communicate with each other through some medium. The primary usage of local-area networks (LANs) is the sharing of hardware, software, or information, such as data files, multimedia files, or electronic mail. Resource sharing provided by local-area networks improves efficiency and reduces overhead.

Wired Equivalent Privacy (WEP)

A form of encryption which provides privacy comparable to that of a traditional wired network. If the wireless network has information that should be secure then WEP should be used, ensuring the data is protected at traditional wired network levels.

Wireless LAN (WLAN)

A user could work with a laptop on the deck or patio while still having access to a high-speed Internet connection. Typically, a wireless LAN uses a frequency band with each unit on a slightly different frequency, thus allowing all units to communicate without interference. Peer-to-peer wireless network consists of a number of computers each equipped with a wireless networking interface card. They can share files and printers this way, but may not be able to access wired LAN resources, unless one of the computers acts as a bridge to the wired LAN using special software.

6. Translate the following sentences into Russian.

1)There are ways to extend the basic operating range of Wireless communications, by using more than a single access point or using a wireless relay /extension point.

2)Multiple access points can be connected to a wired LAN, or sometimes even to a second wireless LAN if the access point supports this.

3) The reliable and cost-effective test solution for military computer devices with PCI, ISA and MULTIBUS I bus based on StarFabric technique.

4) In wireless telecommunications, roaming is a general term referring to the extension of connectivity service in a location that is different from the home location.

5) I would like to expand the range of my @home hotspot with seamless handoff between my access points without dropping a call.

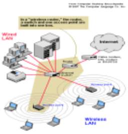
7. Translate the following sentences into English.

1. Коммутация каналов — это наиболее известная технология, используемая для построения сети связи.
2. В компьютерной сети, беспроводная точка доступа — это беспроводная базовая станция, предназначенная для обеспечения беспроводного доступа к уже существующей сети при помощи сертификата Wi-Fi.
3. Необходимо ограничить несанкционированный доступ к учётной записи пользователя.
4. Этот крошечный коротковолновый приемник отличается поразительным диапазоном чувствительности от 40 килогерц до 30 мегагерц, избирательностью и способностью подавлять помехи.
5. Узнайте больше о прямом обращении к многосервисной мобильной платформе, и что внутренние и внешние пользователи вашей профессиональной сети говорят об этом.

SPEAKING SECTION

8. Answer the following questions.

1. Why are Peer-to-peer networks relatively simple?
2. What does it mean that the users handle administration?
3. What are the advantages and disadvantages of a peer-to-peer network?
4. What are domain controllers?
5. What are the advantages and disadvantages of Server Based Networks?

9. 
From Computer Desktop Encyclopedia
© 2007 The Computer Language Co. Inc.
In a "wireless router," the router, a switch and one access point are built into one box.
Wired LAN
Internet
Router
Access point
Access point
Wireless LAN
Prepare a presentation on the topic being discussed.

ACCURACY (4)

A. Passives

! Often in academic writing, we **don't want to focus on** who is doing an action, but on who is receiving or experiencing the action. The passive voice is thus extremely useful in academic writing because it allows writers to highlight the most important participants or events within sentences by placing them at the beginning of the sentence.

Study the following:

This research was done *in 2006*. (passive)

My teacher did this research. (active)

In the first sentence, the emphasis is on the date, in the second on the writer. So the passive is used in written English when the cause (a person or thing) is less important or unknown.

In addition, in academic writing sometimes it is obvious, irrelevant or repetitive to state who the 'doer' of the sentence is: thus the passive voice is a useful way to construct these types of sentences.

Study the following:

The programme *will be completed* next year. (by someone)

It is quite common to show the cause of the action by adding *by …*

Study the following:

The project was helped *by good team spirit.*

It is also a way that the use of informal personal pronouns (I, We, You) can be avoided.

The findings *were evaluated.*

An analysis *will be made.*

10. Change the following into the passive.

1. Relationship type defines a set of associations or a relationship set among a given set of entity types.

2. VDL specifies user views and their mappings to the conceptual schema.

3. Query optimization is the phase that identifies an efficient execution plan for evaluating a query.

4. Flat file database provides user-interface management.

5. The information in the data dictionary validates the existence of the objects, provides access to them, and maps the actual physical storage location.

! An adverb is often inserted in a passive form.

Study the following:

This process *is commonly called* ‘networking’.

11. Change the following sentences from active to passive and insert a suitable adverb from the box below.

1. A virus damaged 40% of the information on the computer.
2. Bill Gates runs the day-to-day operation of Microsoft.
3. They had built the company in 1975.
4. They provided iPods for all students.
5. They tested over 550 computers for this virus over a three-year period.
6. The researchers calculated the percentages to three decimal places.
7. They called their business the Universal Computer Company.

conveniently optimistically helpfully brilliantly
regularly precisely efficiently badly

B. PREFIXES AND SUFFIXES

! Adding affixes to existing words (the base) to form new words is common in academic English. If you know how prefixes and suffixes affect word meaning it is easy to understand the word. Prefixes are added to the front of the base (*like* → *dislike*), whereas suffixes are added to the end of the base (*active* → *activate*). Prefixes usually do not change the class of the base word, but suffixes usually do change the class of the word. Prefixes change or give the meaning. Suffixes show the meaning or the word class.

The most common prefixes used to form new verbs in academic English are: *re-, dis-, over-, un-, mis-, out-*. The most common suffixes are: *-ise, -en, -ate, -(i)fy*. By far the most common affix in academic English is *–ise,* e.g. *verbs + prefix* → *verb*

Prefix	Meaning	Examples
dis-	reverses the meaning of the verb	disappear, disallow, disarm, disconnect, discontinue
over-	too much	overbook, oversleep, overwork
un-	reverses the meaning of the verb	unbend, uncouple, unfasten
mis-	badly or wrongly	mislead, misinform, misidentify

out-	more or better than others	outperform, outbid
be-	make or cause	befriend, belittle
co-	together	co-exist, co-operate, co-own
de-	do the opposite of	devalue, deselect
fore-	earlier, before	foreclose, foresee
inter-	between	interact, intermix, interface
post-	after in time or order	postpone
pre-	before	pre-expose, prejudge, pretest
sub-	under/below	subcontract, subdivide
trans-	across, over	transform, transcribe, transplant
under-	not enough	underfund, undersell, undervalue, underdevelop

e.g. Suffix used to form verbs with the meaning "cause to be".

Suffix	Example
-ise	stabilise, characterise, symbolise, visualise, specialise
-ate	differentiate, liquidate, pollinate, duplicate, fabricate
-fy	classify, exemplify, simplify, justify
-en	awaken, fasten, shorten, moisten

The most common prefixes used to form new nouns in academic English are: *co-* and *sub-*. The most common suffixes are: *-tion, -ity*, *-er, -ness, -ism, -ment, -ant, -ship, -age, -ery*. By far the most common noun affix in academic English is *-tion.*

e.g. noun+prefix → noun

Prefix	Meaning	Examples
anti-	against	anticlimax, antidote, antithesis
auto-	self	autobiography, automobile
bi-	two	bilingualism, biculturalism, bi-metalism
co-	joint	co-founder, co-owner, co-descendant
counter-	against	counter-argument, counter-example, counter-proposal
dis-	the converse of	discomfort, dislike
ex-	former	ex-chairman, ex-hunter
hyper-	extreme	hyperinflation, hypersurface
in-	the converse of	inattention, incoherence, incompatibility
in-	inside	inpatient,
inter-	between	interaction, inter-change, interference
kilo-	thousand	kilobyte
mal-	bad	malfunction, maltreatment, malnutrition
mega-	million	megabyte

mis-	wrong	misconduct, misdeed, mismanagement
mini-	small	mini-publication, mini-theory
mono-	one	monosyllable, monograph, monogamy
neo-	new	neo-colonialism, neo-impressionism
out-	separate	outbuilding,
poly-	many	polysyllable
post-	after in time or order	post-date, post-operative
pre-	before	prehistory
pseudo-	false	pseudo-expert
re-	again	re-organisation, re-assessment, re-examination
semi-	half	semicircle, semi-darkness
sub-	below	subset, subdivision
super-	more than, above	superset, superimposition, superpowers
sur-	over and above	surtax
tele-	distant	telecommunications,
tri-	three	tripartism
ultra-	beyond	ultrasound
under-	below, too little	underpayment, under-development, undergraduate
vice-	deputy	vice-president

e.g. Suffix added to a verb (V), noun (N) or adjective (A) → noun

Suffix	Meaning	Examples
-tion	action/instance of V-ing	alteration, demonstration
-ity	state or quality of being A	ability, similarity, responsibility
-er	person who V-s something used for V-ing person concerned with N	advertiser, driver computer, silencer astronomer, geographer
-ness	state or quality of being A	darkness, preparedness, consciousness
-ism	doctrine of N	Marxism, Maoism, Thatcherism
-ment	action/instance of V-ing	development, punishment, unemployment
-ant/-ent	person who V-s	assistant, consultant, student
-ship	state of being N	friendship, citizenship, leadership

-age	collection of N action/result of V	baggage, plumage breakage, wastage, package
-ery/-ry	action/instance of V-ing place of V-ing	bribery, robbery, misery refinery, bakery
-wards	means in the direction of	northwards

Many adjectives are formed from a base of a different class with a suffix (e.g. *-less, -ous*). Adjectives can also be formed from other adjectives, especially by the negative prefixes (*un-, in- and non-*).

The most common suffixes are -al, -ent, -ive, -ous, -ful, -less.

e.g. Suffix added to verbs or nouns →adjective

Suffix	Examples
-al	central, political, national, optional, professional
-ent	different, dependent, excellent
-ive	attractive, effective, imaginative, repetitive
-ous	continuous, dangerous, famous
-ful	beautiful, peaceful, careful
-less	endless, homeless, careless, thoughtless
-able	drinkable, countable, avoidable,

e.g. adjective + negative →adjective

Prefix	Examples
un-	unfortunate, uncomfortable, unjust
im-/in-/ir-/il-	immature, impatient, improbable, inconvenient, irreplaceable, illegal
non-	non-fiction, non-political, non-neutral
dis-	disloyal, dissimilar, dishonest

e.g. base with both prefix and suffix

	-able	*-tion*	*-tive*	*-ment*	*-ar*
un-	uncomfortable unavoidable unforgettable		unimaginative		
mis-		misinformation		misjudgement mismanagement	
re-	recoverable recyclable	reformulation	reproductive	realignment repayment	
in-	irreplaceable	incoordination inattention	inactive inoperative		
dis-		disconnection		disappointment	
semi-			semiconductive		semi-circular

12. Suggest possible meanings for the words in italic.

1) Children who have little or no access to computers and the internet could end up in a "technology *underclass*", teachers warn.

2) When we arrived, we were told that due to a virus in the computer there was a problem of overbooking but we did not have to worry because they had made a reservation for us in another hotel.

3) The EPIC *Microclimate* System (MCS) provides electronics cooling in ambient temperatures up to 170°F (77°C).

4) It is claimed that computers have created a *post-industrial* economy.

5) Phone numbers that are not listed in the phone book or 118 directory services are often known as *ex-directory*.

6) The *unavailability* of the product is due to the *exceptional* weather.

13. Give the word class and suggest possible meanings for:

1) cancellation	
2) unpredictable	
3) coincidental	
4) saleable	
5) uncooperatively	
6) interviewee	
7) evolutionary	
8) surrealism	
9) protester	
10) symbolically	

C. Prepositions

14. Underline the prepositions in the following text.

The purpose of this paper is to examine the spatial evolution of computers across 317 metro areas in the USA since the introduction of the personal computer. We start by examining the relative distribution of employment across cities, examining how that distribution changes in 1977. 1992 and how cities move through the distribution. For computers, transition matrices are stationary, with the industry exhibiting no tendency to settle down, nor any tendency of retrenchment during periods of national high-tech employment decline. There is no tendency of the relative size distribution of computer employment to collapse, go bimodal, etc. Overall computers exhibit some turbulence, with dramatic big winners and losers among cities, as well as persistence for some cities in employment shares. In attracting or repelling an industry, urban heterogeneity is important. Large, well educated cities near San Jose have a much greater chance of attracting high-tech employment (much lower mean first passage times moving up states) and less chance of losing it. In assessing the determinants of persistence in local employment patterns we examine sources of productivity growth. In conclusion, the paper sets out to find strong evidence of significant dynamic own industry externalities for single plant firms and little evidence of urbanization-Jacobs-knowledge type externalities.

15. Find one example for each way of using prepositions in the text above.

noun + preposition	
verb + preposition	
Adjective(+noun)+ preposition	
phrasal verb	
preposition of place	
preposition of time	
phrase	

NB! The difference between phrasal verbs and verbs with prepositions: The cars are *made in* Korea. (verb + preposition = easy to understand) The writer *made up* the story in a night. (phrasal verb = hard to understand).

16. Study these further examples of preposition use and decide on their type.

1. There are a number *of* viruses to be considered ... (noun +)
2. The results would be applicable to all software programs … (____________)
3. … the data was gathered from a questionnaire (____________________)
4. All the items were placed within their categories (____________________)
5. The results of the investigation are still pertinent … (____________________)
6. The respondents had spent on average 4.9 years … (____________________)
7. You can still buy *computers* made in the *USA* … (____________________)
8. … withi*n* a short period of time (____________________)

17. The following verbs are generally used with these prepositions. Think on your sentences and write them down in the table.

Verb + preposition	
add to	
agree with	

associate with	
believe in	
blame for	
concentrate on*	
consist of	
depend on*	
derive from	
divide into	
invest in	
learn from	
pay for	
point out	
specialise in	

* *focus* on and *rely on* are similar.

NB! With the following verbs more than one preposition is possible. Note the change of meaning in some cases: *compare to/with, look at/into, look for, apply to/for.*

18. Insert a suitable preposition before or after the nouns in the sentences below.

1. The suit is presented in support ____ the value of programmer's work.
2. A small change ____ soft demand can lead to large price rises.
3. He applied ____the committee for a grant.
4. The standard icons can be found ____ many computer systems and mobile devices.
5. Through our research we have collected a range ____definitions of these soft skills.

6. Another study gives further support ____ this viewpoint.

7. All modern computers derive ____ wartime decoding machines.

8. From a brand architecture viewpoint, the company maintains a "monolithic" brand identity - everything being associated ____ the Apple.

9. Most students use search engines to look ____ information.

10. The evidence needs to be looked ____more carefully.

19. Complete the following phrases with the correct prepositions.

1. ____the whole
2. point ____view
3. in respect ____
4. ____spite of
5. in support ____

20. Complete the following sentences with suitable prepositions of place or time.

1. Deciphering machine language code or writing it is as complex a task as one can tackle ____ computing.

2. It also takes into account the differences in color between pixels, often those in or ____ transition areas.

3. One table tracks section properties, such as headers that appear ____ the top of each page, tab settings, and whether the page orientation is portrait (vertical) or landscape (horizontal).

4. Bill Gates was born ____ October 28, 1955.

5. It was designed to inject competition between insurance companies ____ Spain ____ early retirement, i.e. before the normal UK pensionable age.

21. Complete the text with suitable prepositions.

Because networking is such a broad and complex field, no single event represents its point ____origin. We can think ____ the 1960s as the early period, however, because that's when the digital computer began to significantly affect the lives ____ordinary individuals and the operations ____ businesses and government.

____ the 1960s, computer networking was essentially synonymous ____ mainframe computing and telephony services, and the distinction ____ local and wide area networks did not yet exist. Mainframes were typically "networked" to a series _____ dumb terminals _______ serial connections running ____ RS-232 or some other electrical interface. If a terminal ____ one city needed to connect ____ a mainframe ____ another city, a 300-baud long-haul modem would use the existing analog Public Switched Telephone Network (PSTN) to form the connection. The technology was primitive indeed, but it was an exciting time nevertheless.

____May, 1974, the Institute of Electrical and Electronic Engineers (IEEE) published a paper titled "A Protocol ____ Packet Network Interconnection." The paper's authors - Vinton Cerf and Robert Kahn - described a protocol called *TCP* that incorporated both connection-oriented and datagram services.

REFERENCES, USEFUL LINKS AND FURTHER READING

1. Academic Writing', by Malashenko, Elena, 2007.

2. 'Computer Desktop Encyclopedia copyright ©1981-2012 by The Computer Language Company Inc.

UNIT 8

THE INTERNET &WORLD WIDE WEB

VOCABULARY

1. Match the words with their definitions:

1) snarky(adj.)	['snɑːkɪ]	a) present, appearing, or found everywhere
2) intertwine (v.) with obj.	[ˌɪntə'twaɪn]	b) increase very steeply or rapidly
3) launch (v.)	[lɔːnʧ]	c) sharply critical (inf., AmE.)
4) a missile	['mɪsaɪl]	d) to be closely connected
5) skyrocket (v.)	['skaɪˌrɔkɪt]	e) a weapon that is self-propelled or directed by remote control, carrying conventional or nuclear explosive
6) ubiquitous (adj.)	[juː'bɪkwɪtəs]	f) send (a missile, satellite, or spacecraft) on its course
7) harassment (n.)	['hærəsmənt], [hə'ræsmənt]	g) behaviour which is intended to trouble or annoy someone

Before you read

2. Discuss with your partner the following questions.

- **What do you know about the Internet and the World Wide Web?**
- **What are the reasons for using the Internet?**

3. Skim the text to check your ideas.

READING

THE INTERNET OR WWW?

The Internet has become so *ubiquitous* it's hard to imagine life without it. It's equally hard to imagine a world where "www" isn't the prefix of many of our online activities. But just because the **Internet** and the World Wide Web are firmly *intertwined with* each other, it doesn't mean they're synonymous. Let's go back to when it all began.

Mention the history of the Internet to a group of people, and chances are someone will make a *snarky* comment about Al Gore claiming to have invented it. Gore actually said that he "took the initiative in creating the Internet". He promoted the Internet's development both as a senator and as vice president of the United States. So how did the Internet really get started? Believe it or not, it all began with a satellite. It was 1957 when the then Soviet Union *launched* Sputnik, the first man-made satellite. Americans were shocked by the news. The Cold War was at its peak, and the United States and the Soviet Union considered each other enemies. If the Soviet Union could launch a satellite into space, it was possible it could launch a *missile* at North America. President Dwight D. Eisenhower created the **ARPA** in 1958 as a direct response to Sputnik's launch. ARPA's purpose was to give the United States a technological edge over other countries. One important part of ARPA's mission was computer science. In the 1950s, computers were enormous devices that filled entire rooms. They had a fraction of the power and processing ability you can find in a modern PC. Many computers could only read magnetic tape or punch cards, and there was no way to network computers together. ARPA aimed to change that. It enlisted the help of the company Bolt, Beranek and Newman (BBN) to create a computer network. The network had to connect four computers running on four different operating systems. They called the network ARPANET. By October 29, 1969, the first ARPANET network connection between two computers was launched -- and promptly crashed. But happily, the second time around was much more successful and the Internet was born. More and more computers were added to this ever-increasing network and the megalith we know today as the Internet began to form. Although other groups were working on ways to network computers, ARPANET established the protocols used on the Internet today. Moreover, without ARPANET, it may have taken many more years before anyone tried to find ways to join regional networks together into a larger system. In 1973, engineers began to look at ways to connect ARPANET to the PRNET. A packet radio network connects computers through radio transmitters and receivers. Instead of sending data across phone lines, the computers use radio waves. It took three years, but in 1976 engineers successfully connected the two networks. Technicians joined the SATNET to the other two networks in 1977. They called the connection between multiple networks inter-networking, or the Internet for short. Other early computer networks soon joined. They included **USENET**, **BITNET**, **CSNET** and **NSFNET**. In 1990, Tim Berners-Lee developed a system designed to simplify navigation on the Internet. In time, this system became known as the World Wide Web. It didn't take long for some people to mistakenly identify the Internet and the Web as the same thing. The Internet is a global interconnection of computer networks; the World Wide Web is a way to navigate this massive network. In sailing terms, it's like comparing an ocean to a ship.

Most early Internet users were government and military employees, graduate students and computer scientists. Using the World Wide Web, the Internet became much more accessible. Colleges and universities began to connect to the Internet, and businesses soon followed. The creation of the World Wide Web came with the help of a man named Tim Berners-Lee. In 1990, he developed the backbone of the World Wide Web -- the **HTTP**. People quickly developed browsers which supported the use of HTTP and with that the popularity of computers *skyrocketed.* In the 20 years during which ARPANET ruled the Internet, the worldwide network grew from four computers to more than 300,000. By 1992, more than a million computers were connected -- only two years after HTTP was developed.

You might be wondering at this point what exactly HTTP is -- it's simply the widely used set of rules for how files and other information are transferred between computers. So what Berners-Lee did, in essence, was determine how computers would communicate with one another. For instance, HTTP would've come into play if you clicked the source link in the last paragraph or if you typed the http://www.iseu.by **URL** into your browser to get to our university page. But don't get this confused with Web page programming languages like **HTML** and **XHTML**. We use those to describe what's on a page, not to communicate between sites or identify a Web page's location.

So, if there is any difference between www and the Internet? To answer this question, let's look at each element. Simply, the Internet is a network of networks -- and there are all kinds of networks in all kinds of sizes. You may have a computer network at your work, at your university or even one at your house. These networks are often connected to each other in different configurations, which is how you get groupings such as LANs and regional networks. Your cell phone is also on a network that is considered part of the Internet, as are many of your other electronic devices. And all these separate networks -- added together -- are what constitute the Internet. Even satellites are connected to the Internet.

The World Wide Web, on the other hand, is the system we use to access the Internet. The Web isn't the only system out there, but it's the most popular and widely used. (Examples of ways to access the Internet without using HTTP include e-mail and instant messaging.) As mentioned on the previous page, the World Wide Web makes use of hypertext to access the various forms of information available on the world's different networks. This allows people all over the world to share knowledge and opinions. We typically access the Web through browsers, like Internet Explorer and Mozilla Firefox. By using browsers like these, you can visit various Web sites and view other online content.

So another way to think about it is to say the Internet is composed of the machines, hardware and data; and the World Wide Web is what brings this technology to life.

LANGUAGE DEVELOPMENT

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

1) instant messaging	a)
2) ever-increasing network	b)
3) online content	c)
4) allotment	d)
5) online defamation	e)

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

1) программа просмотра Web, навигатор	a)
2)межсетевой обмен, межсетевое взаимодействие	b)
3)протокол передачи гипертекстовых файлов	c)
4) экспериментальная сеть	d)
5) нарушение авторского права	e)

6. Find and learn English and Russian the definitions for the following abbreviations

1) ARP	a)
2) ARPANET	b)
3) PRNET	c)
4) SATNET	d)
5) USENET	e)

6) BITNET	f)
7) CSNET	g)
8) NSFNET	h)
9) WWW	i)
10) HTTP	j)
11) URL	k)

KEY CONCEPTS

ARPA

The U.S. Defense Department's ARPA (Advanced Research Projects Agency) sponsored a study of a "co-operative network of time-sharing computers." A testbed network was begun by connecting a TX-2 mini-computer at MIT via phone line to a computer at System Development Corporation in Santa Monica, California.

BITNET

In 1981, a network called BITNET ("Because It's Time Network") began to link academic institutions worldwide.

CSNET

The Computer Science Network was a computer network that began operation in 1981 in the United States. Its purpose was to extend networking benefits, for computer science departments at academic and research institutions that could not be directly connected to ARPANET (which was the predecessor to the Internet), due to funding or authorization limitations. It played a significant role in spreading awareness of, and access to, national networking and was a major milestone on the path to development of the global Internet. CSNET was funded by the National Science Foundation for an initial three-year period from 1981 to 1984.

NSFNET

In 1986, the National Science Foundation funded, which provides high-speed Internet connections to link universities and research institutions.

HTTP

A protocol used to request and transmit files, especially webpages and webpage components, over the Internet or other computer network.

Internet service provider (ISP)

An Internet service provider is any organization that provides access to the Internet. While nonprofit organizations such as universities and government agencies can be considered to be ISPs, the term is generally applied to a commercial, fee-based service. Typically, a user is given an account that is accessed by logging in through the operating system's Internet connection facility by supplying a user ID and password. Once connected, the user can run Web browsers, e-mail clients, and other programs that are designed to work with an Internet connection. Most personal ISP accounts include a small allotment of server space that users can use to host their personal Web pages. There are generally extra charges for larger allotments of space, for sites that generate high traffic, and for commercial sites. Business-oriented ISPs typically provide a more generous starting allotment along with more extensive technical support and more reliable and higher-capacity servers that are managed 24 hours a day. People entering the business today strive to provide added-value services such as superior Web page hosting facilities, hosting blogs or wikis, or to focus on specialized services for particularly industries (such as real estate). Today's ISPs also face a variety of legal challenges, including customer privacy vs. the war on terrorism, responsibility for copyright infringement, and possible liability for online defamation, *harassment*, or worse.

HTML

Several markup languages have been devised for specifying the organization or format of documents. Today the most commonly known markup language is the Hypertext Markup Language (see html, dhtml, and xhtml), which is the organizational "glue" of the Web (see World Wide Web). HTML is primarily concerned with rendering (displaying) documents. It describes structural features of documents (such as headers, sections, tables, and frames), but it does not really convey the structure of the information within the document. Further, HTML is not extensible — that is, one can't define one's own tags and use them as part of the language. XM L, or Extensible Markup Language, is designed to meet both of these needs. In effect, while HTML is a descriptive coding scheme, XM L is a scheme for creating data definitions and manipulating data within documents. (XML can be viewed as a subset of the powerful and generalized SGML, or Standard Generalized Markup Language.)

URL

An Internet address (for example, *http://www.hmco.com/trade/*), usually consisting of the access protocol (*http*), the domain name (*www.hmco.com*), and optionally the path to a file or resource residing on that server (*trade*).

USENET

In 1979, Unix users Tom Truscott, Jim Ellis, and Steve Bellovin developed a program to exchange news in the form of files copied between the Duke University and University of North Carolina computer systems. This gradually grew into USENET (or netnews), providing thousands of topical newsgroups.

7. Translate the following sentences into Russian.

1. Instant messaging (IM) is a type of online chat, which offers real-time text transmission over the Internet.

2. Thus, XM L documents can be properly rendered by browsers, while applications that are XM L-enabled (or that use XM L-aware ActiveX controls or similar Java facilities, for example) can parse the XM L and identify the data structures and elements in the document.

3. Business-oriented ISPs typically provide more generous starting allotment along with more extensive technical support and more reliable and higher-capacity servers that are managed 24 hours a day.

4. The tort of cyber defamation is considered to be the act of defaming, insulting, offending or otherwise causing harm through false statements.

5. MoteLab provided a public, long-lived testbed for development and testing of sensornetwork applications via an intuitive web-based interface.

6. The ever-increasing body of medical and scientific information, which becomes available through professional publications each year, makes it virtually impossible for the average health professional to keep up with the latest knowledge or developments.

8. Translate the following sentences into English.

1. Как правило, существуют дополнительные наценки за больший объем выделяемого пространства, за сайты, генерирующие высокий сетевой трафик и коммерческие сайты.

2. Типичный полигон может включать в себя программное и аппаратное обеспечение, а также сетевые компоненты.

3. Нарушение авторских прав происходит, когда произведение, охраняемое авторским правом, воспроизведено, распространено, представлено перед аудиторией или переработано без разрешения правообладателя.

4. Существуют два основных типа клеветы: пасквиль или письменная клевета, злословие или устные клеветнические измышления.

5. Веб-контент это текстовой, визуальный или звуковой ресурс (содержимое) с которым встречаются пользователи во время работы с сайтом.

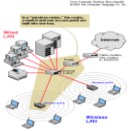
SPEAKING SECTION

9. Answer the following questions and think over five more questions, which you could ask your partner about.

1. Is there any fundamental difference between the Internet and the World Wide Web?

2. Why it's so easy for us to link them together in our minds?

3. What are the advantages and disadvantages of our ever-increasing use of computer technology?



10. Prepare a presentation on the topic being discussed.

WRITING

PRACTICE (1)

A. Writing models: Formal letters

! There are many possible formats for different types of essays, as well as non-academic texts such as letters and CVs. If a selection of formal letters, for example, is studied, it will be seen that different styles of heading and layout are used by different organizations. However, the following models are provided so that you may use the outlines confident that they will be acceptable in almost all situations.

Comparison and argument are common components of essay titles, and the models given here show one way of answering the questions. However, it must be remembered that argument may be only one part of the question, so that both comparison and discussion (plus other elements) could well be needed in the same essay.

Faculties and departments may well give new guidance about what is required in terms of style and layout. Above all, you need to examine a variety of styles of letters, CVs and essays, and to develop a suitable style of your own by synthesizing the most appropriate features.

a) Arts & Social Sciences Admissions Office

Wye House

Central Campus

University of Dorchester

Dorchester BR3 5HT

United Kingdom

b) Ms P Tan

54 Sydney Road

Rowborough RB1 6FD

c) Ref: MB/373

d) 3 May 2002

e) Dear Ms Tan

f) Application for MA International Studies

g) Further to your recent application, I would like to invite you to the university for an informal interview on Tuesday 21st May at 11 am. You will be able to meet the course supervisor, Dr Schmidt, and look round the department.

h) A map of the campus and instructions for finding the university are enclosed.

i) Please let me know if you will be able to attend on the date given.

j) Yours sincerely

k) M. Bramble

l) Mick Bramble
Administrative Assistant
Arts & Social Sciences

Enc.

11. Label the following features of formal letters with the letters (a–l) from the left margin above.

(…) Date
(…) Greeting
(…) Further details
(…) Subject headline
(…) Ending
(…) Address of recipient
(…) Reason for writing
(…) Signature
(…) Request for response
(…) Address of sender
(…) Sender's reference
(…) Writer's name and title

NB!

a) When writing to somebody whose name you do not know, e.g. The Manager, use *Dear Sir* and *Yours faithfully*.

b) A formal letter generally uses the family name in the greeting (*Dear Ms Tan*). Certain organisations may, however, use a first name with a family name or even a first name alone (*Dear Jane Tan/Dear Jane*).

c) If the sender includes a reference it is helpful to quote it in your reply.

12. Write a reply to Mr Bramble making the following points:

a) You will attend the interview on the date given.

b) You would like to have the interview one hour later, owing to train times.

13. Study the following newspaper advert. You have decided to apply for this job. Make notes for your letter of application, and then write the letter, paying attention to layout as well as content.

STAFF REQUIRED FOR RECEPTION WORK AT CITY HOTEL

We are looking for enthusiastic and helpful receptionists (m/f) to join our team. Candidates should be well-presented and able to speak at least two languages. Hotel experience not necessary as training will be given. Ability to get on with people and work in a team more important. Some evening and weekend work. Good conditions and rates of pay. Apply in writing with CV and covering letter to: The Manager, Hotel Nelson, Queens Road, Rowborough RB2 4RN quoting Ref. EN2.

REFERENCES, USEFUL LINKS AND FURTHER READING

1. Academic Writing', by Malashenko, Elena, 2007.

2. Berkowitz, Howard C. Building Service Provider Networks. New York: Wiley, 2002

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

4. www.cnn.com

5. www.sri.com

6. www.computerhistory.org

7. www.ispconsumerguide.com

UNIT 9

E-MAIL AND ITS TRANSFER PROTOCOLS

VOCABULARY

1. Match the words with their definitions:

1) bells and whistles (inf.)	[belznd'(h)wıslz]	a) to indicate
2) append to [v. with obj.]	[ə'pend]	b) resolve (a sentence) into its component parts and describe their syntactic roles
3) breakthrough (n.)	['breıkθruː]	c) Syn. add (something) to the end of a written document
4) designate (v.)	['dezıgneıt]	d) an important discovery or development
5) parse (n.)	['pɑːz]	e) attractive additional features or trimmings

Before you read

2. Discuss with your partner the following questions.

- **What do you know about e-mail?**
- **What are the reasons for sending written messages electronically?**

3. Skim the text to check your ideas.

READING

E-MAIL PROTOCOLS

by Marshall Brain and Tim Crosby

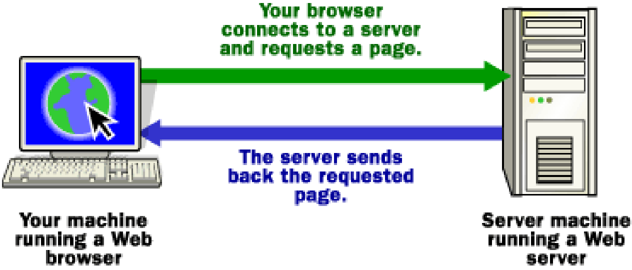
E-mail has become an extremely popular communication tool. According to Darwin Magazine: Prime Movers, the first e-mail message was sent in 1971 by an engineer named Ray Tomlinson. Prior to this, you could only send messages to users on a single machine. Tomlinson's *breakthrough* was the ability to send messages to other machines on the Internet, using the @ sign to *designate* the receiving machine.

An e-mail message has always been nothing more than a simple text message -- a piece of text sent to a recipient. In the beginning and even today, e-mail messages tend to be short pieces of text, although the ability to add attachments now makes many messages quite long.

You've probably already received several e-mail messages today. To look at them, you use some sort of e-mail client. Many people use well-known, stand-alone clients like Microsoft Outlook, Outlook Express, **Eudora** or Pegasus. People who subscribe to free e-mail services like Hotmail or Yahoo use an e-mail client that appears in a Web page. If you're an AOL customer, you use AOL's e-mail reader. No matter which type of client you're using, it generally does four things:

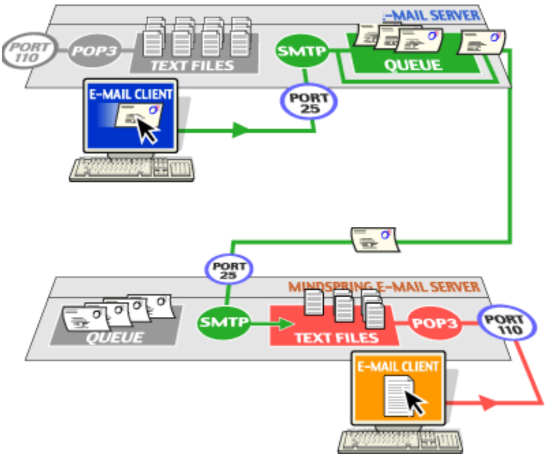
- Shows you a list of all of the messages in your mailbox by displaying the message header. The header shows you who sent the mail, the subject of the mail and may also show the time and date of the message and the message size.
- Lets you select a message header and read the body of the e-mail message.
- Lets you create new messages and send them. You type in the e-mail address of the recipient and the subject for the message, and then type the body of the message.
- Lets you add attachments to messages you send and save the attachments from messages you receive.

Sophisticated e-mail clients may have all sorts of *bells and whistles*, but at the core, this is all that an e-mail client does.


Your browser connects to a server and requests a page.
The server sends back the requested page.
Your machine running a Web browser
Server machine running a Web server

Machines on the Internet can run software applications that act as servers. There are Web servers, FTP servers, telnet servers and e-mail servers running on millions of machines on the Internet right now. These applications run all the time on the server machine and they listen to specific ports, waiting for people or programs to attach to the port. The simplest possible e-mail server would have a list of e-mail accounts (e.g. John Smith's might be jsmith), a text file for each account in the list (e.g. the server would have a text file in its directory named JSMITH.TXT). The server would format those pieces of information and *append* them to the bottom of the JSMITH.TXT file.

There are several other pieces of information that the server might save into the file, like the time and date of receipt and a subject line; but overall, you can see that this is an extremely simple process.



As other people sent mail to jsmith, the server would simply *append* those messages to the bottom of the file in the order that they arrived. The text file would accumulate a series of five or 10 messages, and eventually you would log in to read them. When you wanted to look at your e-mail, your e-mail client would connect to the server machine. In the simplest possible system, it would: ask the server to send a copy of the JSMITH.TXT file, ask the server to erase and reset the JSMITH.TXT file. Then save the JSMITH.TXT file on your local machine. *Parse* the file into the separate messages (using the word "From:" as the separator). Show you all of the message headers in a list. When you double-clicked on a message header, it would find that message in the text file and show you its body.

As you can see, this is a very simple system. Surprisingly, the real e-mail system that you use every day isn't much more complicated than this.

For the vast majority of people right now, the real e-mail system consists of two different servers running on a server machine. One is called the **SMTP server**. The SMTP server handles outgoing mail. The other is either a **POP3 server** or an **IMAP server**, both of which handle incoming mail.

Whenever you send a piece of e-mail, your e-mail client interacts with the SMTP server to handle the sending. The SMTP server on your host may have conversations with other SMTP servers to deliver the e-mail. Let's assume that you want to send a piece of e-mail. Your e-mail ID is brain, and you have your account on yahoo.com. You want to send e-mail to jsmith@mindspring.com. You are using a stand-alone e-mail client like Outlook Express.

When you set up your account at yahoo, you told Outlook Express the name of the mail server -- mail.yahoo.com. When you compose a message and press the Send button, here's what happens:

1. Outlook Express connects to the SMTP server at mail.yahoo.com using port 25.

2. Outlook Express has a conversation with the SMTP server, telling the SMTP server the address of the sender and the address of the recipient, as well as the body of the message.

3. The SMTP server takes the "to" address (jsmith@mindspring.com) and breaks it into two parts: the recipient name (jsmith) and the domain name (mindspring.com). If the "to" address had been another user at yahoo.com, the SMTP server would simply hand the message to the POP3 server for yahoo.com (using

a little program called the delivery agent). Since the recipient is at another domain, SMTP needs to communicate with that domain.

4. The SMTP server has a conversation with a DNS. It says, "Can you give me the IP address of the SMTP server for mindspring.com?" The DNS replies with the one or more IP addresses for the SMTP server(s) that Mindspring operates.

5. The SMTP server at yahoo.com connects with the SMTP server at Mindspring using port 25. It has the same simple text conversation that your e-mail client had with the SMTP server for Yahoo, and gives the message to the Mindspring server. The Mindspring server recognizes that the domain name for jsmith is at Mindspring, so it hands the message to Mindspring's POP3 server, which puts the message in jsmith's mailbox.

If, for some reason, the SMTP server at Yahoo cannot connect with the SMTP server at Mindspring, then the message goes into a queue. The SMTP server on most machines uses a program called sendmail to do the actual sending, so this queue is called the sendmail queue. Sendmail will periodically try to resend the messages in its queue. For example, it might retry every 15 minutes. After four hours, it will usually send you a piece of mail that tells you there is some sort of problem. After five days, most sendmail configurations give up and return the mail to you undelivered.

The SMTP server understands very simple text commands like HELO, MAIL, RCPT and DATA.

In the simplest implementations of POP3, the server really does maintain a collection of text files -- one for each e-mail account. When a message arrives, the POP3 server simply *appends* it to the bottom of the recipient's file. When you check your e-mail, your e-mail client connects to the POP3 server using **port 110**. The POP3 server requires an account name and a password. Once you've logged in, the POP3 server opens your text file and allows you to access it. Like the SMTP server, the POP3 server understands a very simple set of text commands like USER, PASS and etc.

Your e-mail client connects to the POP3 server and issues a series of commands to bring copies of your e-mail messages to your local machine.

You can see that the POP3 server simply acts as an interface between the e-mail client and the text file containing your messages. And again, you can see that the POP3 server is extremely simple. You can connect to it through telnet at port 110 and issue the commands yourself if you would like to. As you can see, the POP3 protocol is very simple. It allows you to have a collection of messages stored in a text file on the server. Your e-mail client (e.g. Outlook Express) can connect to your POP3 e-mail server and download the messages from the POP3 text file onto your PC. Many users want to do far more than that with their e-mail, and they want their e-mail to remain on the server. The main reason for keeping your e-mail on the server is to allow users to connect from a variety of machines. With POP3, once you download your e-mail it's stuck on the machine to which you downloaded it. If you want to read your e-mail both on your desktop machine and your laptop (depending on whether you're working in the office or on the road), POP3 makes life difficult.

IMAP is a more advanced protocol that solves these problems. With IMAP, your mail stays on the e-mail server. You can organize your mail into folders, and all the folders live on the server as well. When you search your e-mail, the search occurs on the server machine, rather than on your machine. This approach makes it extremely easy for you to access your e-mail from any machine, and regardless of which machine you use, you have access to all of your mail in all of your folders.

Your e-mail client connects to the IMAP server using **port 143**. The e-mail client then issues a set of text commands that allow it to do things like list all the folders on the server, list all the message headers in a folder, get a specific e-mail message from the server, delete messages on the server or search through all of the e-mails on the server.

Most e-mail clients have some way to cache e-mail on their local machine. For example, the client will download all the messages and store their complete contents on the local machine. The messages still exist on the IMAP server, but you now have copies on your machine. This allows you to read and reply to e-mail even if you have no connection to the Internet. The next time you establish a connection, you download all the new messages you received while disconnected and send all the mail that you wrote while disconnected. Your e-mail client allows you to add attachments to e-mail messages you send, and also lets you save attachments from messages that you receive. Attachments might include word processing documents, spreadsheets, sound files, snapshots and pieces of software. Usually, an attachment is not text. Since e-mail messages can contain only text information, and attachments aren't text, there's a problem that needs to be solved. In the early days of e-mail, you solved this problem by hand, using a program called uuencode.The uuencode program assumes that the file contains binary information. It extracts 3 bytes from the binary file and converts them to four text characters. What uuencode produces, therefore, is an encoded version of the original binary file that contains only text characters. In the early days of e-mail, you would run uuencode yourself and paste the uuencoded file into your e-mail message.

Considering its tremendous impact on society, having forever changed the way we communicate, today's e-mail system is one of the simplest things ever devised. There are parts of the system, like the routing rules in sendmail, that get complicated, but the basic system is incredibly straightforward.

LANGUAGE DEVELOPMENT

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

1) bells and whistles (informal)	a)
2) e-mail client	a)
3) sendmail queue	b)
4) uuencode	c)

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

1) заголовок сообщения	a)
2)текстовое сообщение	b)
2) автономный, отдельный, не входящий в систему	c)
3) служба доставки сообщений	d)
4) агент передачи (электронной) почты	e)

6. Find and learn English and Russian definitions for the following commands.

1) VRFY	a)
2) EXPN	b)
3) VERB	c)
4) RETR	d)
5) DELE	e)
6) TOP	f)
7) RCPT TO:	g)
8) EHLO	h)

7. Find and learn English and Russian definitions for the following abbreviations:

1) AOL	a)
2) FTP	b)
3) SMTP	c)
4) IMAP	d)
5) POP	e)
6) DNS	f)
7) TLS	g)
8) SSL	h)

KEY CONCEPTS

Eudora

/juːˈdɔərə/ is an email client used on the Apple Macintosh and Microsoft Windows operating systems. It also supports several palmtop computing platforms, including Newton and the Palm OS. The software was named after American author Eudora Welty, because of her short story *Why I Live at the P.O.*[2][3] Eudora was developed in 1988 by Steve Dorner, who worked at the Computer Services Organization of the University of Illinois at Urbana-Champaign.[4] Eudora was acquired by Qualcomm in 1991. In 2006 Qualcomm stopped development of the commercial version, and sponsored the creation of a new open-source version based on Mozilla Thunderbird, code-named Penelope.

POP

is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection.[1] POP and IMAP (Internet Message Access Protocol) are the two most prevalent Internet standard protocols for e-mail retrieval.[2] Virtually all modern e-mail clients and servers support both. POP has been developed through several versions, with version 3 (POP3) being the current standard. Most webmail service providers such as Gmail and Yahoo! Mail provide both an IMAP and POP3 service.

POP supports simple download-and-delete requirements for access to remote mailboxes (termed maildrop in the POP RFC's).[3] Although most POP clients have an option to leave mail on server after download, e-mail clients using POP generally connect, retrieve all messages, store them on the user's PC as new messages, delete them from the server, and then disconnect. Other protocols, notably IMAP, provide more complete and complex remote access to typical mailbox operations. Many e-mail clients support POP as well as IMAP to retrieve messages; however, fewer Internet Service Providers (ISPs) support IMAP. A POP3 server listens on well-known port 110. Encrypted communication for POP3 is either requested after protocol initiation, using the STLS command, if supported, or by POP3S, which connects to the server using Transport Layer Security (TLS) or Secure Sockets Layer (SSL) on well-known TCP port 995.

Available messages to the client are fixed when a POP session opens the maildrop, and are identified by message-number local to that session or, optionally, by a unique identifier assigned to the message by the POP server. This unique identifier is permanent and unique to the maildrop and allows a client to access the same message in different POP sessions. Mail is retrieved and marked for deletion by message-number. When the client exits the session, the mail marked for deletion is removed from the maildrop.

Simple Mail Transfer Protocol (SMTP)
is an Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks. SMTP was first defined by RFC 821 (1982, eventually declared STD 10),[1] and last updated by RFC 5321 (2008)[2] which includes the Extended SMTP (ESMTP) additions, and is the protocol in widespread use today. SMTP uses TCP port 25. The protocol for new submissions (MSA) is effectively the same as SMTP, but it uses port 587 instead. SMTP connections secured by SSL are known by the shorthand SMTPS, though SMTPS is not a protocol in its own right. While electronic mail servers and other mail transfer agents use SMTP to send and receive mail messages, user-level client mail applications typically use SMTP only for sending messages to a mail server for relaying. For receiving messages, client applications usually use either the POP or the IMAP or a proprietary system (such as Microsoft Exchange or Lotus Notes/Domino) to access their mail box accounts on a mail server.

8. Translate the following sentences into Russian.

1. Headers contain tracking information for an individual email, detailing the path a message took as it crossed mail servers.

2. The uuencode command takes the named SourceFile (default standard input) and produces an encoded version on the standard output.

3. A class and package is provided which allows TeX pictures or other TeX code to be compiled standalone or as part of a main document.

4. To check what is currently sitting in the sendmail mail queue use the sendmail -bp command.

5. Internet Message Access Protocol (IMAP) is a protocol for e-mail retrieval and storage.

9. Translate the following sentences into English.

1) Агент доставки электронной почты) — программа, принимающая входящие электронные письма и доставляющая их на электронный ящик получателя.

2) Уровень защищённых сокетов — криптографический протокол, который обеспечивает безопасность связи.

3) Система доменных имён — иерархическая компьютерная распределённая система для получения информации о доменах.

SPEAKING SECTION

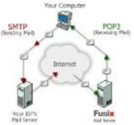
10. Answer the following questions and think over five more questions, which you could ask your partner about.

1) How e-mail gets from your computer to a friend halfway around the world?

2) What is a POP3 server, and how does it hold your mail?

3) What the simplest possible e-mail server would look like in order to get a basic understanding of the process?

4) If all of your e-mail is stored on the server, then how can you read your mail if you are not connected to the Internet?

11. 

Prepare a presentation on the topic being discussed.

WRITING

PRACTICE (2)

A. Writing models: Designing and Reporting Surveys

! Surveys, in which people are asked questions about their opinions or behaviour, are a common feature of academic work, especially in fields such as education, psychology and social sciences.

12. What are the reasons for carrying out surveys? List your ideas below.

a)

b)

c)

13. Study the report of a survey carried out on a university campus. Complete the report by inserting suitable words from the box below into the gaps.

sample	conducted	slightly	respondents	random
questions	majority	questioned	mentioned	interviewees
common	questionnaire	generally	minority	

Introduction

A survey was a) ________ to find out how part-time work affects student life and study. The research was done by asking students selected at b) _______ on the campus to complete a c) _______ (see Appendix 1). 50 students were d) _______ on Saturday April 23rd, with approximately equal numbers of male and female students.

Findings

Of the e) _______, 30% currently had part-time jobs, 20% had had part-time jobs, but half had never done any work during university semesters (see Table 1). f) ______ who were working or who had worked were next asked about the reasons for taking the jobs. The most common reason was lack of money (56%), but many students said that they found the work useful experience (32%) and others g) _______ social benefits (12%).

Table 1. Do you have or have you had a part-time job?

	Men	Women	Total	%
have job now	8	7	15	30
had job before	4	6	10	20
never had job	14	11	25	50

The 25 students with work experience were next asked about the effects of the work on their studies. A significant h) ________ (64%) claimed that there were no negative effects at all. However, 24% said that their academic work suffered i) _______, while a small j) _______ (12%) reported serious adverse results, such as tiredness in lectures and falling marks.

Further k) ______ examined the nature of the work that the students did. The variety of jobs was surprising, from van driver to busker, but the most l) _______ areas were catering and bar work (44%) and secretarial (32%). Most students worked between 10 and 15 hours per week, though two (8%) worked over 25 hours. Rates of pay were m) _______ near the national minimum wage, and averaged J5.20 per hour.

The final question invited students to comment on their experience of part-time work. Many

(44%) made the point that students should be given larger grants so that they could concentrate on their studies full-time, but others felt that they gained something from the experience, such as meeting new people and getting insights into various work environments. One student said that she had met her current boyfriend while working in a city centre restaurant.

Conclusions

It is clear that part-time work is now a common aspect of student life. Many students find jobs at some point in their studies, but an overwhelming majority (88%) of those deny that it has a damaging effect on their studies. Most students work for only 2–3 hours per day on average, and a significant number claim some positive results from their employment.

Obviously, our survey was limited to a relatively small n) _______ by time constraints, and a fuller study might modify our findings in various ways.

14. Question 1 is given above Table 1. What were the other questions in this survey? Using the results above, write possible questions below.

1.
2.
3.
4.
5.

15. What is the main tense in (a) Introduction and Findings and (b) Conclusion? Explain the reason for the difference.

16. Questionnaire Design.

Which is the better question?

1) How old are you?
2) Are you (a) under 20, (b) between 21 and 30, (c) over 30?

17. What is the main difference between the two questions?

1) What do you think of university students?
2) Do you think university students are (a) lazy, (b) hardworking, (c) average

18. You are preparing a survey on one of the following subjects. Prepare a questionnaire of no more than ten questions to collect the most useful data.

a) How overseas students learn vocabulary
b) Student attitudes to the cinema
c) A comparison of undergraduate and post-graduate leisure activities.

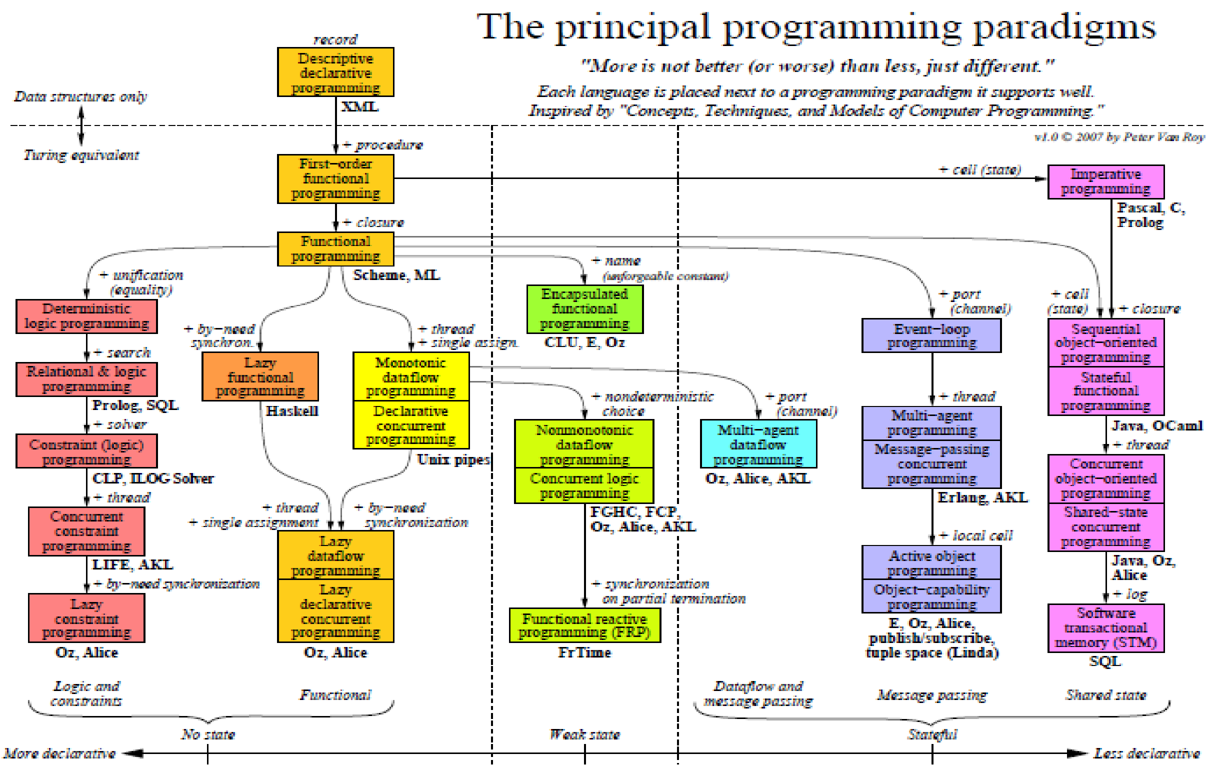
REFERENCES, USEFUL LINKS AND FURTHER READING

1. Academic Writing’, by Malashenko, Elena, 2007.

2. ‘Computer Desktop Encyclopedia copyright ©1981-2012 by The Computer Language Company Inc.

3. http://www.thefreedictionary.com/

Appendix I


The principal programming paradigms
"More is not better (or worse) than less, just different."
Each language is placed next to a programming paradigm it supports well.
Inspired by "Concepts, Techniques, and Models of Computer Programming."
v1.0 © 2007 by Peter Van Roy
Data structures only
Turing equivalent
record
Descriptive declarative programming
XML
+ procedure
First−order functional programming
+ cell (state)
Imperative programming
Pascal, C, Prolog
+ closure
Functional programming
Scheme, ML
+ unification (equality)
Deterministic logic programming
+ search
Relational & logic programming
Prolog, SQL
+ solver
Constraint (logic) programming
CLP, ILOG Solver
+ thread
Concurrent constraint programming
LIFE, AKL
+ by−need synchronization
Lazy constraint programming
Oz, Alice
+ by−need synchron.
Lazy functional programming
Haskell
+ thread
+ single assign.
Monotonic dataflow programming
Declarative concurrent programming
Unix pipes
+ thread
+ single assignment
+ by−need synchronization
Lazy dataflow programming
Lazy declarative concurrent programming
Oz, Alice
+ name (unforgeable constant)
Encapsulated functional programming
CLU, E, Oz
+ nondeterministic choice
Nonmonotonic dataflow programming
Concurrent logic programming
FGHC, FCP, Oz, Alice, AKL
+ synchronization on partial termination
Functional reactive programming (FRP)
FrTime
+ port (channel)
Multi−agent dataflow programming
Oz, Alice, AKL
+ port (channel)
Event−loop programming
+ thread
Multi−agent programming
Message−passing concurrent programming
Erlang, AKL
+ local cell
Active object programming
Object−capability programming
E, Oz, Alice, publish/subscribe, tuple space (Linda)
+ cell (state)
+ closure
Sequential object−oriented programming
Stateful functional programming
Java, OCaml
+ thread
Concurrent object−oriented programming
Shared−state concurrent programming
Java, Oz, Alice
+ log
Software transactional memory (STM)
SQL
Logic and constraints
Functional
Dataflow and message passing
Message passing
Shared state
No state
Weak state
Stateful
More declarative
Less declarative

Adapted from http://www.info.ucl.ac.be/~pvr/paradigms.htm

Учебное издание

Малашенко Елена Александровна

ENGLISH FOR IT STUDENTS

Part II

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Редактор
Корректор
Компьютерная верстка *М. Ю. Мошкова*

Подписано в печать 21.08.2012. Формат 60×90 $^{1}/_{16}$.
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
http://www.iseu.by