TOPIC 1. FUNDAMENTALS OF COMPUTER SCIENCE

 Active vocabulary engineering activities — конструкторська робота theoretical activities — теоретична діяльність design — розробка hardware — апаратне забезпечення software — програмне забезпечення (ПЗ) encompass — охоплювати performance studies — дослідження експлуатаційних характеристик technique — метод queueing theory — теорія черг estimation — оцінка reliability — надійність availability — експлуатаційна готовність, працездатність complicated — складний experimentation — експериментування, проведення експериментів incorporate — включати (до свого складу) development cycle — цикл розробки overlap — перекривати instruction — команда carry out — виконувати software engineering — програмотехніка, інженерія розроблення ПЗ artificial intelligence — штучний інтелект date to — датуватися numerical analysis — числовий аналіз digital computer — цифровий комп’ютер electrical engineering — електротехніка primarily — передусім basics — основи circuit — схема (електрична, електронна) input — 1) n вхід; 2) n введення; 3) v вводити output — 1) n вихід; 2) n виведення; 3) v виводити arbitrary — довільний 6 storage — зберігання represent — представляти device — пристрій binary digit — двійкова цифра transmission — передача notational system — система позначень yield — давати calculus — числення interpretable symbol — інтерпретовний символ employ — застосовувати, використовувати gate — вентиль, логічний елемент needed — необхідний, потрібний extension — розширення, доповнення milestone — віха specification — тут деталізація, уточнення infinite — нескінченний proof — випробування breakthrough — прорив access — доступ execution — виконання critical — важливий

 BASICS OF COMPUTER SCIENCE The field of computer science includes engineering activities such as the design of computers and of the hardware and software that make up computer systems. It also encompasses theoretical, mathematical activities, such as the design and analysis of algorithms, performance studies of systems and their components by means of techniques like queueing theory, and the estimation of the reliability and availability of systems by probabilistic techniques. Since computer systems are often too large and complicated to allow a designer to predict failure or success without testing, experimentation is incorporated into the development cycle. Computer science is generally considered a discipline separate from computer engineering, although the two disciplines overlap extensively in the area of computer architecture, which is the design and study of computer systems. The major subdisciplines of computer science have traditionally been: 1) architecture (including all levels of hardware design, as well as the integration of hardware and software components to form computer systems); 2) software (the programs, or sets of instructions, that tell a computer how to carry out tasks), here subdivided into software engineering, programming languages, operating systems, information systems and data bases, artificial intelligence, and computer graphics; 3) theory, which includes computational methods and numerical analysis on the one hand and data structures and algorithms on the other. Computer science as an independent discipline dates to only about 1960, although the electronic digital computer that is the object of its study was invented some two decades earlier. The roots of computer science lie primarily in the related fields of electrical engineering and mathematics. Electrical engineering provides the basics of circuit design — namely, the idea that electrical impulses input to a circuit can be combined to produce arbitrary outputs. The invention of the transistor and the miniaturization of circuits, along with the invention of electronic, magnetic, and optical media for the storage of information, resulted from advances in electrical engineering and physics. Mathematics is the source of one of the key concepts in the development of the computer — the idea that all information can be represented as sequences of zeros and ones. In the binary number system numbers are represented by a sequence of the binary digits 0 and 1 in the same way that numbers in the familiar decimal system are represented using the digits 0 through 9. The relative ease with which two states (e.g., high and low voltage) can be realized in electrical and electronic devices led naturally to the binary digit, or bit, becoming the basic unit of data storage and transmission in a computer system. Boolean algebra, a notational system developed in the 19th century by an English mathematician George Boole, supplied formalism for designing a circuit with binary input values of 0s and 1s (false or true, respectively, in the terminology of logic) to yield any desired 8 combination of 0s and 1s as output. One of the primary requirements when dealing with digital circuits is to find ways to make them as simple as possible. This constantly requires that complex logical expressions be reduced to simpler expressions that nevertheless produce the same result. Boolean algebra permits an algebraic manipulation of logical statements that can demonstrate whether or not a statement is true and show how a complicated statement can be rephrased in a simpler, more convenient form without changing its meaning. George Boole believed in what he called the ‘process of analysis’, that is, the process by which combinations of interpretable symbols are obtained. It is the use of these symbols according to well-determined methods of combination that he believed presented ‘true calculus’. Today, all our components employ Boole’s logic system — using microchips that contain thousands of tiny electronic switches arranged into logical gates that produce predictable and reliable conclusions. A gate is an electronic circuit such that its output is fully determined by the state of its inputs. Theoretical work on computability, which began in the 1930s, provided the needed extension to the design of whole machines. A milestone was the 1936 specification of the conceptual Turing machine (a theoretical device that manipulates an infinite string of 0s and 1s) by the British mathematician Alan Turing and his proof of the model’s computational power. Another breakthrough was the concept of the stored-program computer, usually credited to the Hungarian-American mathematician John von Neumann. This idea — that instructions as well as data should be stored in the computer’s memory for fast access and execution — was critical to the development of the modern computer. Previous thinking was limited to the calculator approach, in which instructions are entered one at a time. Exercise 1. Find in text the English for: зберігання інформації; база даних; обчислювальні методи; один за одним; мова програмування; споріднені області; цифри від 0 до 9; електротехніка; процес (цикл) розробки; комп’ютерна графіка; приписувати що-небудь кому-небудь (вважати що-небудь чиєюсь заслугою); дослідження експлуатаційних характеристик систем; теорія черг; відносна простота; булева алгебра; структури даних; забезпечити формалізацію; успіхи в розвитку електротехніки; обчислювальна техніка; передбачити відмову; магнітні та оптичні носії; значною мірою перекриватися; так само, як; схемотехніка; штучний інтелект; з одного боку, ... з іншого боку; методи теорії ймовірностей; числовий аналіз; поряд із; бути (стати) наслідком чогонебудь; двійкова система числення; операційна система; система позначень; інженерія розроблення програмного забезпечення.