Unfortunately for us, computers can’t understand spoken English or other natural language. The only language they can understand directly[прямо, безпосередньо, відразу] is machine code, which consists of 1s and 0s (binary code).

Machine code is too difficult to write. For this reason we use symbolic languages to communicate instructions to the computer. For example, assembly languages use abbreviations such as ADD, SUB, MPY to represent instruction. The program is then translated into machine code by a piece of software called an assembler. Machine code and assembly languages are called low=level languages becous thea are closer to the hardware. They are quite complex and restricted[вузький, обмежений] to particular[часткове, деталь] machines. To make the program easy to write, and overcame the problem of intercommunication between different types of computer, software developers designed high-level languages, which a closer to the English language. Here some examples:

Fortran was developed by IBM in 1956 and is still used for scientific and end engineering applications.

COBOL was developed in 1959 and it’s mainly used for business applications.

BASIC was developed in 1960s and was widely used in microcomputer programing because it was easy to learn. Visual Basic is modern version of the old Basic language, used to build graphical elements such as buttons and windows in Windows program.

PASCAL was created in 1971. It is used in universities to teach the fundamentals of programing.

C was developed in the 1980s at AT&T. It used to write system software, graphics and commercial applications. C++ is a version of C, which incorporates object-oriented programming: the programmer concentrates on particular things (a piece of text, a graphics, or a table) and gives each object functions which can be altered without changing[зміна, заміна] the entire [повний, цілісний, весь] to modify;

Java was designed by Sun in 1995 to run on the Web. Java applets animation an interactive features on web pages.

Programs written in high-level languages must be translated into machine code by a compiler or an interpreter. A compiler translates the source code into object code – that is, it convert the entire program into machine code in one go. On the other hand, an interpreter translates the source code line by line as the program running.

It is important not to confuse programming language with markup languages, used to create web documents. Markup languages use instructions, know as markup tags to format and link test files. Some examples include:

HTML, which allows us to describe how information will be displayed on web pages.

XML, which stands for EXtensible Markup Language. While HTML uses pre-defined tags, XML enables us to define our own tags; it is no limited by a fixed set of tags.

Voice XML, which makes web contend accessible via voice and phone.