However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. Programming languages are essential for software development. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Programmable devices have existed for centuries. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL). Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Code-breaking algorithms have also existed for centuries. Following a consistent programming style often helps readability. Programming languages are essential for software development. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. Allen Downey, in his book How To Think Like A Computer Scientist, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.