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. 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. Following a consistent programming style often helps readability. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. 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. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. 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. Programming languages are essential for software development. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. Use of a static code analysis tool can help detect some possible problems. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Unreadable code often leads to bugs, inefficiencies, and duplicated code. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. Ideally, the programming language best suited for the task at hand will be selected. Also, specific user environment and usage history can make it difficult to reproduce the problem. Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly. Computer programmers are those who write computer software.