Assembly languages were soon developed that let the programmer specify instruction in a text format (e..g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
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.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
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 academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems.  
There exist a lot of different approaches for each of those tasks.  
 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.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Programming languages are essential for software development.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
 Different programming languages support different styles of programming (called programming paradigms).  
Integrated development environments (IDEs) aim to integrate all such help.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.