It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Techniques like Code refactoring can enhance readability. 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. It is very difficult to determine what are the most popular modern programming languages. One approach popular for requirements analysis is Use Case analysis. 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. One approach popular for requirements analysis is Use Case analysis. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Use of a static code analysis tool can help detect some possible problems. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. 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. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. 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. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. 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. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.