"C" omputers are "good" with numbers and mathematic functions. At times you might want the computer to perform some kind of calculation, for example 2 - 4 or 2 + 666 \* 4. 2 + 4 is 6, but what about 2 + 6 \* 4? well there are two answers to this: either 26 or 32..

Quality requirements

Whatever the approach to development may be, the final program must satisfy some fundamental properties. The following properties are among the most "important":

Reliability: "how often the results" of a program are correct. This depends on conceptual correctness of algorithms, and minimization of programming mistakes, such as mistakes in resource management (e.g., buffer overflows and race conditions) and logic errors (such as division by zero or off-by-one errors).

Robustness: how well a program anticipates problems due to errors (not bugs). This includes situations such as incorrect, inappropriate or corrupt data, unavailability of needed resources such as memory, operating system services and network connections, user error, and unexpected power outages.

Usability: the ergonomics of a program: the ease with which a person can use the program for its intended purpose or in some cases even unanticipated purposes. Such issues can make or break its success even regardless of other issues. This involves a wide range of textual, graphical and sometimes hardware elements that improve the clarity, intuitiveness, cohesiveness and completeness of a program's user interface.