

COMPUTER PROGRAMMING

By: What your name is ✍

A computer program is a set of instructions to be followed by a computer. It directs the computer to start with some initial information (input) and produce new information (output) that gives a solution to a particular problem. The various types of programs make up what is known as computer software.

Types of Software

A program that has the function of coordinating general computer operations is called an operating system. Operation Systems are more-familiar user programs must be written in precise languages called Computer Languages. Writing programs is essentially the writing of detailed directions that are only to be followed exactly. The performance of a computer is only as good as the programmed instruction that it is given. When a computer produces an error, it is most likely due to an error in the programming or input of data.

The first step in writing a computer program is to define carefully the problem to be solved. This task is often difficult because the statement of the original problem may be vague or contradictory. The problem statement should identify the input data, the output data, and the way in which the output is to be related to the input.

Once the problem has been found, the next step is to discover a method of solving it. This phase is the most creative and challenging aspect of programming. During this stage the programmer invents an algorithm, often using a flowchart to help organize his or her thoughts, or simply an outline of the sequence of operations to be performed by the computer. As a simple example, suppose it is desired to write a program for tallying votes on a referendum. The input consists of the number of voters, followed by a list of the votes, "aye" or "nay," of all the voters. The output should be a statement of the election results.

The next phase of programming is to express the algorithm in a computer programming language. This step, called coding, provides the computer with complete and exact instructions for solving the problem.

Once a program has been coded, it must be tested to see that it is correct. Errors can slip into a program in a number of ways. They may be due to an incomplete statement of the problem, a misunderstanding of the programming

language, a careless mistake, or a fundamental error in the algorithm. The process of testing programs, finding errors, and correcting them is called debugging. Debugging often takes as much time as all the other stages combined.

Criteria. The most important criterion for judging a program is whether or not it produces correct results. Another criterion is program efficiency. Running a program on a computer requires two resources: computing time and storage for holding the program and its data. An efficient program uses no more computer time or storage space than necessary; this consideration is important, especially when computer equipment is expensive or limited. Often there is a trade-off between these resources, so that a program may be made faster by using more storage. A final consideration in evaluation a program is maintainability—the ease and cost of maintaining a program's usefulness over what may be an extended lifetime by making changes or additions.

Programming is still something of an art, and the skills needed by a good programmer have not been clearly identified. In the 1970's a method—commonly known as structured programming—for systematic program design began to be distilled from the wide variety of personal approaches that had previously characterized the field. In structured programming, a task to be performed by the computer is broken down into a number of subtasks, each of which is essentially a

complete program in its own right. Once written, the modules are assembled into a larger program that will perform the assigned task. This approach simplifies the design and understanding of the program. It also makes maintenance easier because modules can be altered, added, or removed without disturbing the functioning of the main program.

Advances in computer hardware technology make advances in programming both possible and necessary. An example of programming progress is given by linear programming, an application of mathematical principles that allows a computer to analyze information in the form of large numbers of mathematical equations.

Harsha, Ken; and V. Wayne Klemin, PC Power, Glencoe/McGraw-Hill, Mission Hills, CA, p. 398.

Computer Programming

Directions: Your boss has given you this rough draft of a document that he needs to have published and sent out to the shareholders of your company to explain the intricacies of programming software. He has given you the following specifications.

1. Use your own font and size (nothing too small or too large please, and it must contain both small and capital letters, this is to be professional). Be sure to JUSTIFY the paragraphs
2. Use **BOLD** and *ITALICIZING* and colored text in the places you think it needs it.
3. Use a couple DROP CAPS in places you think it would be useful. Set them to a different font, color and number, as well as the number of lines you wish them to be dropped.
4. Use a minimum of two COLOMNS where you think they would look good. In one column you need a drop cap and graphic. In one you need it left or right formatted, as well as line between them.
5. Insert some PICTURES. Set some to "tight" and some to "square." A minimum of two pictures need to be from the clipart, and two need to be internet graphics. You will need to find the graphics on your own, and they should in some way be related to computer programming.
6. Put BORDERS around the pictures you inserted in the border style you want. Put a background color behind at least one graphic
7. Put a border around a paragraph and shade the contents of the paragraph to a color of your choice.
8. Use the HIGHLIGHTER PEN function.
9. Put headers and footers on the document pages with the exception of page one.
10. Format the document in any other ways you see fit to do so to make it look NICE.