BIT NOTES

Computer Programming

Karna Bahadur Shrestha

Problem solving using Computer

There are number of problems that is need computer to be solved. Let us assume the problems:

- Calculating simple interest.
- Calculate volume of cube.
- Calculate area of ellipse.

We can use computer to solve above problems efficiently. A computer consist of collection of hardware and software, they both are complementary to each other. Software is a set of programs written to solve particular problem. The programs of software may be written in different language those are available such as(C, C++, JAVA, Python) etc. On the basis of instruction given in the program, the computer gives output/result.

Before writing code, we must be clear about the processing steps to be performed by the computer. A program must be planned before coding in programming language in order to ensure that the program/instructions are appropriate to solve the problem and are in the correct sequence. Activities we have to plan in sequence to write effective code are shown below:

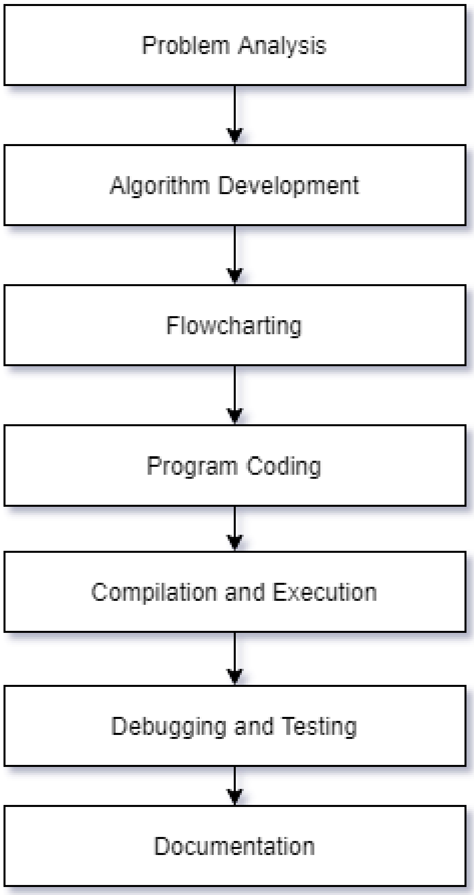


Figure 1Steps in problem solving using computer

Problem Analysis

Problem analysis is the process of defining a problem and decomposing overall system into smaller parts to identify possible inputs, processes and outputs associated with the problem.

Steps in problem analysis

1. **Specifying the Objective :**
   First, we need to know what problem is actually being solved. Making a clear statement of the problem depends upon the size and complexity of the problem. Smaller problems not involving multiple subsystems can easily be stated and then we can move onto the next step of "Program Design". However, a problem interacting with various subsystems and series of programs require complex analysis, in-depth research and careful coordination of people, procedures and programs.

2. **Specifying the Output :**
   Before identifying inputs required for the system, we need to identify what comes out of the system. The best way to specify output is to prepare some output forms and required format for displaying result. The best person to judge an output form is the end user of the system i.e. the one who uses the software to his benefit. Various forms can be designed by the programmer which must be examined to see whether they are useful or not.

3. **Specifying Input Requirements :**
   After having specified the outputs, the input and data required for the system need to be specified as well. One needs to identify the list of inputs required and the source of data. For example, in a simple program to keep student's record, the inputs could be the student's name, address, roll-numbers, etc. The sources could be the students themselves or the person supervising them.

4. **Specifying Processing Requirements :**
   When output and inputs are specified, we need to specify process that converts specified inputs into desired output. If the proposed program is to replace or supplement an existing one, a careful evaluation of the present processing procedures needs to be made, noting any improvements that could made. If the proposed system is not designed to replace an existing system, then it is well advised to carefully evaluate another system that addresses a similar problem.

Algorithm Development

An algorithm is an effective step-by-step procedure for solving a problem in a finite number of steps. In other words, it is a finite set of well-defined instructions or step-by-step description of the procedure written in human readable language for solving a given problem. An algorithm itself is division of a problem into small steps which are ordered in sequence and easily understandable. Algorithms are very important to the way computers process information, because a computer program is basically an algorithm that tells computer what specific tasks to perform in what specific order to accomplish a specific task. The same problem can be solved with different methods. So, for solving the same problem, different algorithms can be designed. In these algorithms, number of steps, time and efforts may vary more or less.

Characteristics of an Algorithm

An algorithm must possess following characteristics:

Finiteness: An algorithm should have finite number of steps and it should end after a finite time.

Input: An algorithm may have many inputs or no inputs at all.

Output: It should result at least one output.

Definiteness: Each step must be clear, well-defined and precise. There should be no any ambiguity.

Effectiveness: Each step must be simple and should take a finite amount of time.

Algorithm : Calculation of Simple Interest

```
Step 1: Start
Step 2: Read principle (P), time (T) and rate (R)
Step 3: Calculate I = P*T*R/100
Step 4: Print I as Interest
Step 5: Stop
```

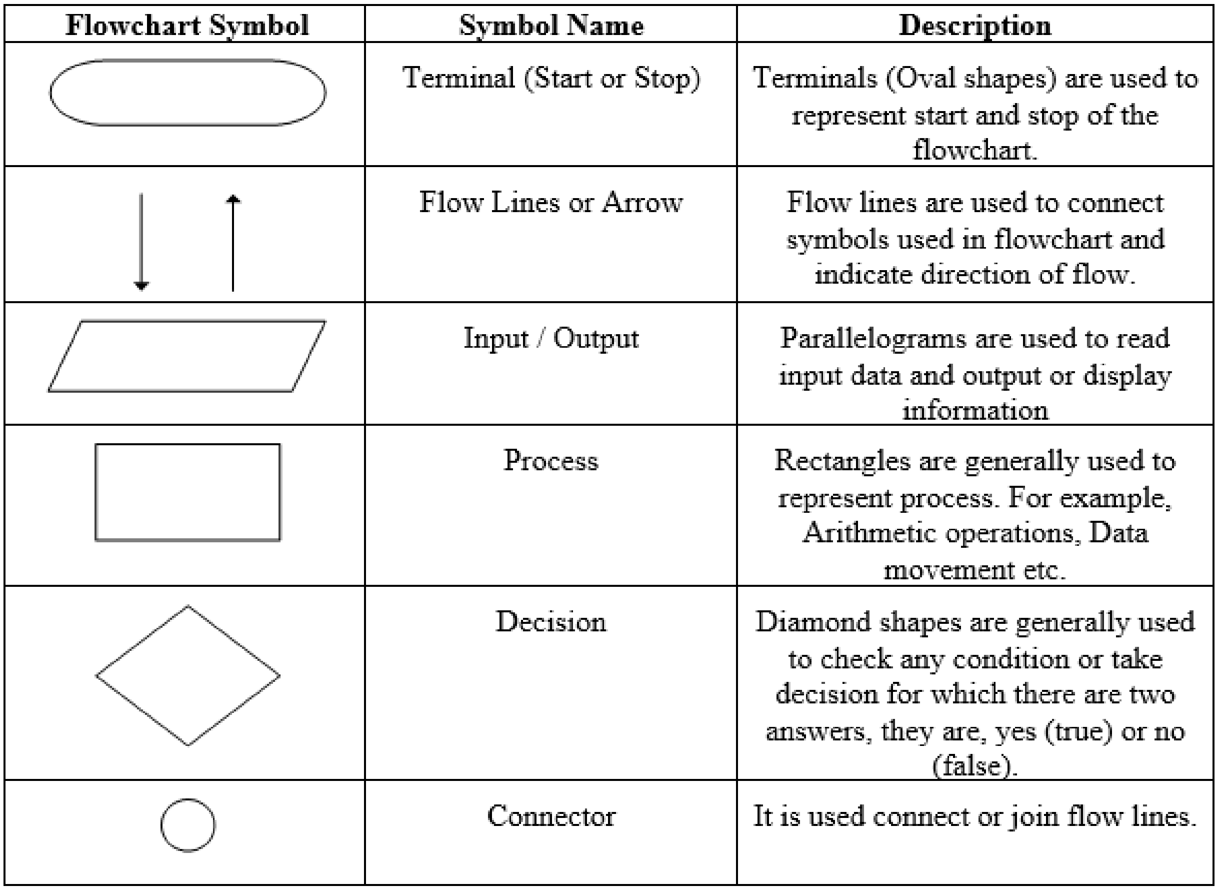
Flowcharting

Flowchart is basically a pictorial or diagrammatic representation of an algorithm using standard symbols.

In other words, flowchart is a graphical representation that explains the sequence of operations to be performed in order to solve a problem under consideration.

Standard Flowchart Symbols

To express different operations in the flowchart various standard symbols are used. All symbols are connected among themselves in order to show the flow of information and processing. Different symbols as prescribed by American National Standard Institute (ANSI) which are frequently required while drawing flowchart are tabulated below:



Flowchart Symbol	Symbol Name	Description
	Terminal (Start or Stop)	Terminals (Oval shapes) are used to represent start and stop of the flowchart.
	Flow Lines or Arrow	Flow lines are used to connect symbols used in flowchart and indicate direction of flow.
	Input / Output	Parallelograms are used to read input data and output or display information
	Process	Rectangles are generally used to represent process. For example, Arithmetic operations, Data movement etc.
	Decision	Diamond shapes are generally used to check any condition or take decision for which there are two answers, they are, yes (true) or no (false).
	Connector	It is used connect or join flow lines.

Following guidelines must be followed while preparing the flowcharts:

- Standard symbols should be used while drawing flowchart.
- Ensure that flowchart has START (or BEGIN) and STOP (or END).
- Flowchart should be neat, clean and easy to follow. There should be no any ambiguity.
- The usual direction of flowchart is from top to bottom or from left to right.
- The terminal symbol, that is, START/BEGIN or STOP/END should have only one flow line.
- Only one flow line should come out from process symbol.
- Only one flow line should enter a decision symbol, but two or three flow-lines, one for each possible answer, can leave the decision symbol.
- If the flowchart is lengthy and complex connector symbol should be used to reduce the number of flow lines.

- Avoid intersection of flow lines.
- Use annotation symbol to describe steps more clearly.

Example of Flowchart

Flowchart example for calculating simple interest is shown below:

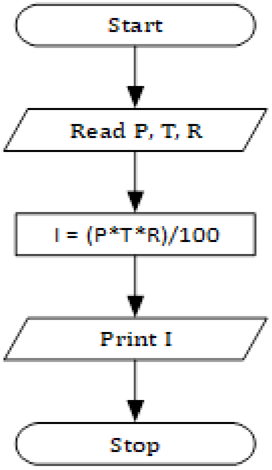


Figure: Flowchart for calculating simple interest

Advantages of Flowchart

Drawing flowchart while solving any problem has following advantages:

Effective Communication : Flowcharts are better way of communicating the logic of the system.

Effective Analysis : Using flowchart problem can be analyzed more efficiently.

Easy Debugging and Efficient Testing : The Flowchart helps in debugging and testing process.

Efficient Coding : The flowcharts are very useful during program development phase.

Proper Documentation : Flowcharts serves as a good program documentation, which is needed for various purpose.

Efficient Program Maintenance : Maintenance of operating programs becomes easy with the help of flowchart.

Disdvantages of Flowchart

Flowchart has following disadvantages:

Complex Logic: For complicated logic, flowchart becomes complex and clumsy.

Difficulty in Modifications: If change is required in the logic then flowchart needs to be redrawn and requires a lot of time.

Coding (Programming)

In this stage, process of writing actual program takes place. A coded program is most popularly referred to as a source code. The coding process can be done in any language (high level and low level). The actual use of computer takes place in this stage in which the programmer writes a sequence of instructions ready for execution. Coding is also known as programming.

Good program possess following characteristics :

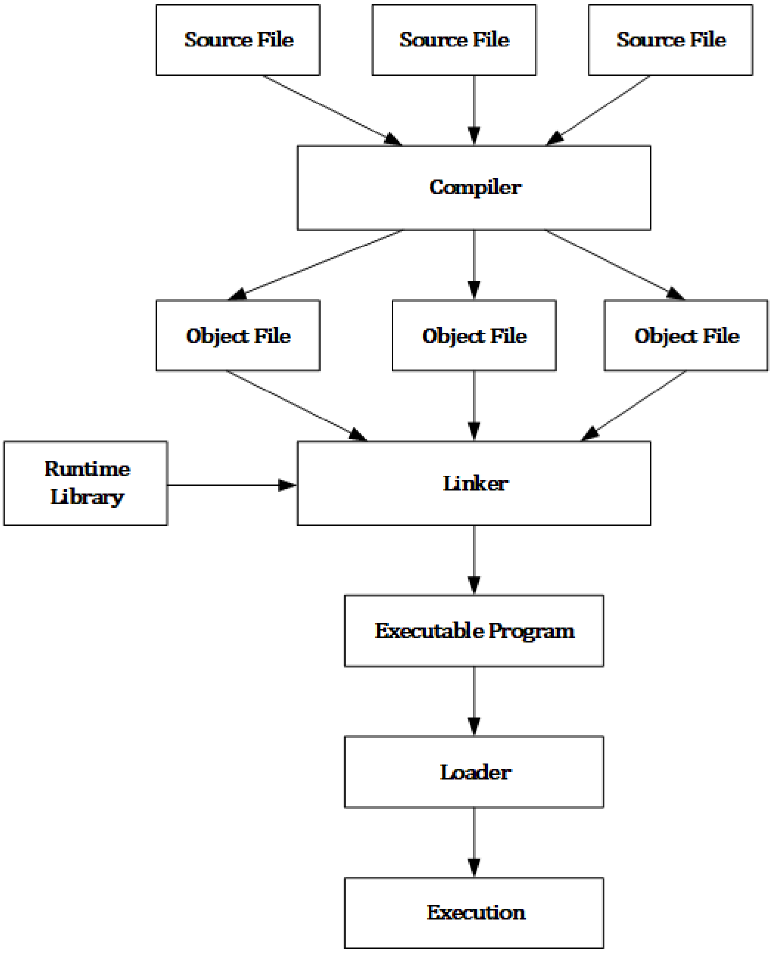
- Comment clauses in the program help to make the program readable and understandable by people other than the original programmer.
- It should be efficient.
- It must be reliable enough to work under all reasonable conditions to provide a correct output.
- It must be able to detect unreasonable error conditions and report them to the end user or programmer without crashing the system.
- It should be easy to maintain and support after installation.

Example:

```
#include<stdio.h>

int main()

{

   int length,breadth,area;

   printf("Enter the length and breadth of rectangle:");

   scanf("%d %d",&length,&breadth);

   area=length*breadth;

   printf("The area of the rectangle is: %d",area);

   return 0;

}
```

Compilation and Execution

A source code must go through several steps before it becomes an executable program. In the first step the source code is checked for any syntax errors. After the syntax errors are traced out a source file is passed through a compiler which first translates high level language into object code (A machine code not ready to be executed). A linker then links the object code with pre-compiled library functions, thus creating an executable program. This executable program is then loaded into the memory for execution. General compilation process is shown in Figure below:



Debugging and Testing

Debugging is the process of finding errors and removing them from a computer program, otherwise they will lead to failure of the program. Even after taking full care during program design and coding, some errors may remain in the program and these errors appear during compilation or linking or execution. Debugging is generally done by program developer.

Testing is performed to verify that whether the completed software package functions or works according to the expectations defined by the requirements. Testing is generally performed by testing team which repetitively executes program with intent to find error. After testing, list of errors and related information is sent to program developer or development team.

Documentation

The program documentation is the process of collecting information about the program. The documentation process starts from the problem analysis phase to debugging and testing. Documentation consists two types of documentation, they are:

- Programmer's Documentation
- User's Documentation

Programmer’s documentation contains all the technical details. Without proper documentation it is very difficult even for the original programmer to update and maintain the program. A programmer’s documentation contains the necessary information that a programmer requires to update and maintain the program. These information includes:

- Program analysis document, with a concise statement of program’s objectives, outputs and processing procedures.
- Program design documents with appropriate flowcharts and diagrams.
- Program verification documents for outlining, checking, testing and correction procedures along with the list of sample data and results.
- Log used to document future program revision and maintenance activity.

User documentation is required for the end user who installs and uses the program. It consists instructions for installation of the program and user manual.