Problem Solving

When a program is to be developed, a number of steps have to be followed before one could start writing the actual program. One cannot just sit in front of a computer and start typing programming code! It is very important that the program is first designed properly. If a program is not design properly, the program is going to face a lot of problems at a later stage. A problem could have several solutions that work out properly and not only one. However, coming up with the best solution is very difficult but important.

To design a good solution to the problem facing the programmer, one has to go through the following steps carefully:

- problem must be well defined for requirements
- specifications are created and targets for the solution are drawn
- any inputs and outputs required are carefully specified

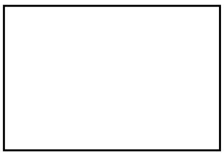
Algorithm

When a solution to a problem is designed, it is designed in a definite number of logical steps that a program follows in order to solve a problem. This is called an **algorithm**. In other words, an algorithm is a step by step solution to a problem.

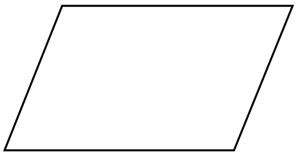
One can use one to two tools to write down an algorithm – Flowcharts or Pseudocode

Flowcharts

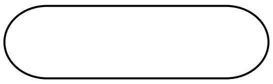
Flowcharts are a very important tool that can be used to design an algorithm. Flowcharts show the steps of an algorithm as a diagram using short text in pre-defined symbols. Each symbol stands for a certain type of programming activity. The symbols are linked together by arrows which indicate the flow of the program itself. It is necessary for each box to contain a few words to explain what event is taking place. These are the various symbols that can be used in a flowchart:



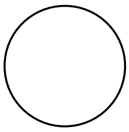
processing



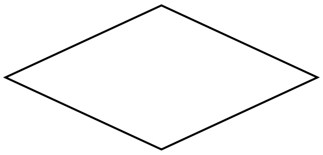
Input/output



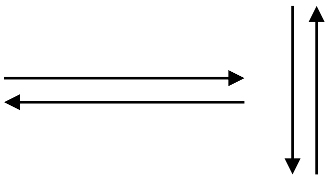
terminal



connector



decision



Flow direction

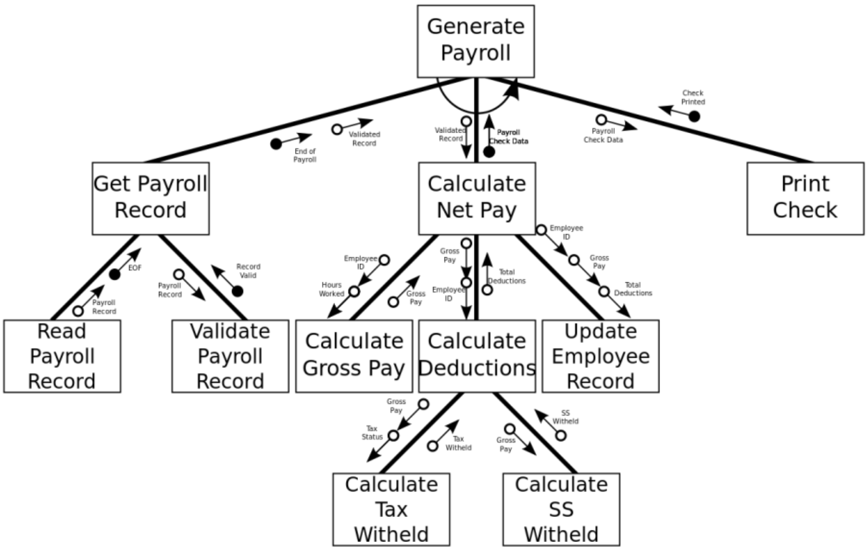
Pseudocode

Pseudocode uses simple English-like statements to describe each step of an algorithm without giving any attention to the grammar of the language. However, it is very important to use a new line for every step to be taken and proper indentation would be very useful. The problem with pseudocode is that there are no well-defined rules on how to write the steps and therefore the design might not be perfectly understood by the programmer.

Structure Charts

Sometimes the problem to be solved is too complicated. So, the programmer, before designing the algorithm, tries to break down the huge problem into smaller problems. The programmer creates a structure chart to show, using a diagram, how the problem was broken down into smaller problems. It also shows the hierarchical relationships among the small problems.

A structure chart is NOT a flowchart. It has nothing to do with the logical sequence of tasks. It does NOT show the order in which tasks are performed. It does NOT illustrate an algorithm. The following is an example of a structure chart of a program that calculate the payroll of employees in a company.


Generate Payroll
End of Payroll
Validated Record
Validated Record
Payroll Check Data
Payroll Check Data
Check Printed
Get Payroll Record
Calculate Net Pay
Print Check
EOF
Payroll Record
Payroll Record
Record Valid
Employee ID
Hours Worked
Gross Pay
Gross Pay
Employee ID
Total Deductions
Employee ID
Gross Pay
Total Deductions
Read Payroll Record
Validate Payroll Record
Calculate Gross Pay
Calculate Deductions
Update Employee Record
Gross Pay
Tax Status
Tax Witheld
Gross Pay
SS Witheld
Calculate Tax Witheld
Calculate SS Witheld