Unit 4

#  PROGRAM DEVELOPMENT PROCESS

While writing a computer program, it is absolutely necessary to write .each and every instruction in the correct sequence. The logical control of element within the computer program is the most important aspect of programming.

The various stages in the development of a computer program are :

* + - Problem Definition
		- Program Design
		- Coding
		- Debugging
		- Testing
		- Documentation
		- Maintenance

#### Problem Definition:

The first step in the process of program development is the thorough understanding and identification of the problem for which is the program or software is to be dev eloped. In this step the problem has to be defined formally. All the factors like Input/output, processing requirement, memory requirements, error handling, interfacing with other programs have to be taken into consideration in this stage.

#### Program Design:

The next stage is the program design. The software developer makes use of tools like algorithms and flowcharts to dev elop the design of the program.

#### Algorithm:

An algorithm represents the logic of the processing to be performed. It is a sequence of instructions which are designed in such a way that if they are executed in the specified sequence, the desired goal is achieved. It is imperative that the result be obtained after execution of a finite number of steps.

In an algorithm,

* + Each and every instruction has to be precise and clear.
	+ The instruction has to be executed in a finite time.
	+ W hen the algorithm terminates the desired result should be achieved.

#### Flowchart :

A flowchart is a pictorial representation of the algorithm. It represents the steps involved in the procedure and shows the logical sequence of processing using boxes of different shapes. The instruction to be executed is mentioned in the boxes. These boxes are connected together by solid lines with arrows, which indicate the flow of operation.

The first step in the design of a program is the algorithm. The algorithm is then represented in the form of a flowchart and the flowchart is then expressed in the computer language to actually prepare the computer program.



***Fig. Symbols used in a flowchart.***

**The symbols used in the flowchart are: Terminal (Start/Stop):**

The symbol denotes the beginning, ending and halt operation in the program logic. Thus it is the first and the last Symbols! in any nov-chart to indicate the start and stop of the process.

#### Input/output:

Indicates when an input or output operation is to performed in the process.

#### Processing:

This represents the arithmetic operations like addition, subtraction etc, and the movement of data in the process

#### Decision:

The decision symbol is used to represent a point where one or more alternatives is possible. The criteria for decision making are written in the decision box. On each path the condition for which the particular path is to be followed has to be written. During the execution the appropriate path will be followed depending upon the result of the decision.

#### Flow lines:

These indicate the flow of operation ie. The sequence of instruction execution. Flow lines are normally drawn with arrowhead indicating the direction of flow of the program.

#### Connectors:

A connector is used to indicate the logic flow from one page to another when the flowchart becomes very long. Connectors do not represent any operation. They are mainly used in the flowchart for clarity and convenience.

#### Types of logic used in the flowchart:

**Sequential Execution:** In this logic the instructions are executed one after the other sequentially.

**Transfer of control:** This is a logic which is used when the option to be chosen depends upon the result of the decision. The control is transferred to a particular path if the result of the decision branches to that path.

**Looping:** In looping or repetitive logic, an instruction or a number of instructions are executed more than once. The instructions are executed till the decision criteria is true. The decision criteria can be placed before the loop or after the loop depending upon the statements which are to be executed in the loop.

#### Let us see some examples of writing algorithms and flowcharts.

1. **Write an algorithm and develop a flowchart to convert the temperature input in Celsius scale to Fahrenheit scale.**

**Solution :**

First write a detailed stepwise algorithm to do the conversion

**Step 1:** Start.

**Step 2:** Input temperature in Celsius (C).

**Step 3:** Convert to Fahrenheit (F) using the formula F = 9/5 \* C + 32.

**Step 4**: Print the temperature in Fahrenheit (F).

**Step 5:** Stop.

Next on the basis of this step wise algorithm develop the flowchart using the appropriate flowchart symbols as follows:

#### Fig 6.3

1. **Write an algorithm and flowchart to read two numbers A and B and compare them. If**

**A is greater than B print, A is greater than B else print B is greater than A. Algorithm :**

**Step 1:** Start.

**Step 2:** Input values of A and B.

**Step 3:** Compare values of A and B (Is A > B?).

**Step 4:** If yes then print “A is greater than B”.

**Step 5:** If no, the print “B is greater than A”. Now draw the flowchart for the above.



#### Other techniques which can be useful in designing of the program are:

**Modular Programming:**

Using this method the entire program is divided into smaller manageable modules so that the smaller modules can be designed, coded and debugged separately.

#### Top-Down Design:

Here the overall problem is first defined in terms of general subtask. These subtasks are divided into further sub tasks.

#### Coding:

Once the design process is complete, the actual computer program is written, i.e. the instructions are written in a computer language. Coding is generally a v ery small part of the entire program development process and also a less time consuming activity in reality. In this process all the syntax errors i.e. errors related to spelling, missing commas, undefined labels etc. are eliminated. For effectiv e coding some of the guidelines which are applied are

* Use of meaningful names and labels of variables,
* Simple and clear expressions,
* Modularity with emphasis on making modules generalized,
* Making use of comments and indenting the code properly,
* Av oiding jumps in the program to transfer control.

#### Debugging:

At this stage the errors in the programs are detected and corrected. This stage of program development is an important process. Debugging is also known as program validation.

Some common errors which might occur in the programs include:

* Un initialization of variables,
* Reversing of order of operands,
	+ Confusion of numbers and characters,
	+ Inv erting of conditions eg jumping on zero instead of on not zero.

#### The tools or methods which can be used to debug the programs include:

* + **Simulator** -This is a computer program which simulates the execution of the program on another computer.
	+ **Logic Analyzers:** This test instrument detects the states of digital signals during each clock cycle and stores them in memory. It then displays this data on the monitor.
	+ **Breakpoints:** This is introduced in the program to halt the execution at some intermediate point. At this point, the v alues of the various inputs, outputs, variables etc. can be checked.
	+ **Software interrupts:** This is an instruction which saves the current value of the program counter and then branches to a specific memory location. At this location there is the debugging program which displays the status information.
	+ **Memory dump:** This gives a listing of the current contents of a section of the memory. Memory dumps are common in most simulator programs, and microcomputer systems.
	+ **Trace routine:** This program prints the current status of the processor at specified intervals.

#### Testing (Validation):

The program is tested on a number of suitable test cases. A test plan of the program has to be done at the stage of the program design itself. This ensures a thorough understanding of the specifications. The most trivial and the most special cases should be identified and tested. It is always useful to include the maximum and minimum values of all variables as test data.

#### Documentation:

Documentation is a very essential step in the program development. Documentation help$ the users and the people who maintain the software. This ensures that future modification if required can be done easily. Also it is required during redesigning and maintenance.

#### Maintenance:

Updating and correction of the program for changed conditions and field experience is accounted for in maintenance.

Maintenance becomes essential in following situations:

* + - * Change in specification,
			* ChangelrY equipment,
			* Errors which are found during the actual execution of the program. Modularity, structured programming, thorough testing and debugging and proper documentation greatly reduce the time and cost of maintenance of the software.

#### Write algorithms and draw flowcharts for the following:

1. Input the marks of five subjects of a student. Calculate the percentage and print the grades as follows:

Grade A for 90% and above, grade B for percentage greater than or equal to 75 and less than 90 and grade C for percentage less than 75.

1. Find the area and perimeter of a rectangle given the sides s1 and s2.
2. a) Algorithm :

**Step 1 :** Start,

**Step 2 :** Input marks of five subjects (ml,m2,m3,m4,m5).

**Step 3** : Calculate percentage (p = (ml -i- m2 + m3 + m4 + m5)/5).

**Step 4 :** Check if percentage (p) >= 90.

**Step 5 :** If yes, print “A grade”.

**Step 6 :** If no, check if percentage (p) >= 75.

**Step 7 :** If yes, print “B grade”.

**Step 8** : If no, print “C grade”. Step 9: Stop.

#### Stop

1. Algorithm :

**Step 1 :** Start.

**Step 2 :** Enter the sides of the rectangle s1 and s2.

**Step 3 :** Calculate area (A) = s1 x s2.

**Step 4 :** Calculate perimeter (P) = 2 \* (s1 + s2).

**Step 5 :** Output area (A) and perimeter (P).

**Step 6 :** Stop.



