UNIT – 1 (part-2)

Introduction of Logic Development Tools

BCA SEM – 1

PROBLEM SOLVING METHODOLOGIS AND PROGRAMMING IN C

Code : CS-02

Presented By : Dhruvita Savaliya

Topics :

- Introduction of Logic
- Necessary Instructions for Developing Logic
- Basics of Flow Chart
- Dry-run and its Use
- Other Logic development techniques

❖ Introduction of Logic :

- Good logic in programming is a fundamental key to being a good developer.
- Developing good logic is the fool-proof method for you to learn and navigate through different programming languages flexibly.
- Generally program is a solution of any problem which contains complex operation and input output process.
- Every problem solve with the help of arithmetic and logical operations.
- In simple terms, logic is a technique of writing code for solving any problem is simple form.
- User can solve any problem using simple or complex logic.
- Program solution can represented in two different forms :

1. Textual form
2. Graphical form

Necessary Instructions for Developing Logic:

- The best practice of logic development is on paper work of any small or big program solution.
- The first step is on paper work for developing a logic of any problem.
- To write program solution step by step including input/ output statements and actual logic of the program in textual form, that is known as algorithm.
- To represent program logic with different symbols, that is known as flowchart.
- This method is very useful to know the entire process of program.
- First of all think about the different methods or techniques to find out the program solution.
- Select the best method among the possible solution.
- List out numbers of variable required to solve the program and decide appropriate data type of every variable.
- Think about the input statements, it means which variable values we have to receive from the user.
- Think about an arithmetic and logical statements according to program logic.
- Display result of program according to particular format

- Basically we can describe solutions of any problem using following formats :

1. Represent solutions step by step in written format.
   →ALGORITHM
2. Represent solutions using graphical format
   →FLOWCHART

1. Algorithm :

- Before solving a problem, one should know what to do, how to do it, and what types of steps should be taken.
- So, an algorithm is a step-by-step method for solving a problem.
- An algorithm refers to a set of instructions that define the execution of work to get the expected results.
- To make the program work properly, we must have to properly design the algorithm.
- Designing the algorithm helps to utilize the computing resources effectively.
- Generally, the algorithms are written in natural language or in plain English language.
- We can represent algorithms via flowcharts, pseudo code, and others.

Example :

Step 1 : START

Step 2 : INITIALIZE X=0,Y=0 ,Z=0

Step 3 : INPUT X,Y

Step 4 : Z=X

Step 5 : X=Y

Step 6 : Y=Z

Step 7 : PRINT X,Y

Step 8 : STOP

Advantages of algorithm :

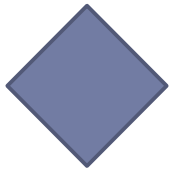
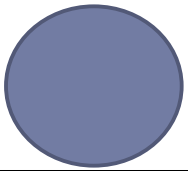
- Algorithms are easy to write.
- There is a use of human-readable techniques to understand logic.
- For big problems, algorithms can be written with moderate steps.

Disadvantages of algorithm :

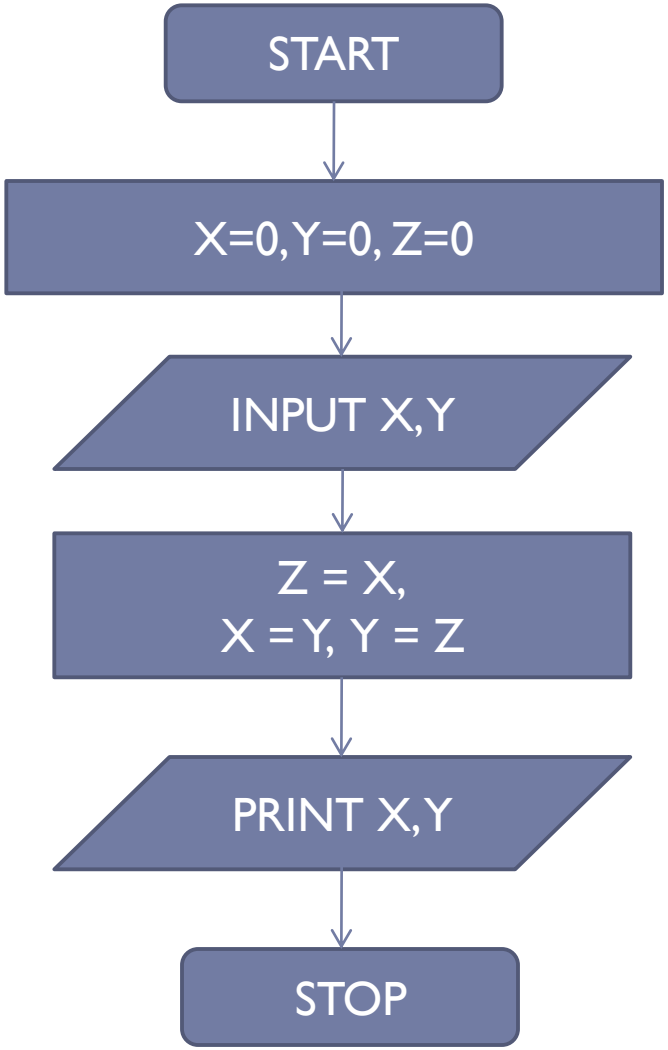
- Algorithms are difficult to debug.
- It is hard to show branches and loops using an algorithm.
- Jumping (or goto statements) makes the algorithm hard to trace the problems.

2. Flowchart :

- The Flowchart is a **graphical or symbolic representation** of process of problem solution.
- It uses various symbols to show the operations and decisions to be followed in a program.
- It flows in sequential order.
- Each step in the process is represented by different shapes and displays a short detail of the process step.
- Execution process in flowchart should be followed from top to bottom side and use appropriate keywords like START – STOP or BEGIN – END.

NAME	SYMBOL	USE
Terminal		This symbol represented the starting and ending process of flowchart (START / STOP or BEGIN – END)
Parallelograms		For receive information from the user or display information to the user. (INPUT / OUTPUT)
Process		It indicates a normal process executed or arithmetic operation.
Decision/ Conditional / Diamond		To make choice according to the condition being TRUE or FALSE /YES or NO .
Connector		Used to connect different parts of flowchart.
Arrows / Flow Lines		Join 2 symbols and also represents flow of execution.

Example :



Advantages of flowchart

- Proper debugging
- Effective analysis
- Efficient coding
- Proper documentation
- Efficient program maintenance

Disadvantages of flowchart

- Time-consuming
- Complex
- Difficult to modify
- It has no standard

Algorithm	Flowchart
It is a procedure for solving problems.	It is a graphic representation of a process.
The process is shown in step-by-step instruction.	The process is shown in block-by-block information diagram.
It is complex and difficult to understand.	It is intuitive and easy to understand.
It is convenient to debug errors.	It is hard to debug errors.
The solution is showcased in natural language.	The solution is showcased in pictorial format.
It is somewhat easier to solve complex problem.	It is hard to solve complex problem.
It costs more time to create an algorithm.	It costs less time to create a flowchart.

❖Dry – Run :

- Dry run is a step by step testing process of program logic which gives the variable status and output of a particular statement.
- It is execution process of program without compiler.
- It means, dry run is generally drawn on paper before actual execution so programmer can know actual output of the statement or about any error in the program.

- A dry run is the process of a programmer manually working through their code to trace the value of variables. There is no software involved in this process.
- A dry run is working through a section of a program manually.
- This is useful for locating errors, particularly run-time errors.
- A dry run can be performed on both a section of the program and the whole program.
- In the dry run process a trace table is useful for checking the values of variable while following the code of the program.
- A trace table traces the sequence in which program instructions are carried out and what the data values are at each stage.
- "A dry run is done by first creating a trace table with one column for each data type. The initial values are entered and the source code is worked through one command at a time changing the value of variables as needed."

- First initialize x, y, z with value : 0
- User Enter value of X is 5 and value of Y is 3

Dry run of Swap two variable values			
STEP	X	Y	Z
1	0	0	0
2	5	0	0
3	5	3	0
4	5	3	5
5	3	3	5
6	3	5	5
7	VALUE OF X IS : 3		
8	VALUE OF Y IS : 5		

❖Use of dry run in Programming :

- To know the status of variable at every step.
- To find out error in any inner part or block of the program.
- Dry run table is drawn without actual compilation process of program so programmer's syntax and logical knowledge is improved.
- For debugging program.
- To check any syntax or logical error from the program.
- To know the order of execution process of program.
- Very useful in looping programs to know the iteration station on different level.