
DFD
DATA
FLOW
DIAGRAM

Data Flow Diagram

DFD is the abbreviation for **Data Flow Diagram**. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. It is a graphical tool, useful for communicating with users ,managers and other personnel. it is useful for analyzing existing as well as proposed system.

It provides an overview of

- What data is system processes.
- What transformation are performed.
- What data are stored.
- What results are produced , etc.

Characteristics of DFD

- DFDs are commonly used during problem analysis.
- DFDs are quite general and are not limited to problem analysis for software requirements specification.
- DFDs are very useful in understanding a system and can be effectively used during analysis.
- It views a system as a function that transforms the inputs into desired outputs.
- The DFD aims to capture the transformations that take place within a system to the input data so that eventually the output data is produced.
- The processes are shown by named circles and data flows are represented by named arrows entering or leaving the bubbles.
- A rectangle represents a source or sink and it is a net originator or consumer of data. A source sink is typically outside the main system of study.

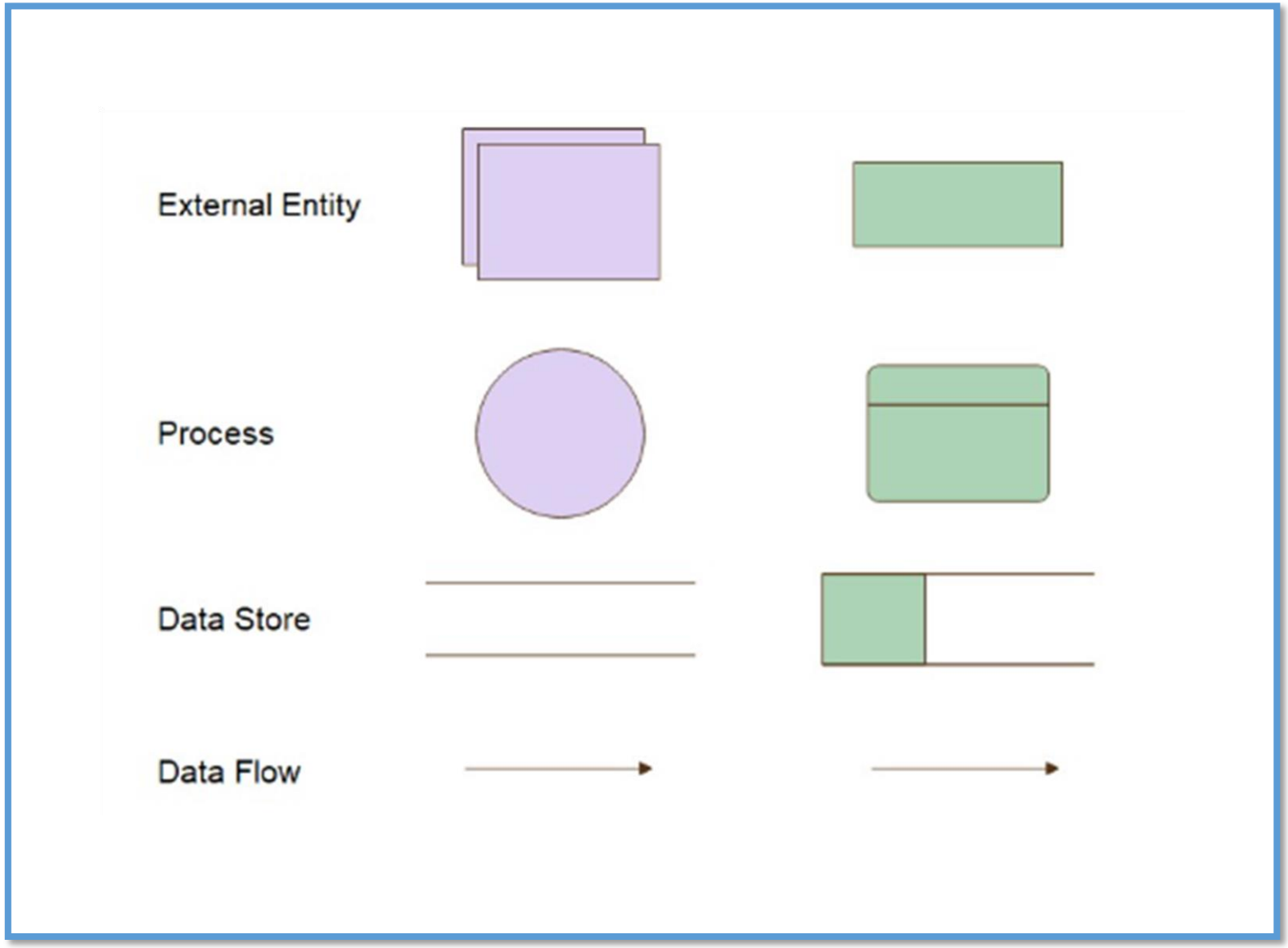
Components of DFD :

Process

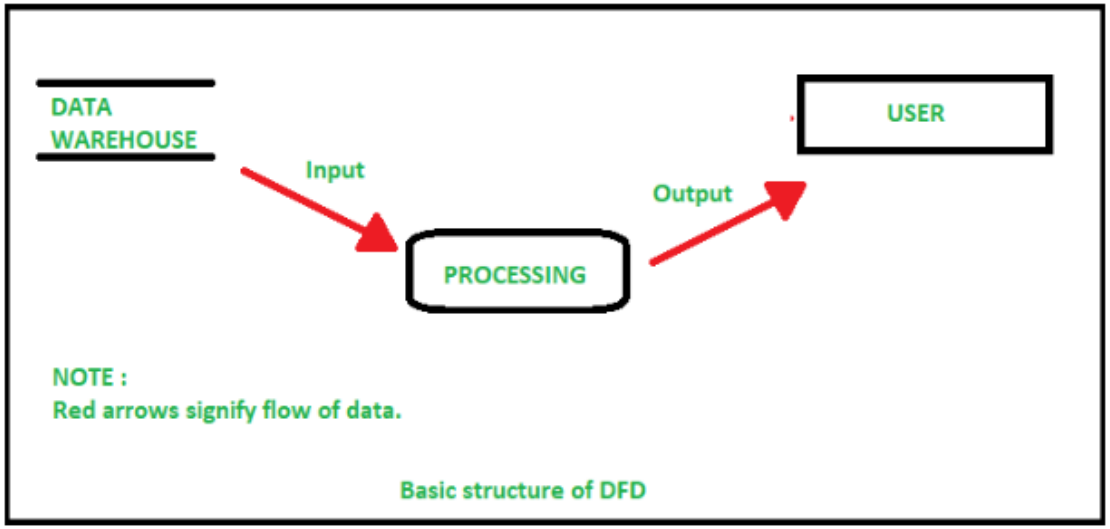
Data Flow

Data Store

External Entity


External Entity
Process
Data Store
Data Flow

Advantages of DFD	Disadvantages of DFD
• It helps us to understand the functioning and the limits of a system. • It is a graphical representation which is very easy to understand as it helps visualize contents. • Data Flow Diagram represent detailed and well explained diagram of system components. • It is used as the part of system documentation file. • Data Flow Diagrams can be understood by both technical or nontechnical person because they are very easy to understand.	• At times DFD can confuse the programmers regarding the system. • Data Flow Diagram takes long time to be generated, and many times due to this reasons analysts are denied permission to work on it.



Basic structure of DFD

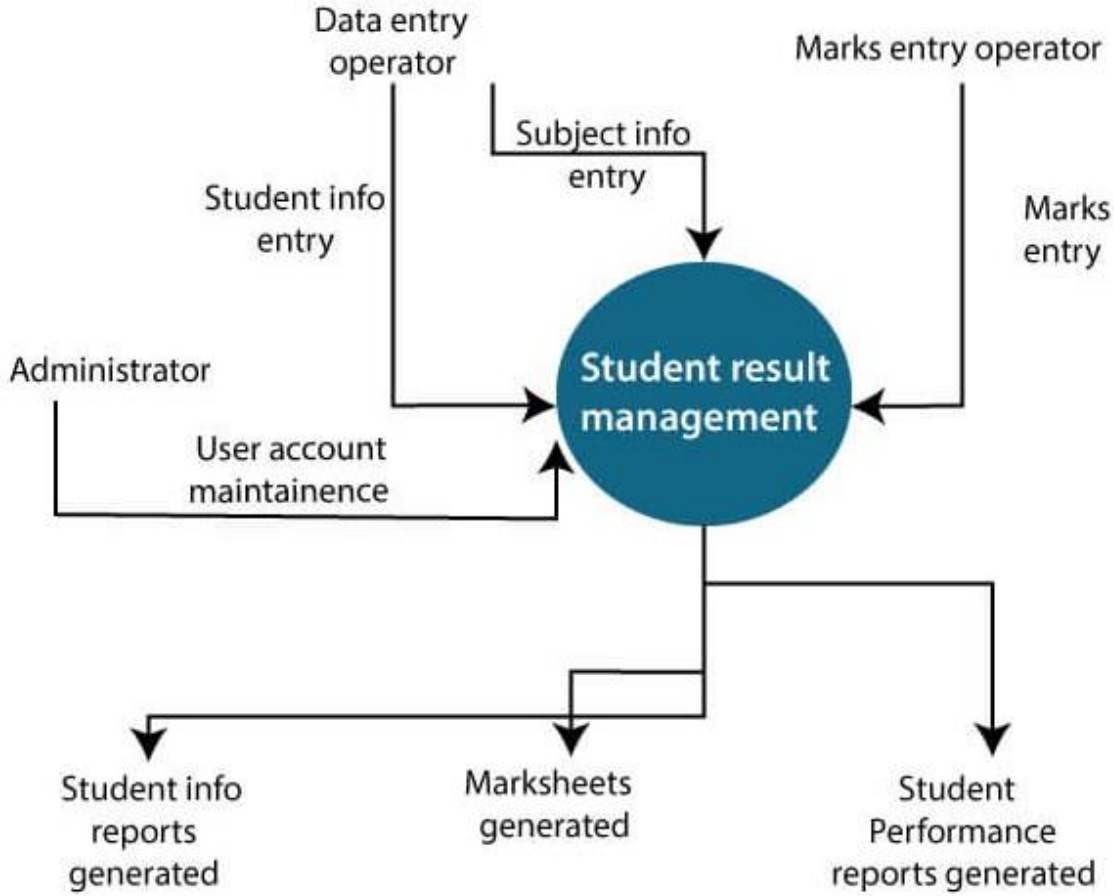


Fig: Level-0 DFD of result management system

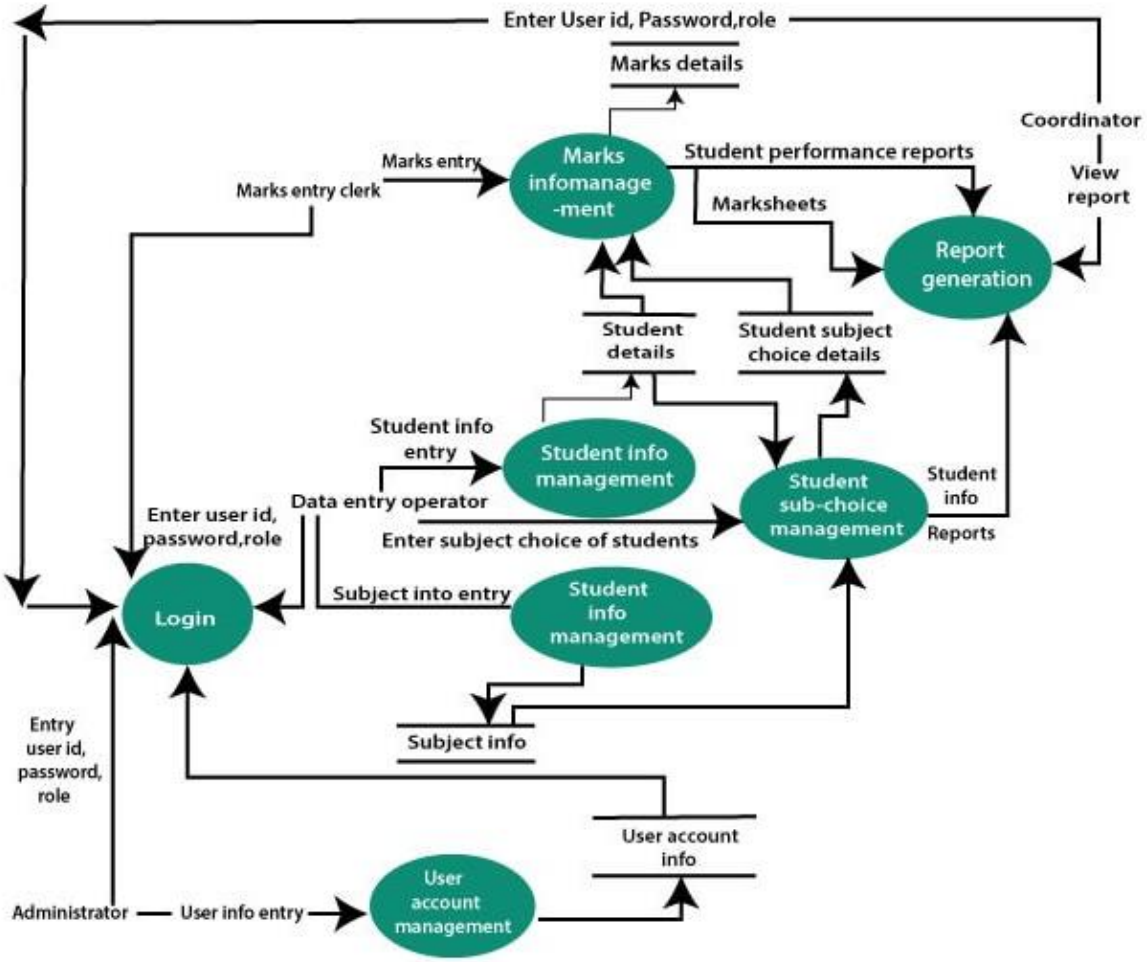
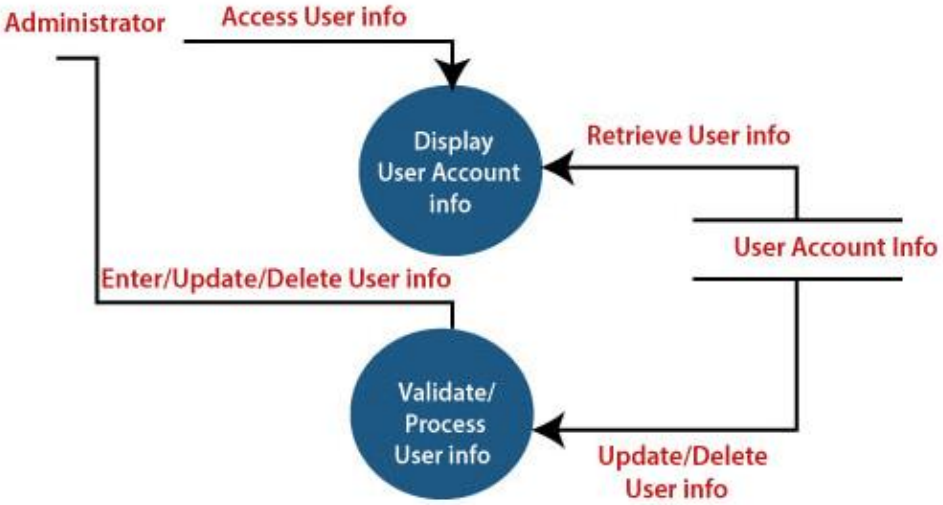


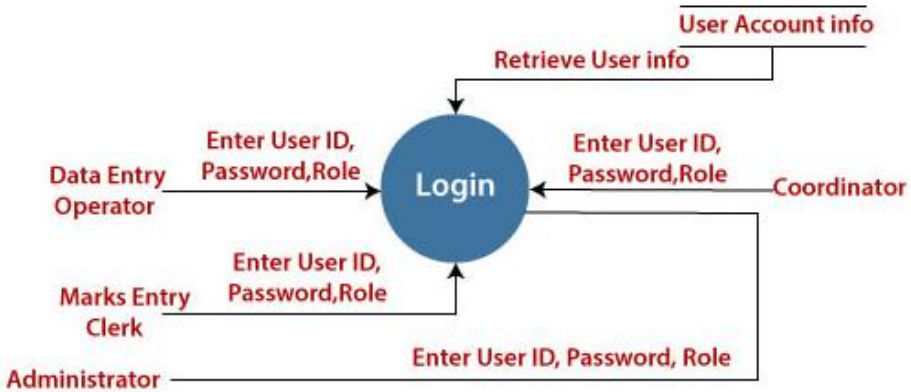
Fig: Level-1 DFD of result management system

1.User Account Maintenance

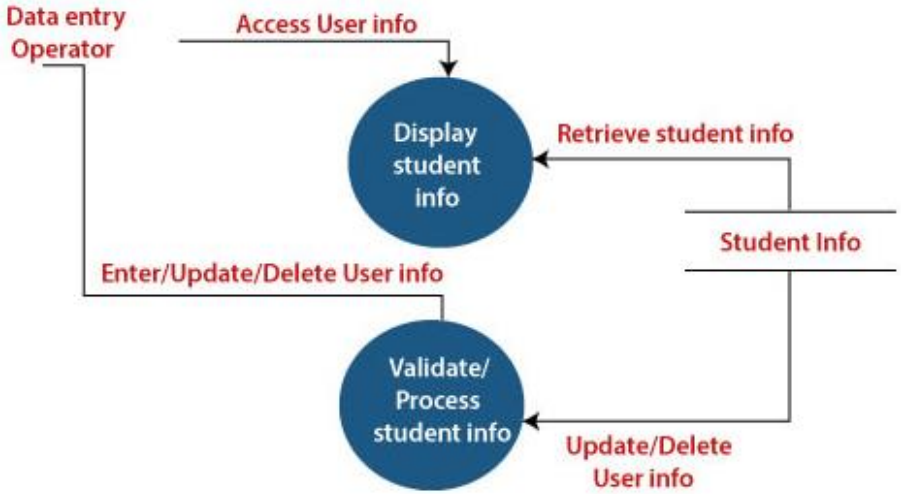


2. Login

The level 2 DFD of this process is given below:

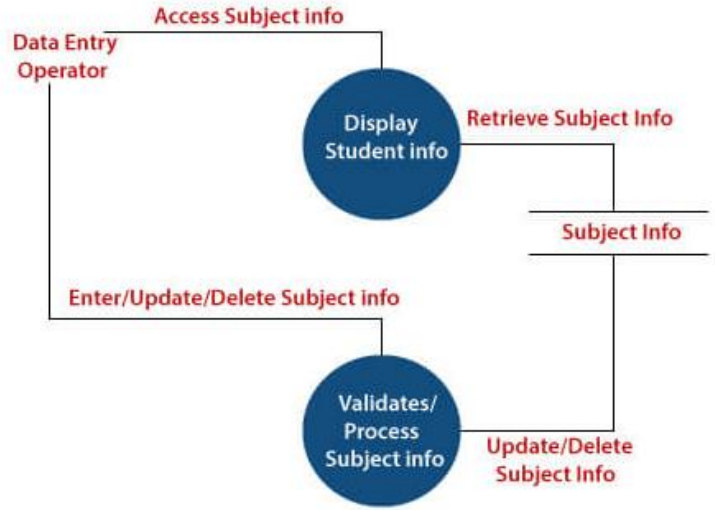


3. Student Information Management



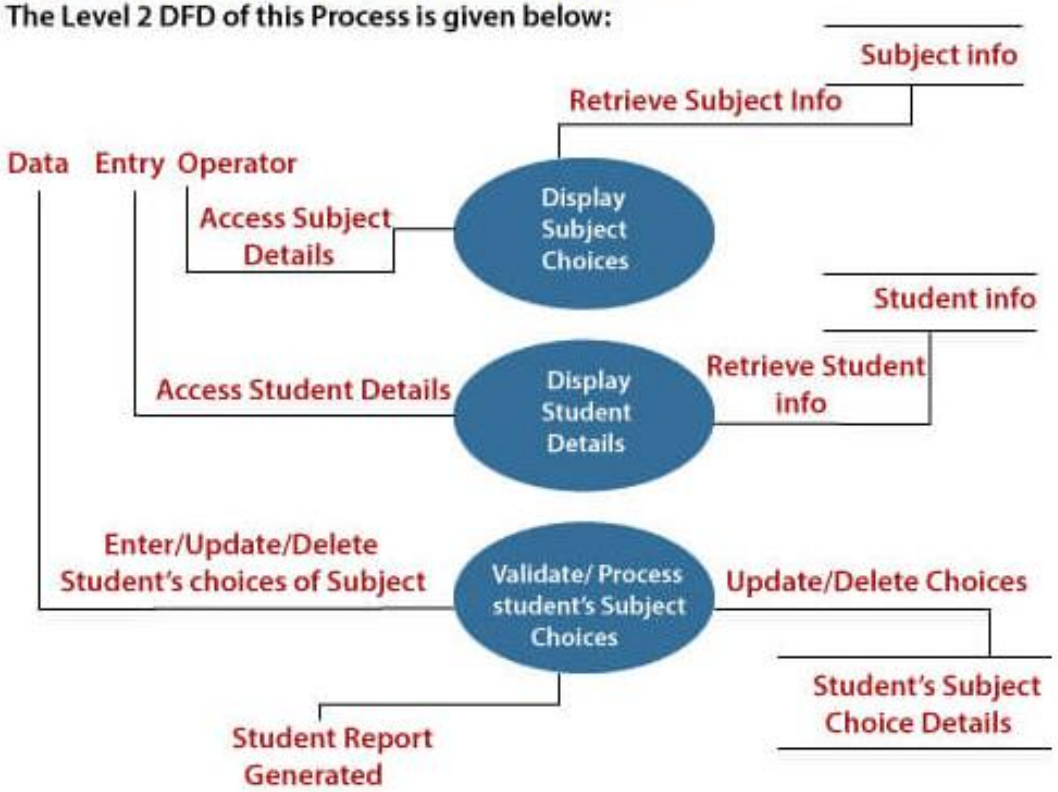
4. Subject Information Management

The level 2 DFD of this process is given below:



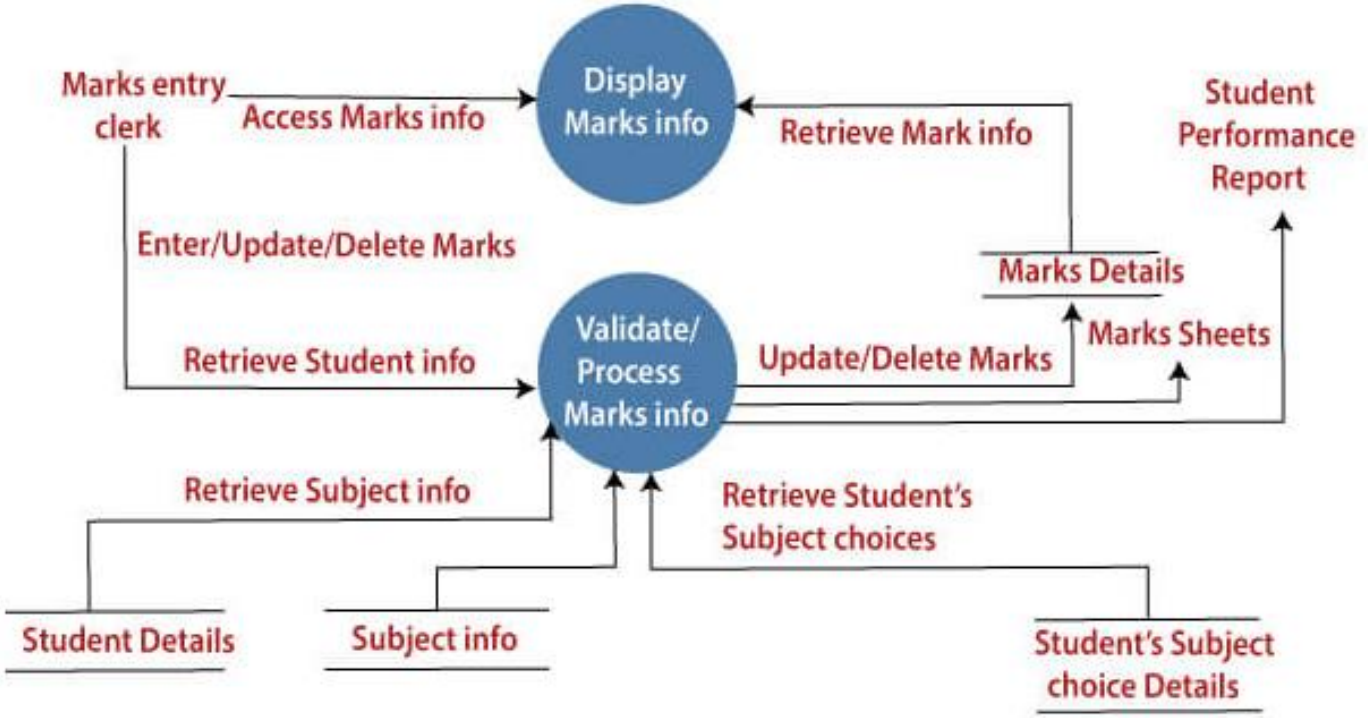
5. Student's Subject Choice Management

The Level 2 DFD of this Process is given below:

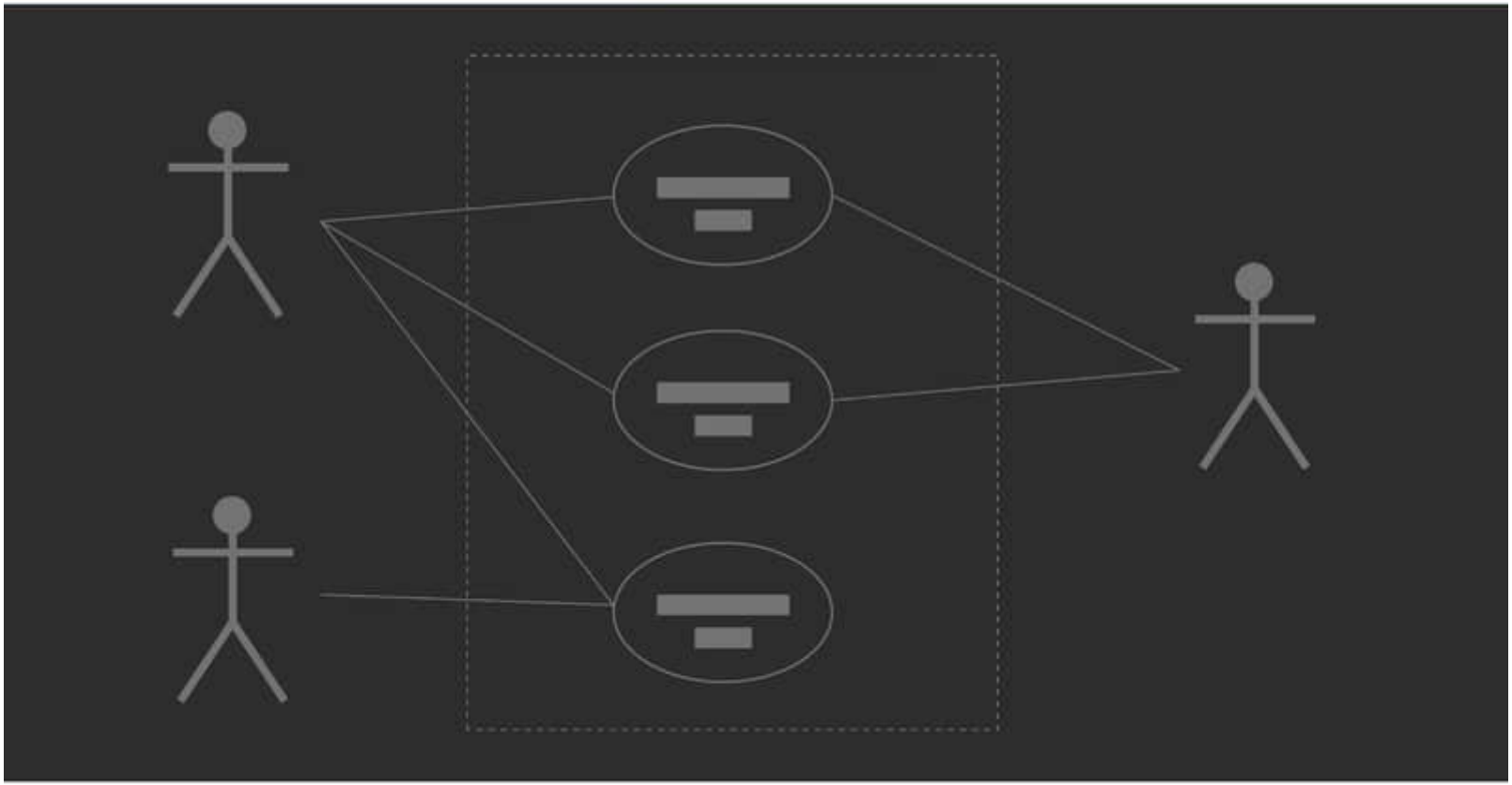


6. Marks Information Managment

The Level 2 DFD of this Process is given below:



Use Case

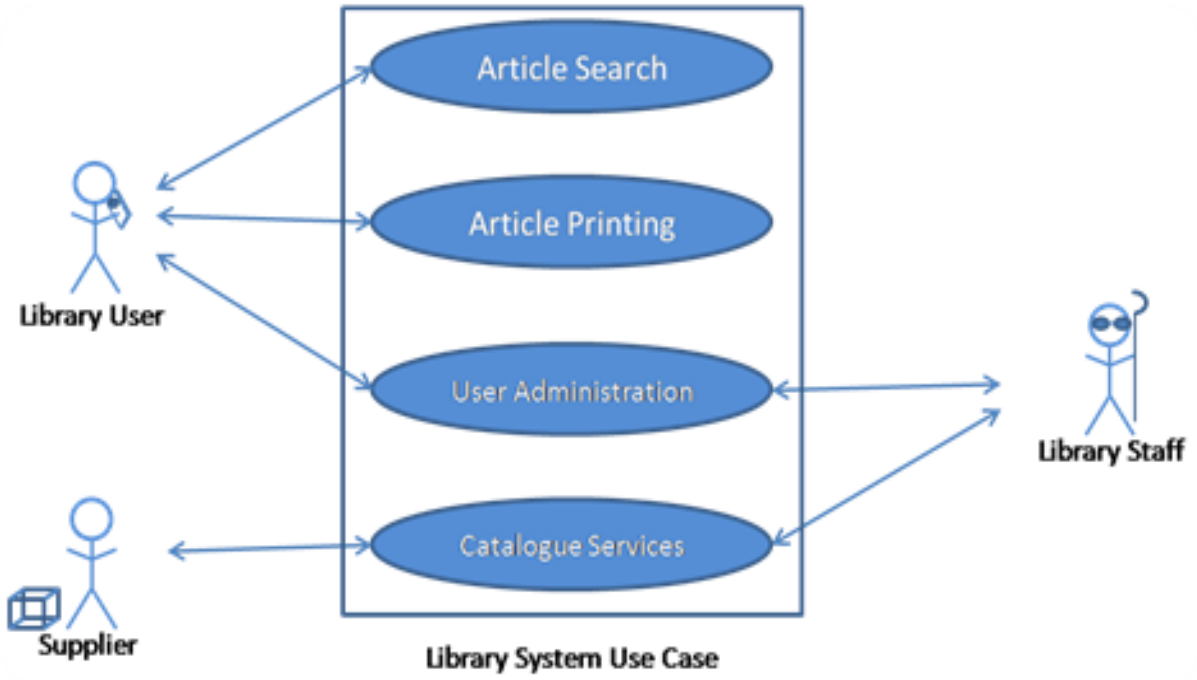


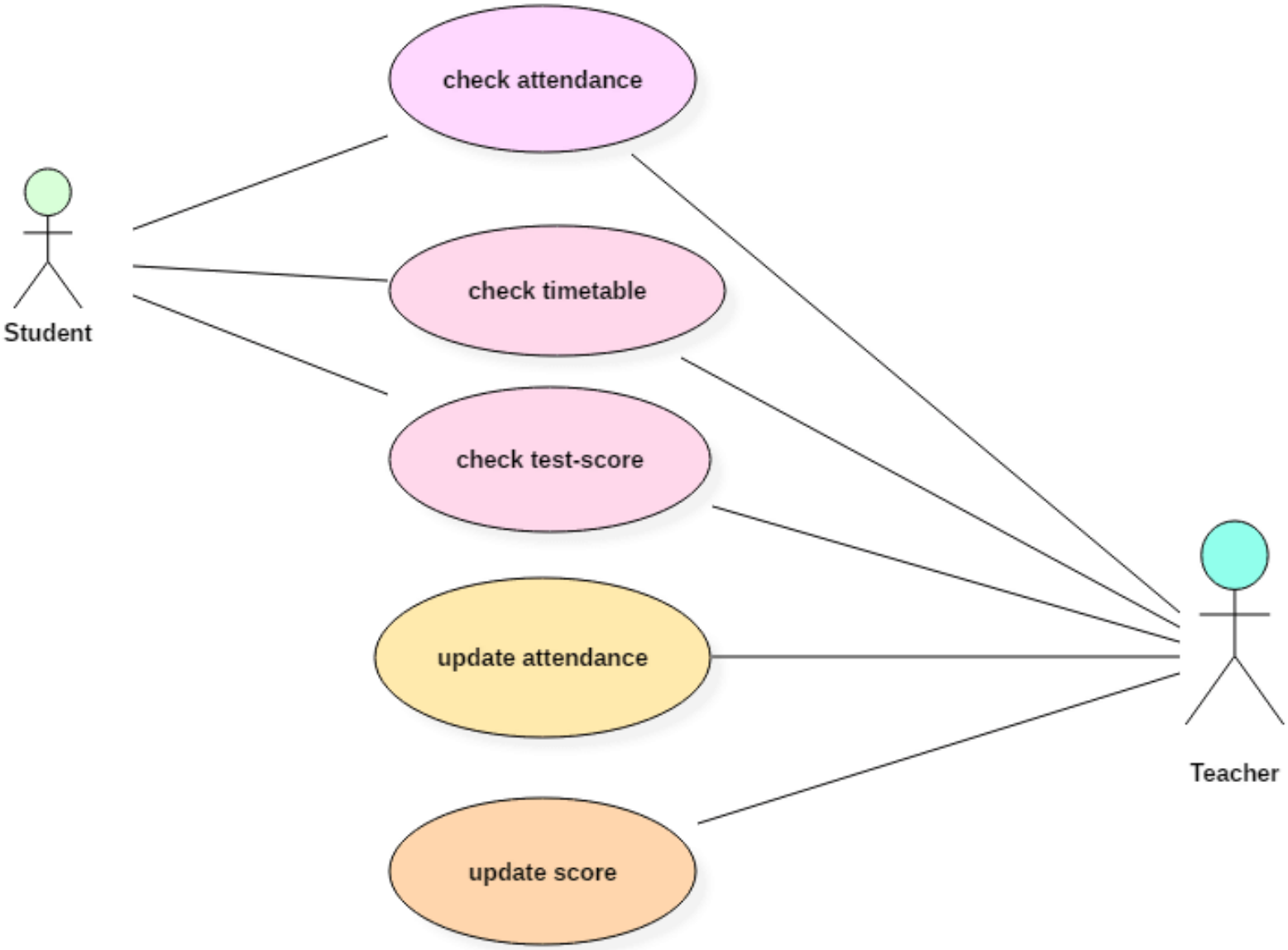
A use case is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.

Components of Basic Model

There are various components of the basic model:

1. Actor
2. Use Case
3. Associations




check attendance
check timetable
check test-score
update attendance
update score
Student
Teacher

When to use a use-case diagram?

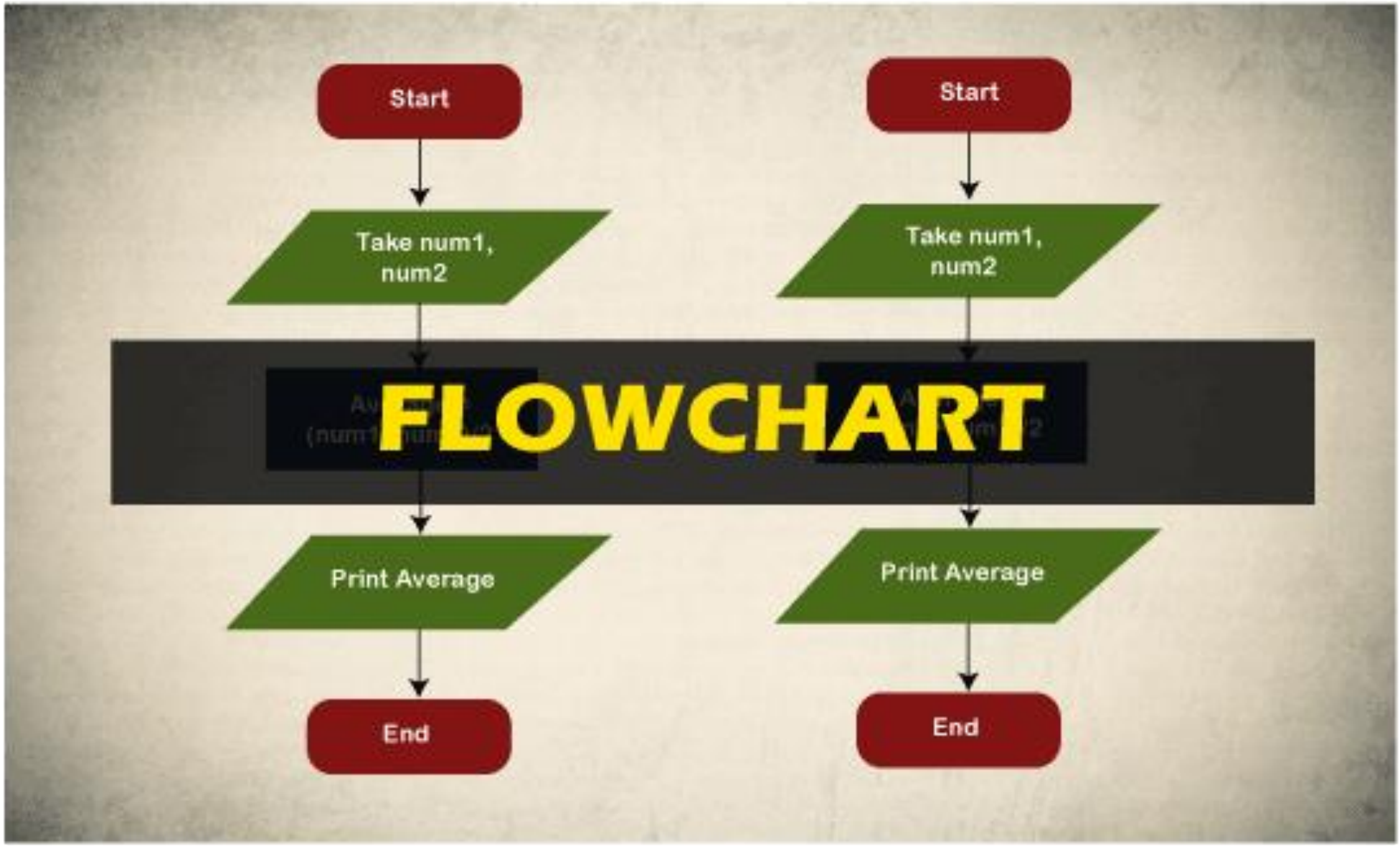
A use case is a unique functionality of a system which is accomplished by a user. A purpose of use case diagram is to capture core functionalities of a system and visualize the interactions of various things called as actors with the use case. This is the general use of a use case diagram.

The use case diagrams represent the core parts of a system and the workflow between them. In use case, implementation details are hidden from the external use only the event flow is represented.

With the help of use case diagrams, we can find out pre and post conditions after the interaction with the actor. These conditions can be determined using various test cases.

In general use case diagrams are used for:

1. Analyzing the requirements of a system
2. High-level visual software designing
3. Capturing the functionalities of a system
4. Modeling the basic idea behind the system
5. Forward and reverse engineering of a system using various test cases.


Start
Take num1, num2
Print Average
End
Start
Take num1, num2
Print Average
End
FLOWCHART

What is FlowChart?

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

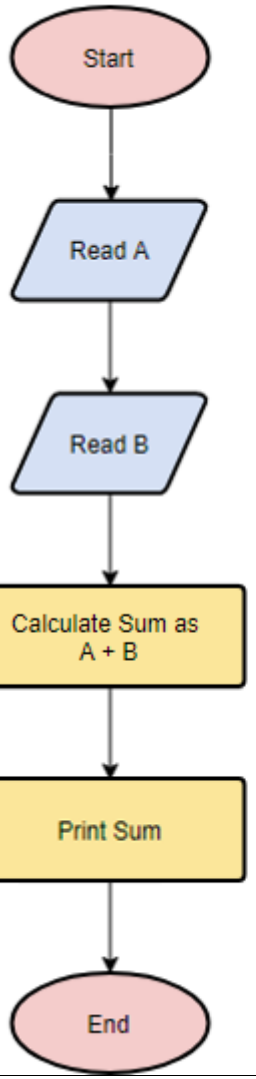
→ A flowchart (or flow chart) is a diagram that shows the steps in a process.

→ Flowcharts are often used for visualizing the sequence of actions or information needed for training, documenting, planning, and decision-making.

→ They often use symbols, shapes, and arrows to illustrate how one step leads to another.

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectagle represents a process
	Decision	A diamond indicates a decision

Find the sum of 529 and 256



Start

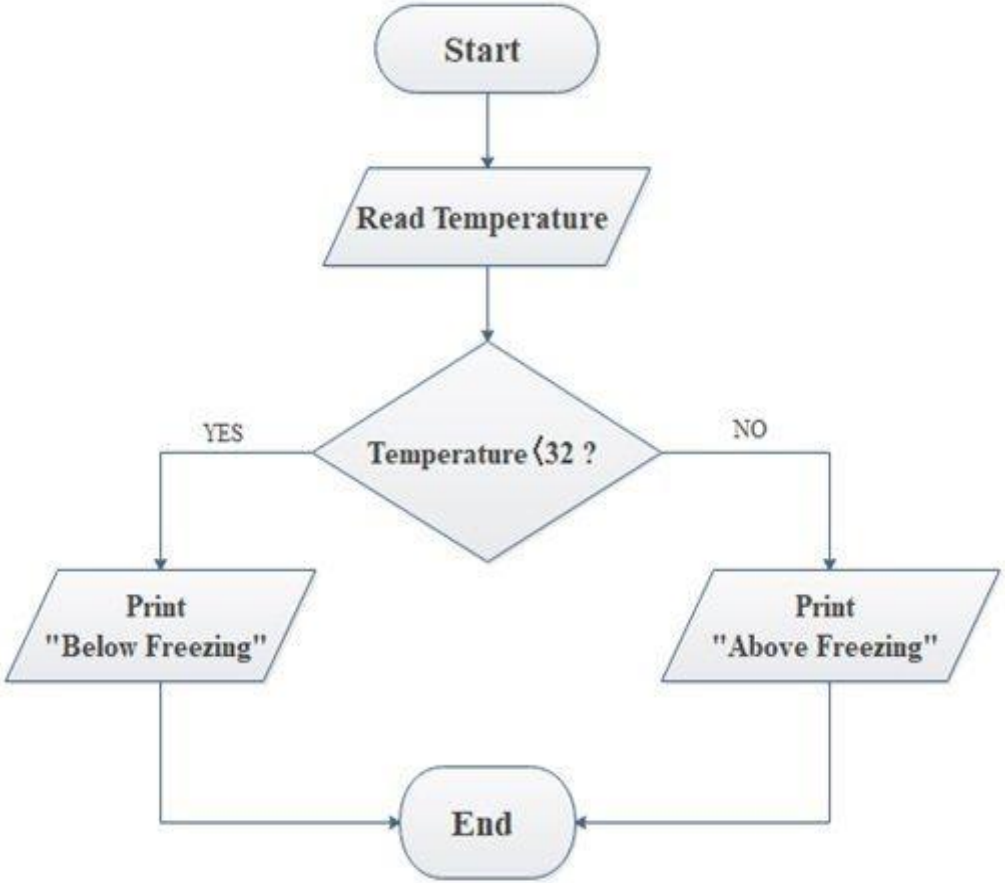
A = 529

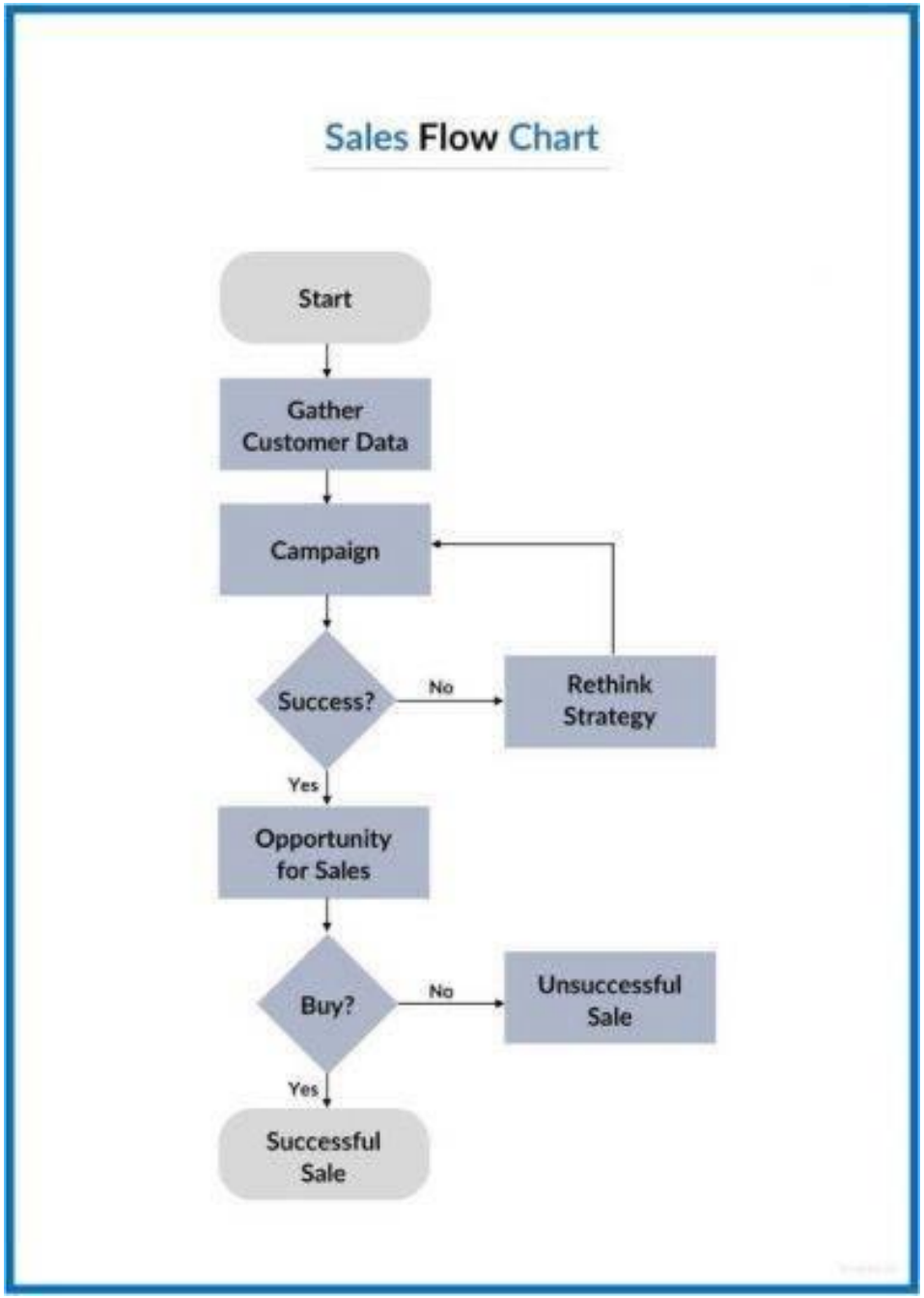
B = 256

Sum = 529 + 256

Sum = 785

End




Sales Flow Chart
Start
Gather Customer Data
Campaign
Success?
No
Rethink Strategy
Yes
Opportunity for Sales
Buy?
No
Unsuccessful Sale
Yes
Successful Sale

Algorithm Vs. Flowchart

Algorithms and flowcharts are different mechanisms used for designing different programs, particularly in computer programming. An algorithm is a step-by-step summary of the procedure, while on the other hand, a flowchart illustrates the steps of a program graphically.

What is an Algorithm?

An algorithm is a procedure or set of rules that defines how a program is to be executed. Or we can say that it is a set of instructions for solving a well-defined computational problem.

What is a Flowchart?

A flowchart is a graphical representation of the steps a program takes to process data. In this, we can use several geometric patterns to illustrate the numerous actions the program carries out.

With the help of the flowchart, the designer can efficiently segregate the various elements of the process. Also, it facilitates the analysis by giving step-by-step instructions on the problem.

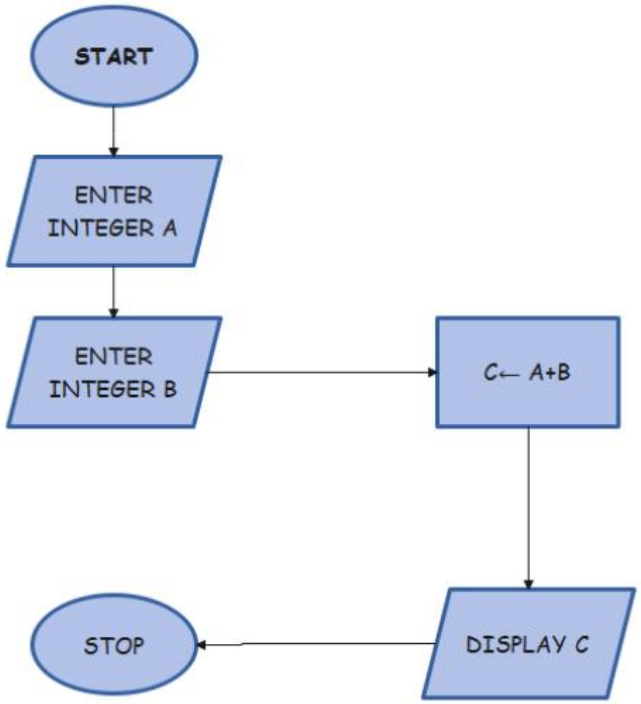
Difference Between Algorithm And Flowchart

Parameters	Flowchart	Algorithm
Description	A flowchart is a graphical representation of the steps a program takes to process data. In this, we can use several geometric patterns to illustrate the numerous actions the program carries out.	An algorithm is a procedure or set of rules that defines how a program is to be executed. Or we can say that it is a set of instructions for solving a well-defined computational problem.
Complexity	It is easy to design and also very user friendly.	It is comparatively difficult to create and also a bit challenging to be understood by a layman.
Geometrical diagrams	It utilizes different types of geometrical shapes, symbols, and patterns.	An algorithm does not include any sort of geometrical pattern.
Scope of Usage	A flowchart can be used in different disciplines to describe a process.	Algorithms are used in the domain of mathematics and computer science.

Use	A flowchart is used in documenting, designing, and analyzing a program in different disciplines.	An algorithm is used to represent the concept of decidability.
Users	A Flowchart doesn't demand the knowledge of a computer programming language.	An algorithm demands the knowledge of a computer programming language.
Debugging	It is easy to debug the errors in flowcharts.	It is difficult to debug the errors in algorithms.
Implementation	In flowcharts, no rules are used.	In algorithms, predefined rules are used.
Branching and Looping	Simple to display branching and looping.	Hard to display branching and looping.
Solution	In a flowchart, the solution is represented in a graphical format.	In an algorithm, the solution is presented in non non-computer language.

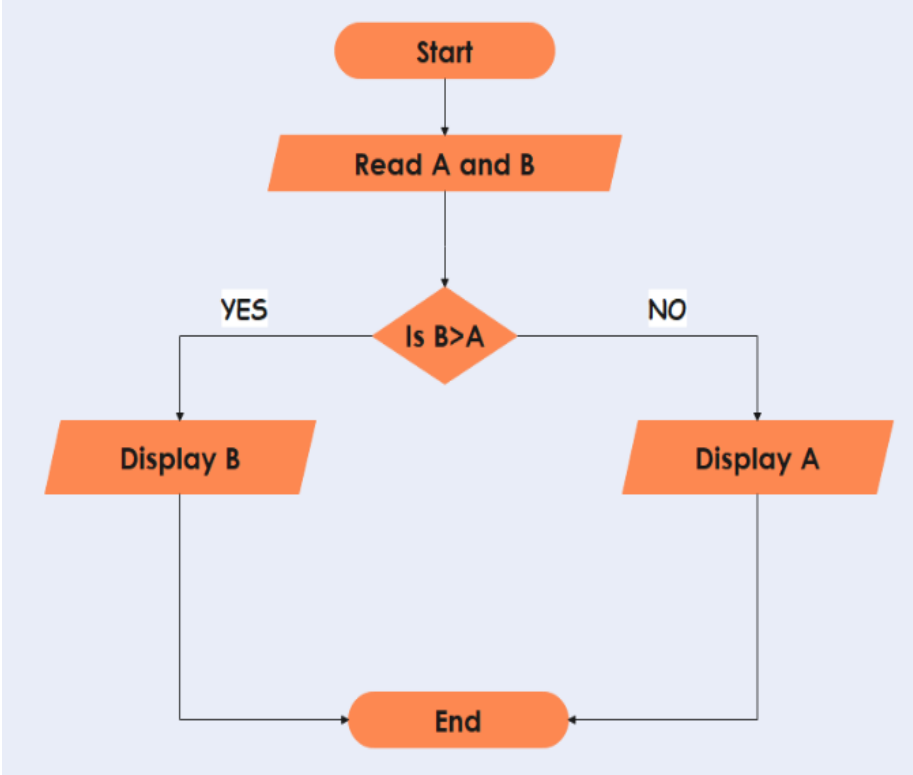
Example 1: Find the Sum of Two Numbers Entered

- Step 1: Read the Integer A.
- Step 2: Read Integer B.
- Step 3: Perform the addition by using the formula: C= A + B.
- Step 4: Print the Integer C.



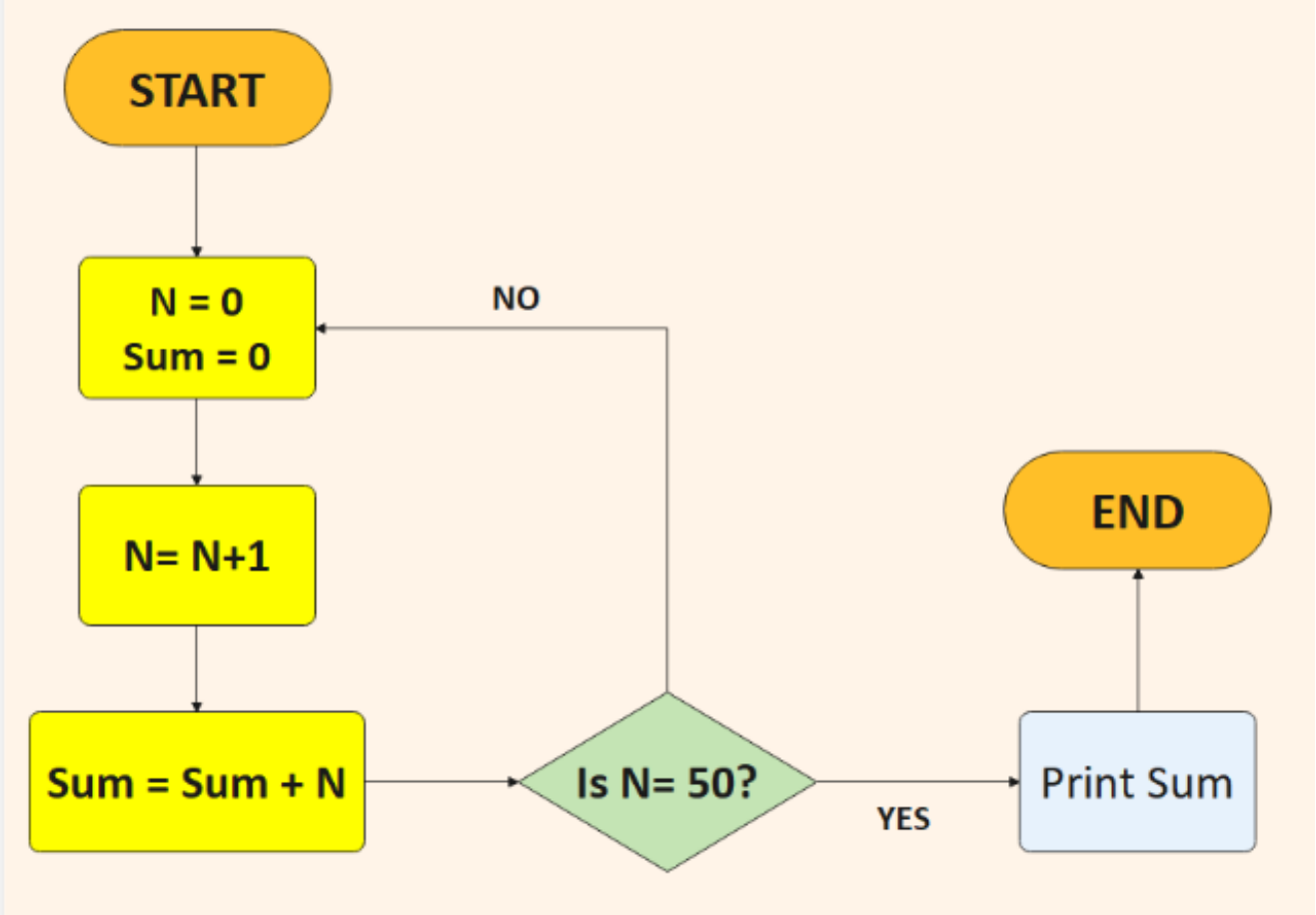
Example 2: Determining the Largest Number Among All the Entered Integers

- Step 1: Read the Integer A.
- Step 2: Read Integer B.
- Step 3: If B is greater than A, then print B, else A.



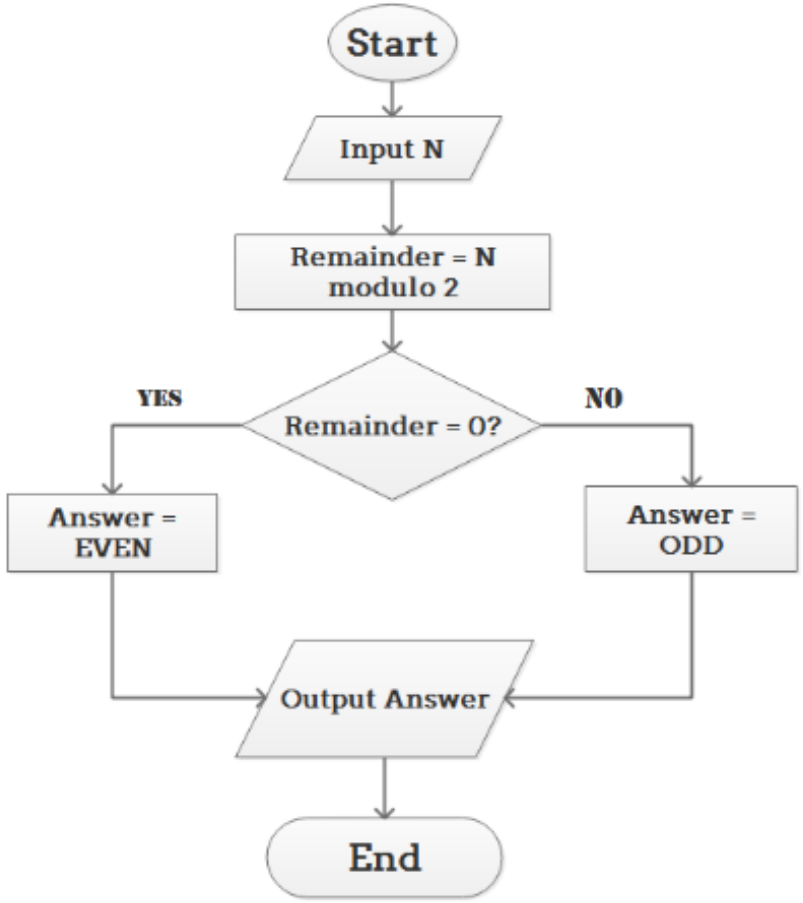
Example 5: Calculate the Sum of The First 50 Numbers

- Step 1: Declare number N= 0 and sum= 0
- Step 2: Determine N by N= N+1
- Step 3: Calculate the sum by the formula: Sum= N + Sum.
- Step 4: Add a loop between steps 2 and 3 until N= 50.
- Step 5: Print Sum.



Example 9: Determine and Output Whether Number N is Even or Odd

- Step 1: Read number N.
- Step 2: Set remainder as N modulo 2.
- Step 3: If the remainder is equal to 0, then number N is even, else number N is odd.
- Step 4: Print output.


Start
Input N
Remainder = N modulo 2
Remainder = 0?
YES
NO
Answer = EVEN
Answer = ODD
Output Answer
End

Example 10: Calculate the Interest of a Bank Deposit

- Step 1: Read amount.
- Step 2: Read years.
- Step 3: Read rate.
- Step 4: Calculate the interest with the formula "Interest=Amount*Years*Rate/100.
- Step 5: Print interest.