

**DATA: -**

Data is defined as the description of basic facts about the activities of a business.  
Examples: - Name, RollNo, date, cost, Basic Pay etc

**INFORMATION: -**

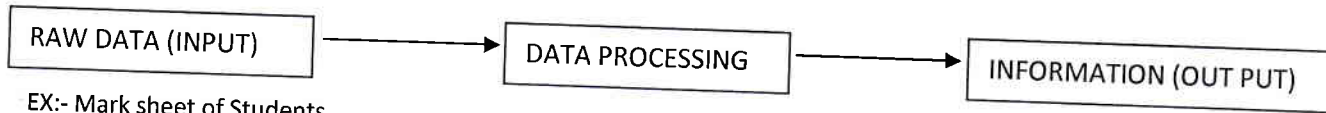
It is the set of Data which has been converted or organized into a more useful form.

Example: - Payslip, Marksheet etc

**DATA PROCESSING: -**

It is defined as sequence of operations on data to convert into a more useful form or information. It includes classification and transfer of data from one place to another.

Otherwise it can be defined as the process of collecting all items of data together to produce meaningful information. Data processing is mostly associated with business and commercial work.



EX:- Mark sheet of Students

**METHODS OF DATAPROCESSING**

Data Processing can be accomplished through following method.

1. MANUAL DATA PROCESSING.
2. SEMI-MANUAL DATA PROCESSING.
3. MECHANICAL DATA PROCESSING.
4. ELECTRONIC DATA PROCESSING.

**MANUAL DATA PROCESSING.**

In this type of Data Processing data is processed and transferred manually from one place to another. It deals with Abacus, Slide-Rule, Logarithm table etc. This method of Data Processing is very slow and an error may occur in the output.

**SEMI-MANUAL DATA PROCESSING.**

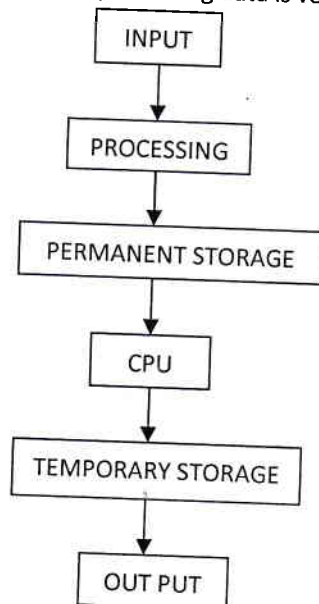
This method of Data Processing involves the use of calculator, totaling machines etc. This type of Data Processing is slightly faster than manual Data Processing and less error occur in output in compare to manual processing.

**MECHANICAL DATA PROCESSING.** This method of Data Processing used devices like printing press, Milo Meter and facet calculator.

**ELECTRONIC DATA PROCESSING**

In this method, processing is done by using some modern Electronic device like computers. So computer is known as Electronic Data Processing machine or EDP.

This method of processing data is very fast and output obtained is accurate.



## MODES OF DATA PROCESSING

1. BATCH PROCESSING
2. ONLINE PROCESSING.

### BATCH PROCESSING

It is processing of data where data is collected as input in batches and results are obtained periodically. Data collection is usually done off line.

Ex: - Pay Roll.

### ONLINE

It is a Data processing system where the result of data processing transactions is available immediately. The transaction of processing, file updating and data collection processes are also handled simultaneously by the online application system.

Ex: - Banking system

Stock Exchange

Airlines reservation system.

### DATA STORAGE

It is the holding of data in electromagnetic for access by the computer. Mainly there are two kind of data storage.

- i) Primary Storage
- ii) Secondary Storage

Primary storage is the data that is stored in RAM & other memory devices Secondary storages data is stored on external storage device such as hard disk, tape, CD etc.

### DATA RETRIEVAL

Data retrieval is process of identifying and extracting data from a data base, based on a query provided by the user or applications. It enables the fetching of data from a data base in order to display on a monitor or use within an application.

# CHAPTER –5: PROBLEM SOLVING METHODOLOGY

## Problem solving

Solving problems is the core of computer science. Programmers must first understand how a human solves a problem, then understand how to translate this "algorithm" into something a computer can do, and finally how to "write" the specific syntax (required by a computer) to get the job done. It is sometimes the case that a machine will solve a problem in a completely different way than a human.

Computer Programmers are problem solvers. In order to solve a problem on a computer you must:

1. Know how to represent the information (data) describing the problem.
2. Determine the steps to transform the information from one representation into another

## Algorithm, Pseudo code and Flowchart

### Algorithm

An algorithm is a set of specific steps to solve a problem. Think of it this way: if you were to tell your 3 year old niece to play your favorite song on the piano (assuming the niece has never played a piano), you would have to tell her where the piano was, and how to sit on the bench, and how to open the cover, and which keys to press, and which order to press them in, etc, etc, etc.

#### Definition:

- An algorithm is a well-defined procedure that allows a computer to solve a problem.
- Algorithm is defined as the step-by-step solution of problem in user's language.
- It is considered as an effective procedure for solving a problem in finite number of steps.
- Another way to describe an algorithm is a sequence of unambiguous instructions.
- In fact, it is difficult to think of a task performed by your computer that does not use algorithms.

The characteristics of Algorithm are

- Precise
- Unambiguous
- Finite termination
- Unique solution

Example:

1. Algorithm to find out sum of two numbers to be taken as input.

Step-1 Read the 1st number x Step-2 Read the 2nd number y Step-3 Sum=x+y  
Step-4 Print Sum

This is an example where only sequence is exhibited

2. Algorithm to find out larger between two numbers to be taken as input.

Step-1 Read the 1<sup>st</sup> number x Step-2 Read the 2<sup>nd</sup> number y Step-3 If  $x > y$

Then Print x Else if  $x < y$  Then Print y

Else Print —Both are Equal —

This is an example where Branching is exhibited

3. Algorithm to find out sum of first 10 natural numbers. Step-1  $i=1$ , Sum=0

Step-2 Repeat step 3 and 4 while  $i < 10$  Step-3 Sum= Sum+i

Step-4  $i=i+1$  Step-5 Print Sum

This is an example where Repetition is exhibited

## Pseudocode

It is a concise description algorithm in English language that uses programming language constructs. It contains outlines of the program that can be easily converted to program. It focuses on the logic of the algorithm without giving stress on the syntax of programming language. This is meant for understanding the logic of the program easily. Flowchart can be considered as an alternative to pseudo code. Several constructs/key words of programming language can be used in the algorithm to write the pseudo code.

Some of them are

If ... Endif

Do while ... end do While ... end while Repeat ... until

For ... end for Case end case Call

Return

## Flowchart

Flowchart is a graphical or symbolic representation of the process of solution to a problem or algorithm. It helps to visualize the complex logic of the solution of the problem in a simplified manner through diagrammatic representation. Each step of the algorithm is presented using a symbol and a short description. The different symbols used for the flowchart are

<b>S y m b o l</b>	<b>Purpose</b>	<b>Description</b>
	Flow line	Indicates the flow of logic by connecting symbols.
	Terminal(Stop /Start)	Represents the start and the end of a flowchart.
	Input/output	Used for input and output operation.
	Processing	Used for arithmetic operations and data-manipulations.
	Decision	Used for decision making between two or more alternatives.
	On-page Connector	Used to join different flowline
	Off-page Connector	Used to connect the flowchart portion on a different page.
	Predefined Process/Function	Represents a group of statements performing one processing task.

# Generation of Programming Languages

## Programming Language

Programming language is a tool to express the logic or instructions for understanding of the computer. Any programming language has two components:

- a) Syntax
- b) Semantics

Syntax refers to the rules to be followed for writing valid program statements. Compiler can detect errors in syntax while compiling the program. Semantics is associated with logic of the program. Compiler cannot detect the semantic error. The user of programmer can diagnose semantic error.

There are good numbers of High level languages, each meant for specific area of data processing. Commonly known languages are BASIC, FORTRAN, COBOL, Pascal, C, C++ etc. While FORTRAN is good for Numerical and scientific calculation, COBOL is good for Business applications involving large amount of data handling.

## Generations of Programming Language

The Programming languages can be classified into 4 generations:

1st Generation: Machine Language 2nd Generation: Assembly Language 3rd Generation:

High Level Language 4th Generation: Very High Level Language

Machine Level language contains instructions in binary form i.e. in 0s and 1s. Thus writing instruction was very difficult and needs heavy expertise. This was used in early days computers.

Assembly level language instructions were written using symbolic codes known as mnemonics. In comparison to Machine language, it is relatively easier to write program, but still it requires lot of expertise. A translator called assembler is used to translate assembly language program to machine level language.

High level language contains instructions in English like words so that user will feel easier to formulate and write the logical statements of the program. Here the logic may spread over multiple statements as against a single statement in assembly language. It uses a translator called compiler for translation of High level language program to machine level language program. There are many High level languages used for programming such as BASIC, FORTRAN, COBOL, PASCAL, C, C++ etc.

Very High Level language otherwise called as 4GL uses nonprocedural logical statements.

A typical example of 4GL is the query language such as SQL.

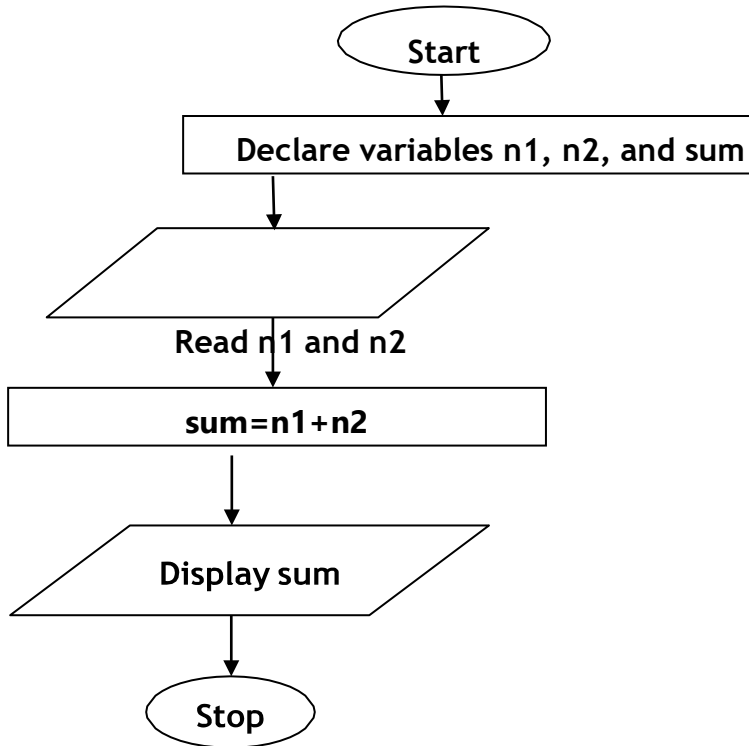
## Structured Programming Language

Structured Programming is also known as Modular Programming. In this type of programming technique, the program shall be broken into several modules. This helps in managing memory efficiently as the required module of the program will be loaded into the memory only and not the entire program. This will also enhance code reuse. Writing, understanding, debugging and modifying the individual module of the program is also easier.

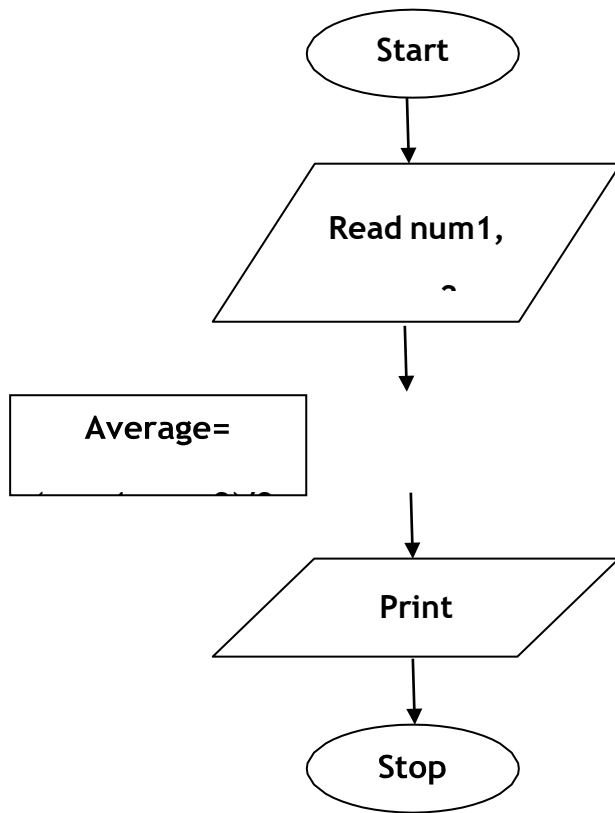
### Examples of Problem solving through Flowchart

#### Example

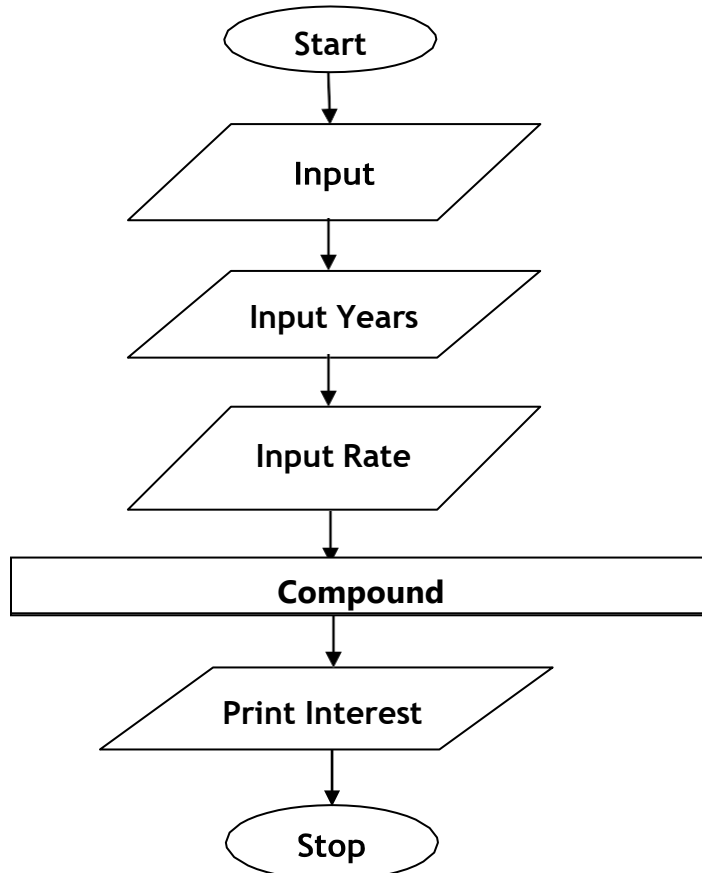
1. Add two numbers entered by the user.



2. Flowchart to calculate the average of two numbers

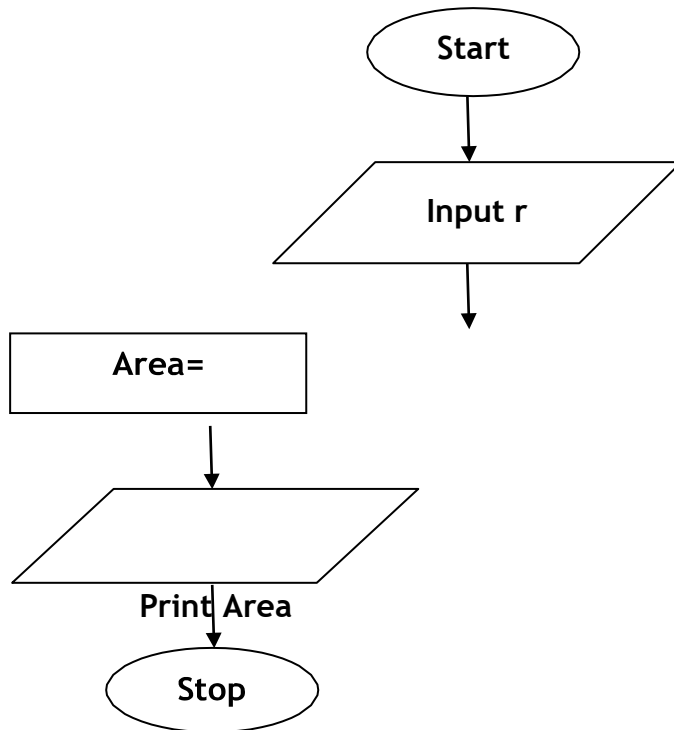


3. Flowchart to Calculate the Interest of a Bank Deposit

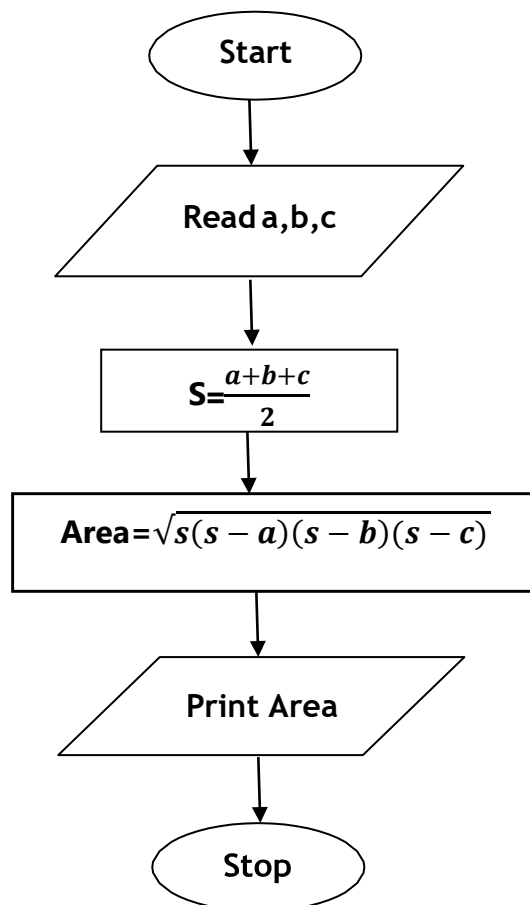




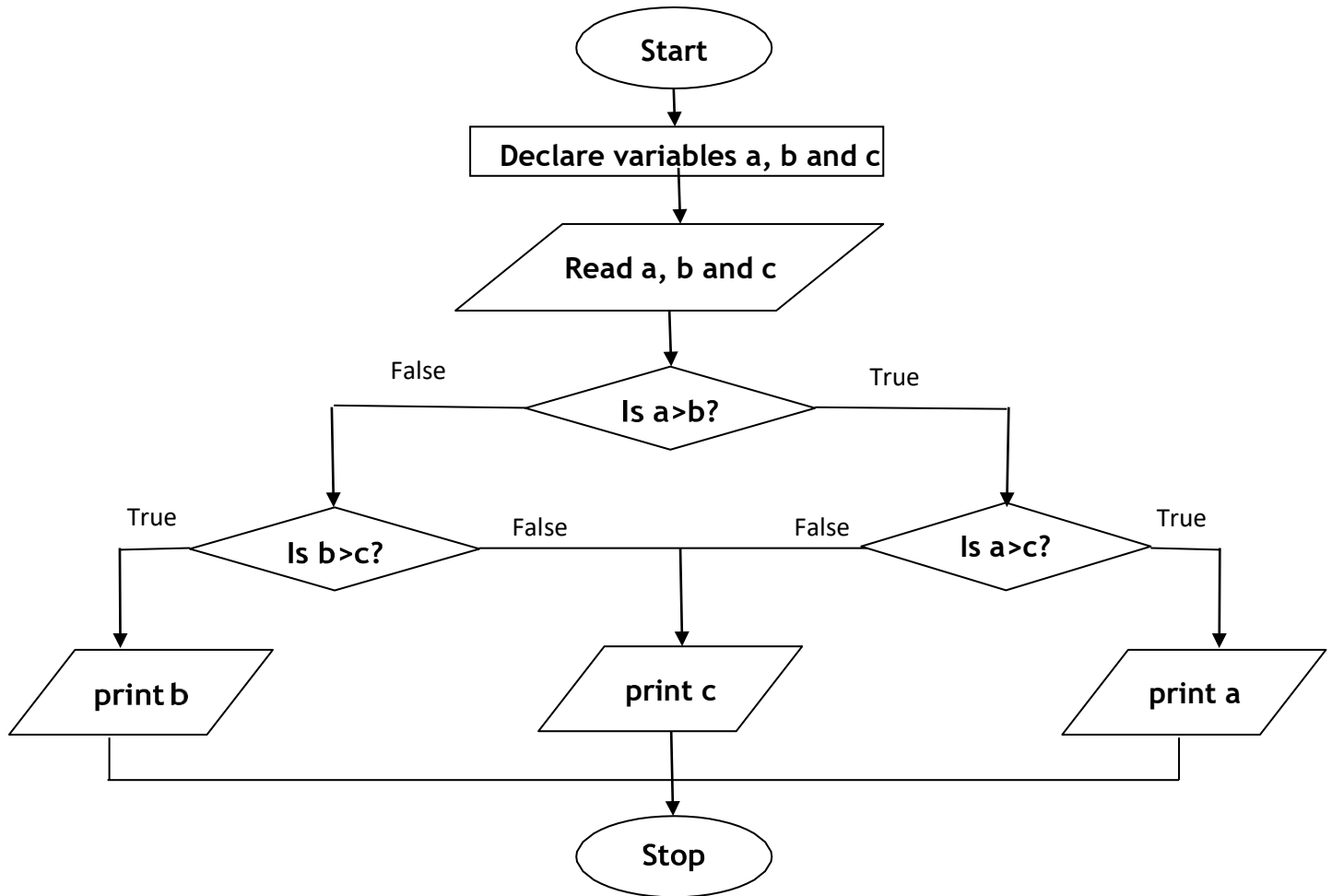
4. Flowchart to calculate the area of a circle.



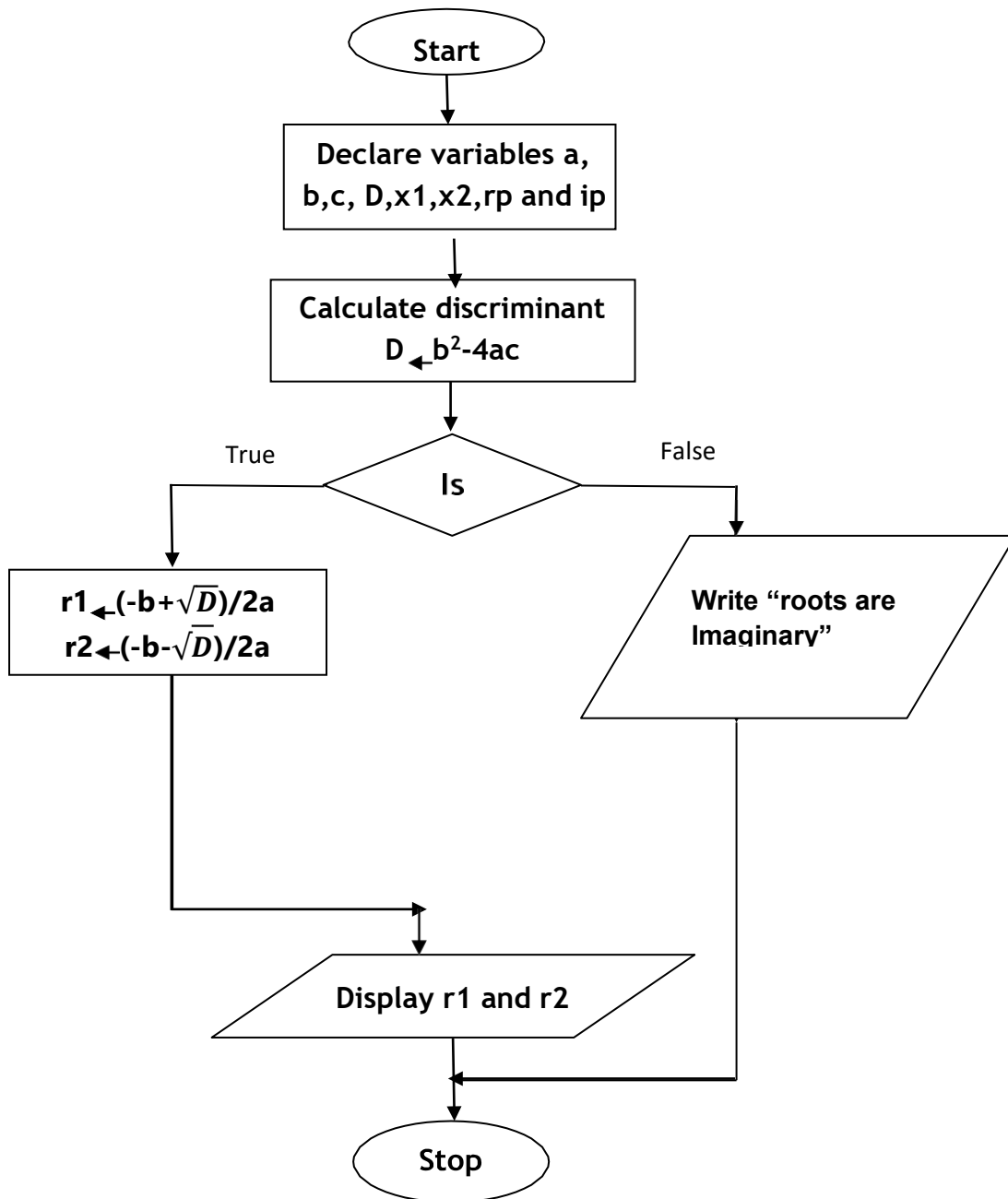
5. Flowchart to calculate the area of a triangle.



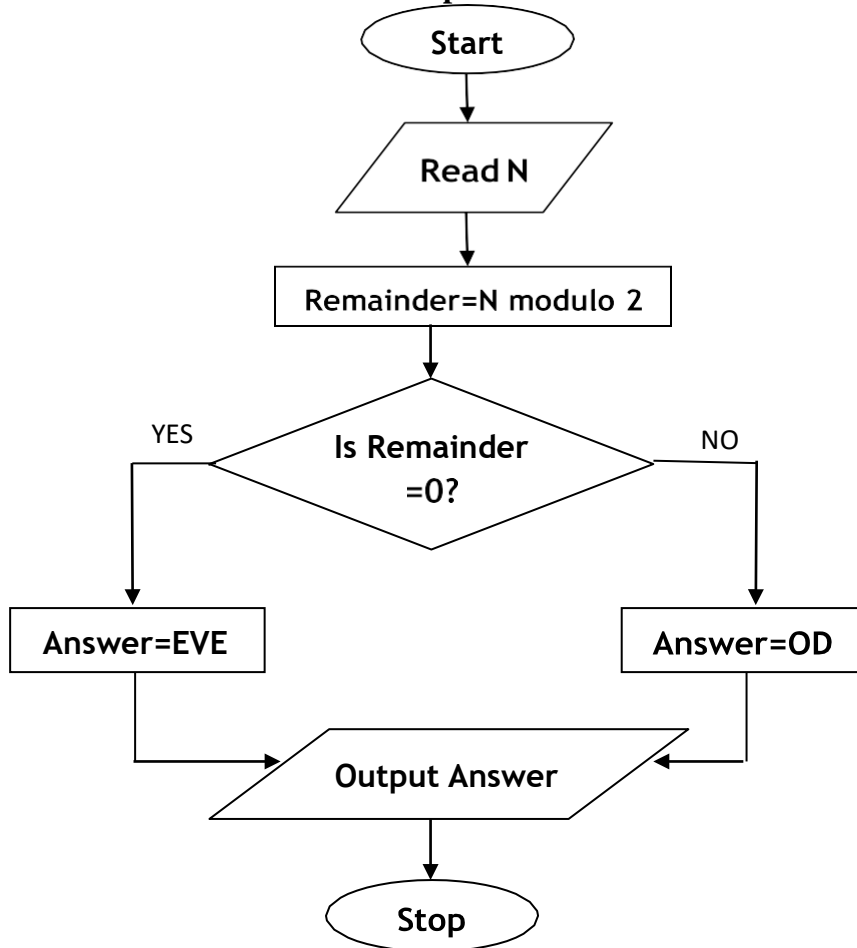
6. Find the largest among three different numbers entered by the user.



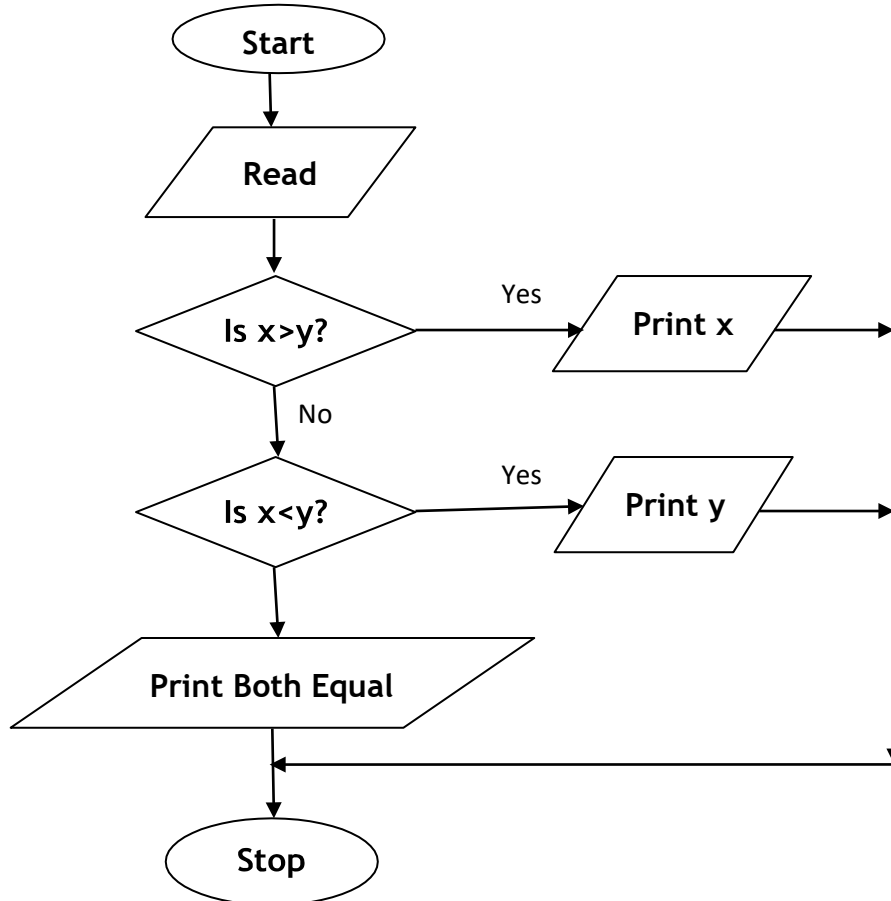
7. Find all the roots of a quadratic equation  $ax^2+bx+c=0$



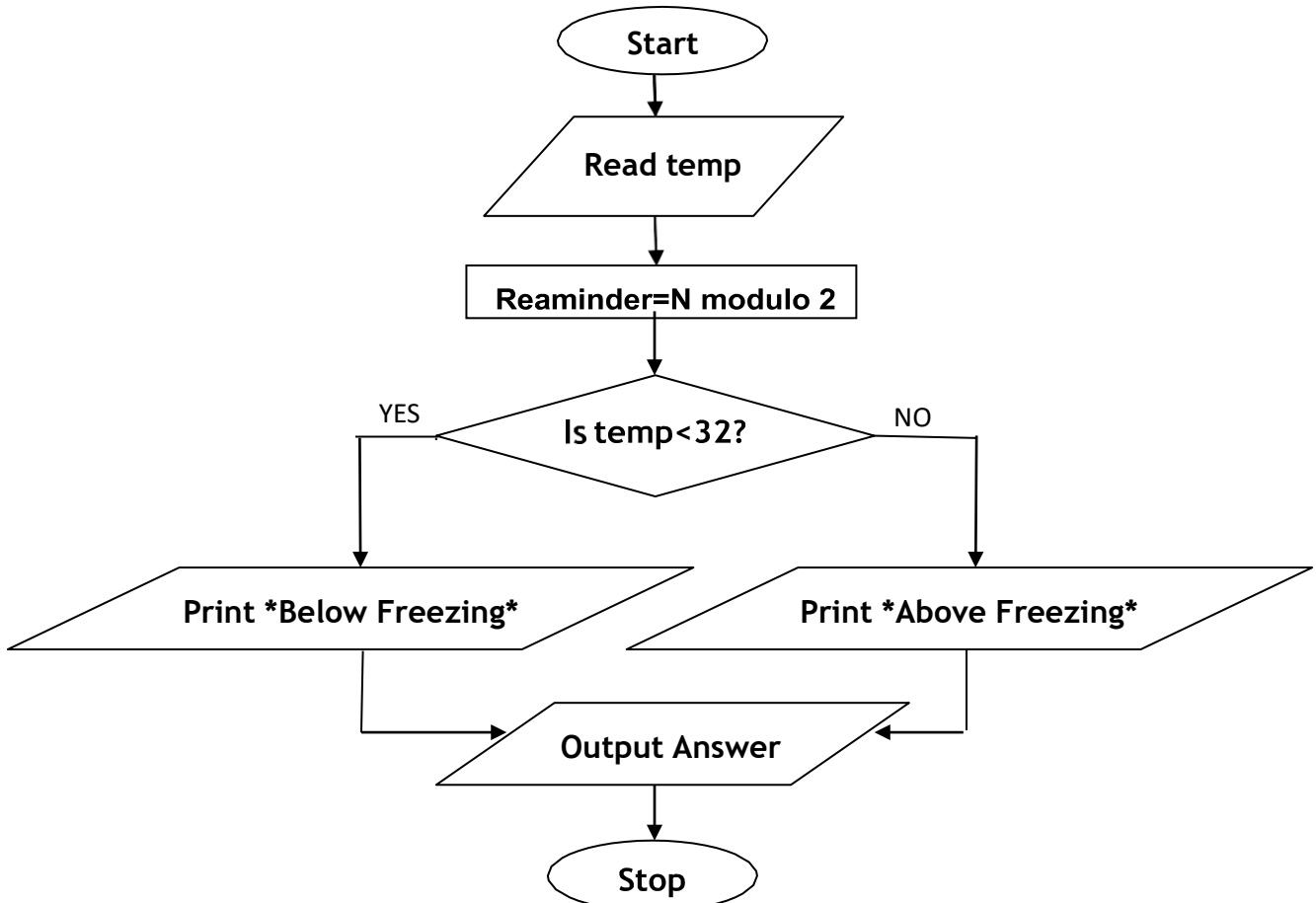
8. Flowchart to Determine and Output Whether Number N is Even or Odd



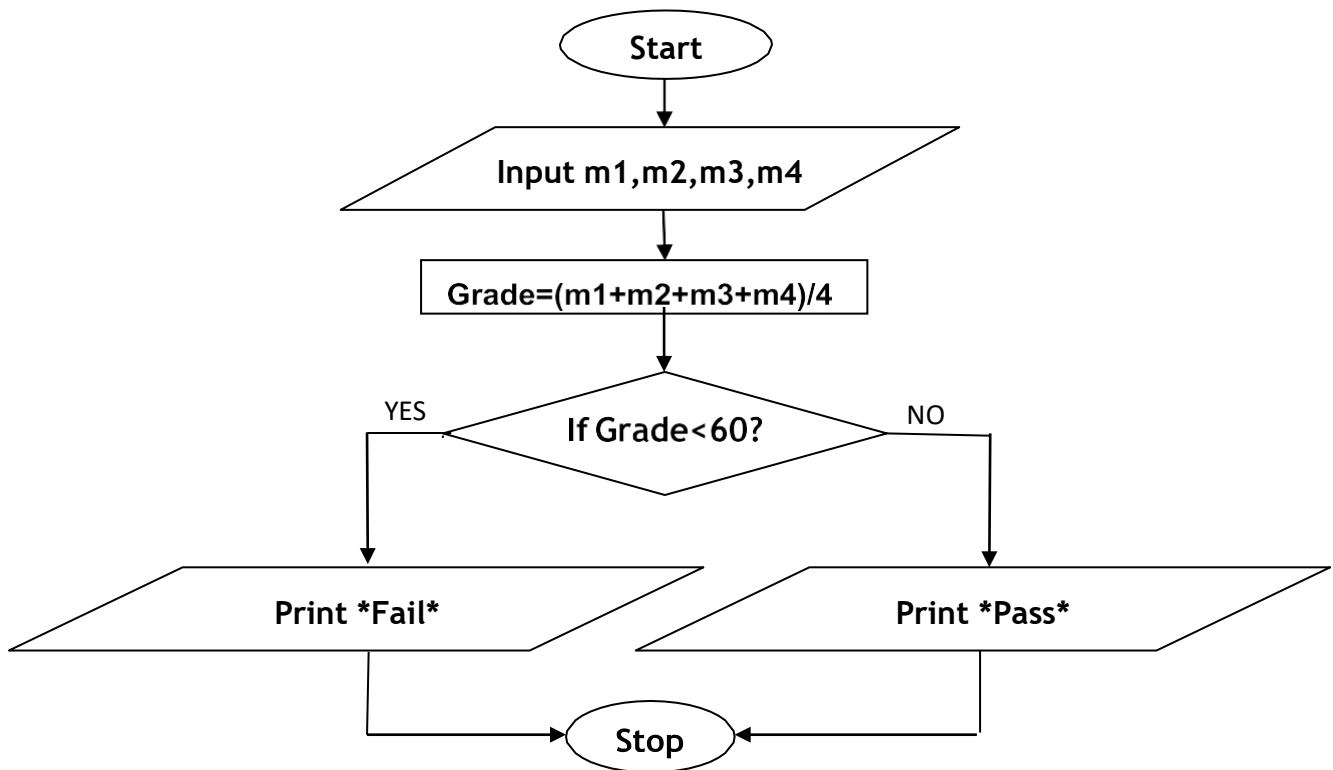
9. Flowchart to find out larger between two numbers to be taken as input



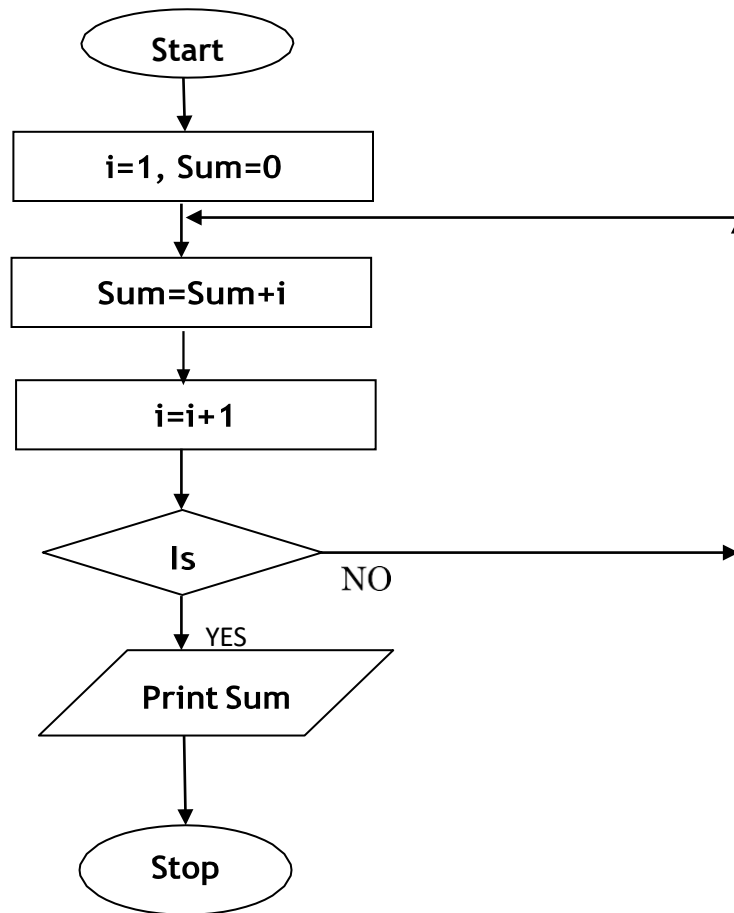
10. Flowchart to Determine Whether a Temperature is Below or Above the Freezing Point



11. Flowchart to Determine Whether A Student Passed the Exam or Not



12. Flowchart to find out sum of first 10 natural numbers



# C - PROGRAMMING

Def:- "C" is a general-purpose procedural programming language. It was initially developed by Dennis Ritchie at Bell Laboratory during the year 1972.

It was mainly used to develop software like operating system, database and compilers.

It is an excellent language to learn to program for beginners.

## C - introduction.

- ① Keywords and Identifiers
- ② Variables and constants.
- ③ C - Data Types
- ④ C - INPUT/OUTPUT
- ⑤ C - OPERATORS

## Character set

:- A character set is a set of alphabets, letters, and some special character that are valid in C language.

# Alphabets

Upper case - A, B, C, ..., Z

Lower case - a, b, c, ..., z

digits - 0 to 9

special character -  $\langle$ ,  $\rangle$ ,  $($ ,  $)$ ,  $\{$ ,  $\}$   
etc

C accepts both lower case and uppercase alphabets as variables and function.

## C - keywords

Keywords are predefined reserved words used in programming that have special meanings to the compiler.

As "C" is a case sensitive language, so all the keywords must be written in lower case.

## Example of keywords

auto, break, case, char, continue, do,

const, double, else, for, if, goto, float,

int etc.



## C- Identifiers :- Identifiers refers to name

given to entities such as variables, function etc. Identifiers are created to give a unique name to an entity to identify it during the execution of program.

### Example

~~at~~ money, sum

Here money, sum are identifiers

Also remember identifiers names must be different from keywords.

Hence you can not use "int" as an identifier because "int" is a keyword as discussed earlier.

## Rules for naming Identifiers :-

① A valid identifier can have letters (both uppercase and lowercase), digits.

and underscore (-)

② The first letter of an identifier should be either a letter or an underscore.

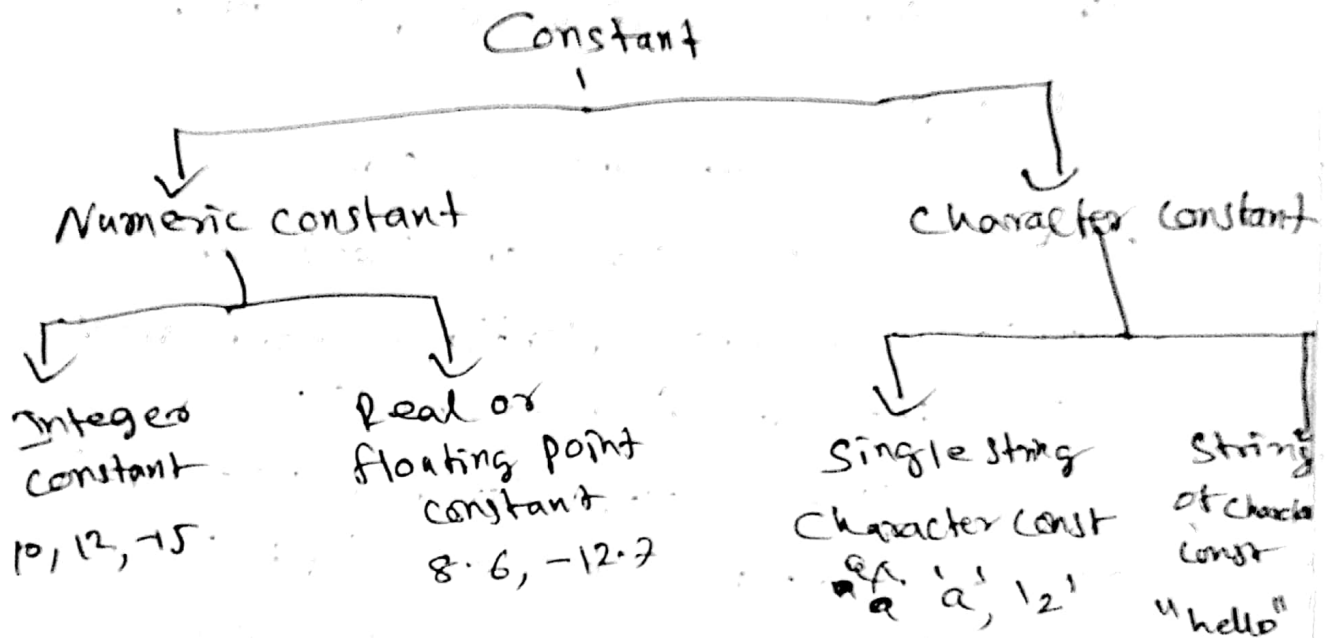
③ Keywords can not be used as identifiers.

④ Can not be longer than 31 characters but it is a good practice to be shorter length.

Example - sum, b-pay, etc.

# CONSTANT

Constants in "C" are the fixed values that are used in program and its value does not change during the entire execution of program.



## Rules

### Integer constant

- ① An integer constant must be at least one digit.
- ② It must not have a decimal point.
- ③ It can either be positive or negative.
- ④ no comma or blanks are allowed within an integer constant.  
Ex - 27, -15 etc.

(4)

## Real or Floating point constant

- ① A real or floating point constant must at least contain a decimal point
- ② It can either be positive or negative
- ③ no commas or blanks are allowed within a real constant.

Ex - 18.93, -15.65 etc.

## Single String Character constant

- ① any single character enclosed within a single quotation ('')

Ex - 'a', '2' etc.

## String of character const.

- ① String of characters enclosed within double quotation ("")

Ex - "hello", "yes"

## VARIABLE

A variable is any characteristic, number or quantity that can be measured or counted. A variable is also called as data item.

Ex - age, sex, sam etc

(5)

Variable declaration :- In "C" programming

Variables which are to be used in different parts of the function have to be declared.

In program variable declaration tells the computer two things -

① The name of the variables.

② The type of data the variable will hold.

Type Declaration :- C-DATA TYPES

<u>Basic data types</u>	<u>Byte required</u>
-------------------------	----------------------

① int	2 byte
-------	--------

② float	4 byte
---------	--------

③ char	1 byte
--------	--------

Example

```
int sum;
```

```
float average, area;
```

```
char name;
```

INPUT/OUTPUT STATEMENT :-

Input means to provide the program with some data to be used in the program.

61

## Example

scanf

output

output means to display data on screen or write the data to a printer or a file.

Ex -

printf

Structure of C program

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
main()
```

```
{ → Beginning of program
```

Type declaration

```
----- } C-statement  
-----  
-----
```

```
getch();
```

```
} → End of C program.
```

(7)

## Assignment

### Example-I

write a program in C to add two numbers

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
main()
```

```
{  
    int a, b, c;
```

```
    printf("Enter the first number");
```

```
    scanf("%d", &a);
```

```
    printf("Enter the second number");
```

```
    scanf("%d", &b);
```

```
    c = a + b;
```

```
    printf("sum of two numbers %d", c);
```

```
    getch();
```

```
}
```

K. DAS  
raise

ERN attend

Example-2

Assignment

Write a program in C. to calculate area of a circle.

```
#include <stdio.h>
#include <conio.h>
main()
{
    int r;
    float a;
    printf("Enter the value of radius");
    scanf("%d", &r);
    a = 3.14 * r * r;
    printf("Area is = %f", a);
    getch();
}
```