

PROGRAMMING IN “C”

Bachelor of Computer Application
BCA

The logo of Sahyog Pratishtha is a circular emblem. It features a globe in the center with a grid of latitude and longitude lines. The text "SAHYOG PRATISHTHA" is written in a circular path around the globe. Below the globe, the words "Sahyog", "Knowledge", and "Career" are written in a smaller font.

SEMESTER-I

Prepared by,
Ms. Deepa Mishra
MCA

Chapter-01

Principles of Programming

Remember your last visit to the Super Market. You might have purchased many items. What did you do after picking all the items and putting them into the carriage? Probably you met with the billing clerk to make the payment. Have you observed his way of preparing the bill? He picks the items one by one and enters something into the computer repeating the same task for all the items. Within a count of few minutes or even may be within few seconds he gives the bill, you pay and come out carrying all the required things. So, what made him to process the bill so fast? It is nothing but the **“program”** running in the computer’s memory.

Program:

- A computer program is a collection of instructions that can be executed by a computer to perform a specific task.
- Computer program is usually written by a computer programmer in a programming language.

Programmer:

- One who writes the programs/instructions for the computer is known as Programmer.

Programming:

- The act of writing computer programs using some programming language is called computer programming.

Programming Language:

- Programming language is a language which is used to communicate with the computers i.e. it is a language using which programmers create programs/instructions for the computers so that computer can perform some actions.
- Computers do not understand Hindi, Marathi, English, Gujarati etc. they only understand programming language.
- Like, we human prefer our native language Hindi, Marathi etc. for communication purpose, similarly computers prefer programming languages.

Use of Programming Language:

- Programming Languages are used for making softwares/Applications.

Types of Softwares:

1. System Software:

- “Operating System” is an example of system software.
- Some of the key examples of operating systems are: MS Windows, Linux, Ubuntu, Unix etc.

2. Application Software:

- Word Processors(MS-WORD,GOOGLE DOCS)
- Database Software(MS-ACCESS)
- Multimedia Software (VLC Player, Media Player) are the few examples of Application Software.

Types of Programming Language:

1. Machine Language
2. Assembly Language
3. High Level Language

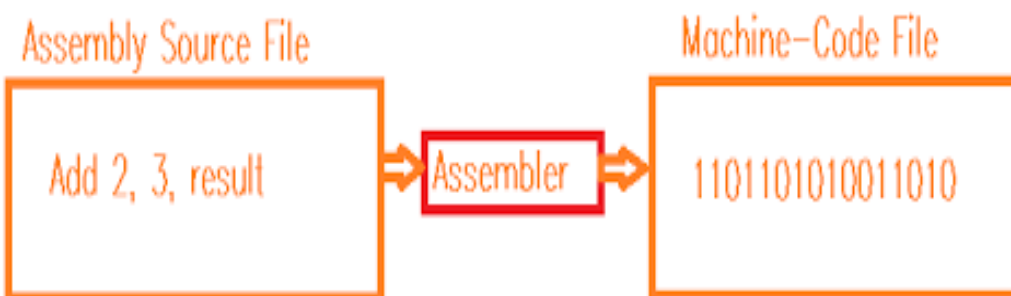
1. Machine Language:

- Machine language is the only language a computer is capable of understanding.
- Machine Language is represented in the sequence of 0's and 1's i.e. the instructions are given to computer in the form of 1 and 0 only.
- It is also known as binary language.
- It also comes under low level programming Language and difficult to understand.
- E.g.: 01100110 10101010



2. Assembly Language:

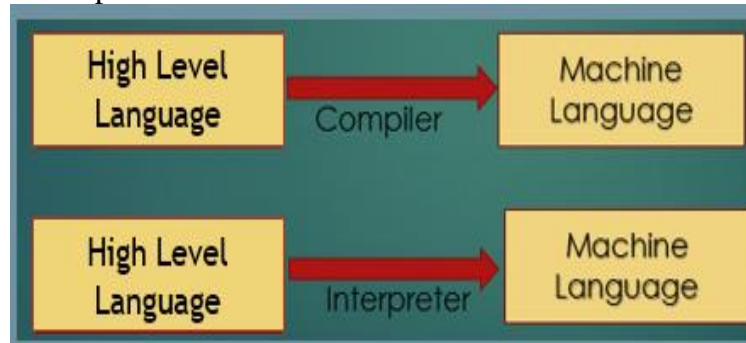
- Assembly Language is again a low level language.
- It uses “Mnemonics code” for the instructions.
- Mnemonics codes are the symbolic constants used for instructions.
- E.g. Add, Sub, Mov etc. are the mnemonics code.
- Assembly language is used in micro-controller based embedded systems.
- Example of embedded systems are AC, Washing Machine and Elevator etc.
- But, Assembly language is not understandable to computers because they only understand machine language.
- So the instructions/code written in assembly language are first converted into machine language, so that computer can understand it and perform some operation on it.
- The convertor/translator which is used for converting assembly language into machine language are called “Assembler”.



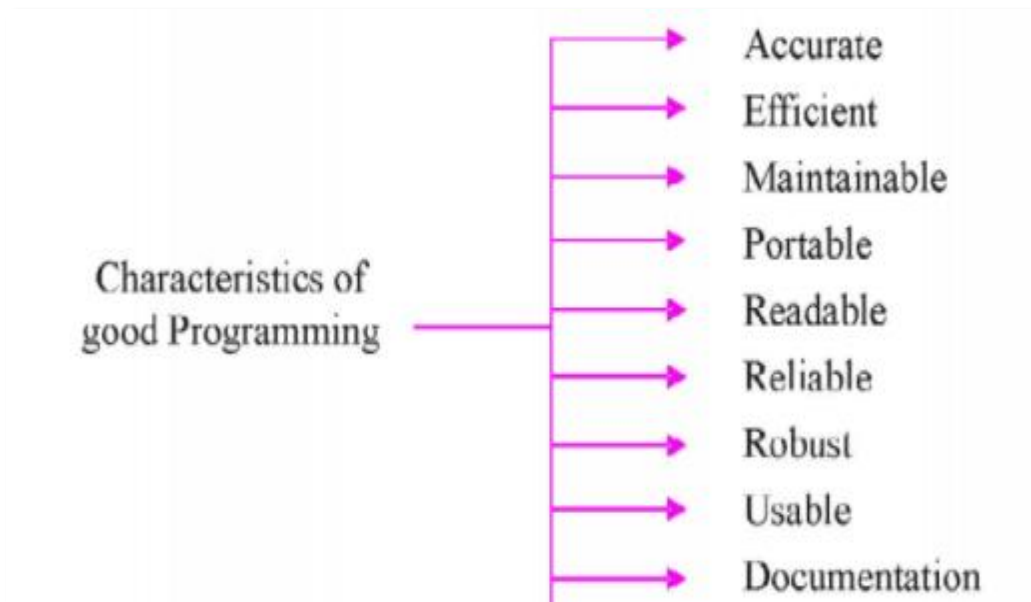
3. High Level Language:

- High level language is very much user friendly and easier for programmers to understand.
- It uses English-Like structure and basic mathematical syntax.
- Eg: 2+6, 10/2, printf(“Good Morning”); etc.
- As computer doesn't understand high level language so, the program written in high level language need to be converted into machine language so that computer can understand it.
- The translator/convertor which is used to convert high level language into machine language are called “Compiler” or “Interpreter”.
- C, C++, JAVA, C#, PYTHON, JAVASCRIPT, PHP, RUBY etc. are the example of high level language.

- C, C++ programming language uses compiler for translation.
- Python, Javascript ,Php uses interpreter for translation.



Characteristics of Programming:



1. Accurate:

The programming of a program is based on a certain problem. The problem must be pre-defined clearly with specification of requirements. It is expected from the programmer to design a program that strictly follows these requirements. So, the designed program must be accurate to perform the specified task. Such characteristic of programming is referred 'accurate'.

2. Efficient:

Every program utilizes the resources of the computer system. It is expected from the programming that the designed program utilizes these resources in an efficient manner. It means the program must not spend much time or over use the processor in executing its coded instructions. Such characteristic of the programming is referred 'efficient'.

3. Maintainable:

When proper structuring method is used in programming, the designed program can be made maintainable. Here maintainable means the ability to change as per the new needs. With very little modification a program should work for the new needs. Such characteristic of programming is referred 'maintainable'.

4. **Portable:**

It is expected from the designed program that it can be carried to any platform to solve the task. If the programming is done keeping many systems rather than one system in mind the designed programs are portable. Once the program is portable, it can be easily transferred from one machine to another. Such characteristic of programming is referred 'portable'.

5. **Readable:**

The program designed by the programmer must be self-readable as he is the first reader. Generally the designed program is also read by co-programmers or others. So, the designed program must contain proper comments to explain the coded instructions. This commenting will help in understanding and reading the program. Such characteristic of programming is referred 'readable'.

6. **Reliable:**

The designed program must perform as per the need all the time. It should also produce the intended results for any sort of inputs. In case of improper inputs, it should stop only after displaying proper error messages. These will indicate the cause of termination of the program. Such programs are created with the 'reliable' characteristic of the programming.

7. **Robust:**

The designed program is expected to continue with its functionalities even at the unexpected errors. It is the art of programming that takes care of all the possible errors before completing the design. Such programs keep on doing their work even at worst situations. Such characteristic of programming is referred 'robust'.

8. **Usable:**

The designed program must be easy to use. It must be designed with proper Interactive messages so that the user can easily get accustomed to it. Proper thinking in interface design will prove its worthiness. The documentation of the program must be prepared in good format to train the users. Such characteristic of the programming is referred 'usable'.

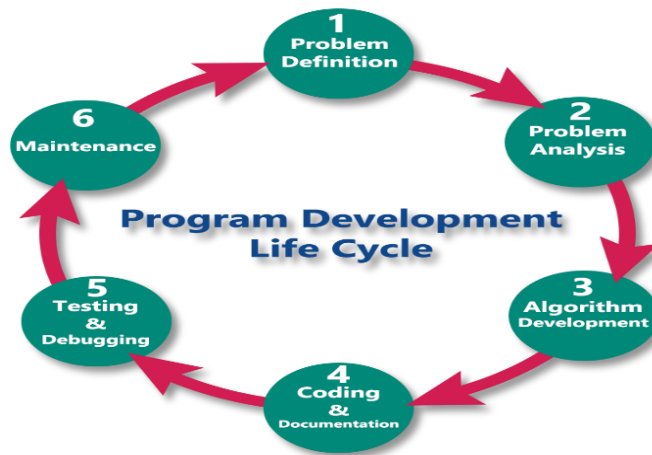
9. **Documentation:**

The usage of comments or remarks to explain the coded instructions and the modules of the program is called documenting the program. Such documented program will also help in providing other characteristic of the programming. The comments will help to find the errors as well as to rectify them. When the program is lengthy one proper documentation will properly connect the components of the program. It improves the readability as well as usability of the program. Such characteristic of programming is referred 'documentation'.

Stages of program Development:

- The process of designing the complete program from the scratch is called Program development.
- The program development is involved with many stages. The stages of the program development are also called phases or steps.

The following stages or steps are generally followed to develop a program:



1. Problem Definition:

In this phase we need to understand the problem statement, what is our requirement, what should be the output of the problem? These are defined in this first phase of the program development.

2. Problem Analysis:

In phase 2, we determine the requirements like variables, functions, etc. to solve the problem. That means we gather the required resources to solve the problem defined in the problem definition phase.

3. Algorithm Development:

During this phase, we develop a step by step procedure to solve the problem using the specification given in the previous phase. This phase is very important for program development. That means we write the solution in step by step statements.

4. Coding & Documentation:

This phase uses a programming language to write or implement the actual programming instructions for the steps defined in the previous phase. In this phase, we construct the actual program. That means we write the program to solve the given problem using programming languages like C, C++, Java, etc.

5. Testing & Debugging:

During this phase, we check whether the code written in the previous step is solving the specified problem or not. That means we test the program whether it is solving the problem for various input data values or not. We also test whether it is providing the desired output or not.

6. Maintenance:

During this phase, the program is actively used by the users. If any enhancements found in this phase, all the phases are to be repeated to make the enhancements. That means in this phase, the solution (program) is used by the end-user. If the user encounters any problem or wants any enhancement, then we need to repeat all the phases from the starting, so that the encountered problem is solved or enhancement is added.

Algorithm:

- An algorithm is a procedure or step-by-step instruction for solving a problem.
- They form the foundation of writing a program.

For writing any programs, the following has to be known:

1. Input
2. Tasks to be performed
3. Output expected

Examples of Algorithms in Programming:

1. Write an algorithm to add two numbers.

Step 1: Start

Step 2: Declare variables num1, num2 and sum.

Step 3: Read: num1 and num2.

Step 4: Add num1 and num2 and assign the result to sum.

Sum=num1+num2

Step 5: Display: sum

Step 6: Stop

2. Write an algorithm to calculate square of a number.

Step 1: Start

Step 2: Declare 2 variables num, result

Step 3: Read: num

Step 4: calculate square and assign the result to result variable

result=num * num

Step 5: Display: result

Step 6: Stop



3. Write an algorithm to display smallest no. between 2 nos.

Step 1: Start

Step 2: Declare 2 variables first, second

Step 3: Read: first and second

Step 4: check which one is the smallest

if (first < second)

Display: first

else

Display: second

Step 6: Stop

4. Write an algorithm to check eligibility for voting.

Step 1: Start

Step 2: Declare a variable: age

Step 3: Read: age

Step 4: check the age

if (age>=18)

Display “you are eligible”

else

Display “you are not eligible”

Step 6: Stop

5. Write an algorithm to check whether student is failed or passed.

Step 1: Start

Step 2: Declare a variable: marks

Step 3: Read: marks

Step 4: check the marks

if(marks \geq 35)

Display: “you are passed”

else

Display: “you are failed”

Step 6: Stop

VIDEO LINK: <https://youtu.be/g2q8cPxIIgE>

Flowchart:

- Flowchart is the graphical/diagrammatic representation of an algorithm.
- Programmers often use it as a program-planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information and processing.

Symbols Used In Flowchart:

1. Terminal:

- The oval symbol indicates Start, Stop and Halt in a program’s logic flow.
- Terminal is the first and last symbols in the flowchart.



2. Input/Output:

- A parallelogram denotes any function of input/output type.
- Program instructions that take input from input devices and display output on output devices are indicated with parallelogram in a flowchart.



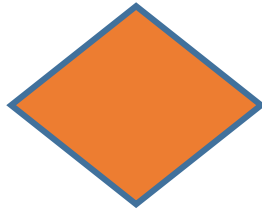
3. Processing:

- A box represents arithmetic instructions.
- All arithmetic processes such as addition, subtraction, multiplication and division are indicated by action or process symbol.



4. **Decision :**

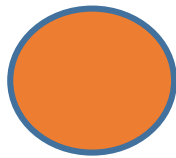
- Diamond symbol represents a decision point.
- Decision based operations such as yes/no question or true/false are indicated by diamond in flowchart.



5. **Connectors:**

A. On-Page Connector:

- Connects 2 or more parts of flowcharts which are on the same page.
- On-page connectors are referenced using numbers



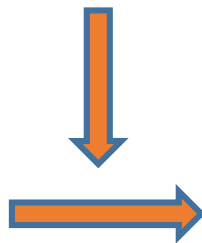
B. Off-Page Connector:








- Whenever flowchart becomes complex or it spreads over more than one page.
- Off-page connectors are referenced using alphabets
- It is useful to use connectors to avoid any confusions.



6. **Flow lines:**

- Flow lines indicate the exact sequence in which instructions are executed.
- It indicates the flow of logic by connecting symbols.

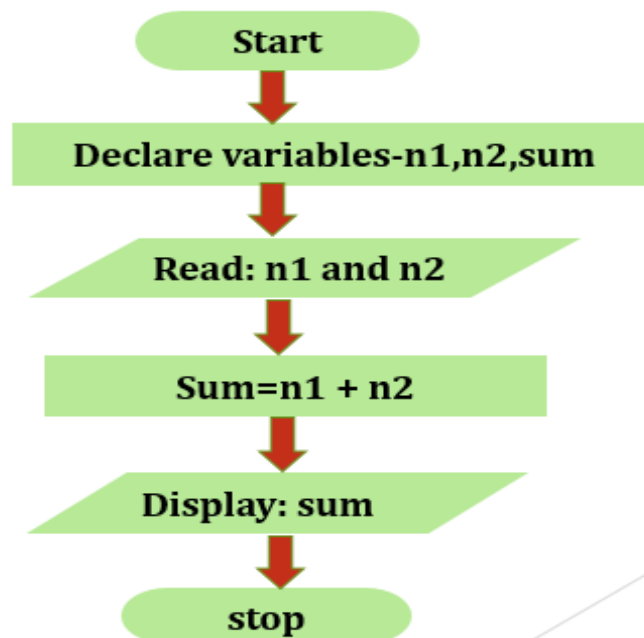


Symbol	Purpose	Description
	Flow Line	Indicates the flow of logic by connecting symbols.
	Terminal(Start or Stop)	Represents start and end of a flowchart
	Input/Output	Used for input and output operations.
	Processing	Used for arithmetic operations.
	Decision	Used for decision making
	On-page Connector	Connects 2 or more parts of flowcharts which are on the same page.
	Off-page Connector	Connects two parts of a flowchart which are spread over different pages.

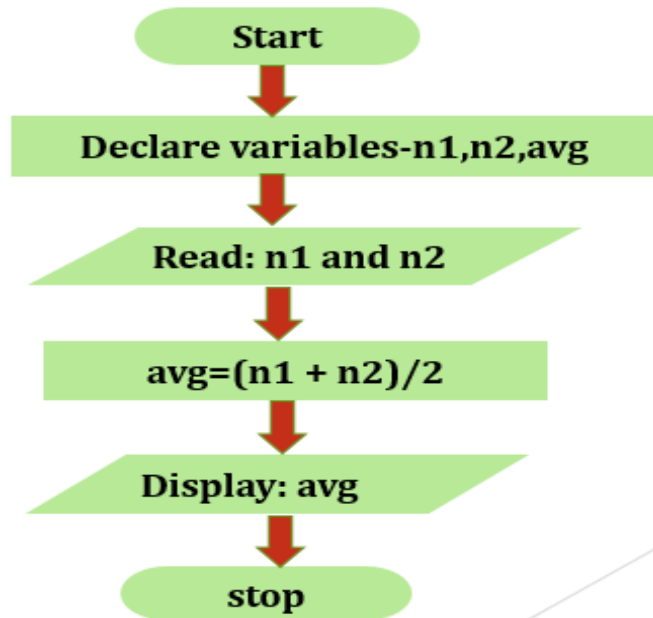
VIDEO LINK: <https://youtu.be/Blz3nSMzfi8>

Examples of flowcharts in programming:

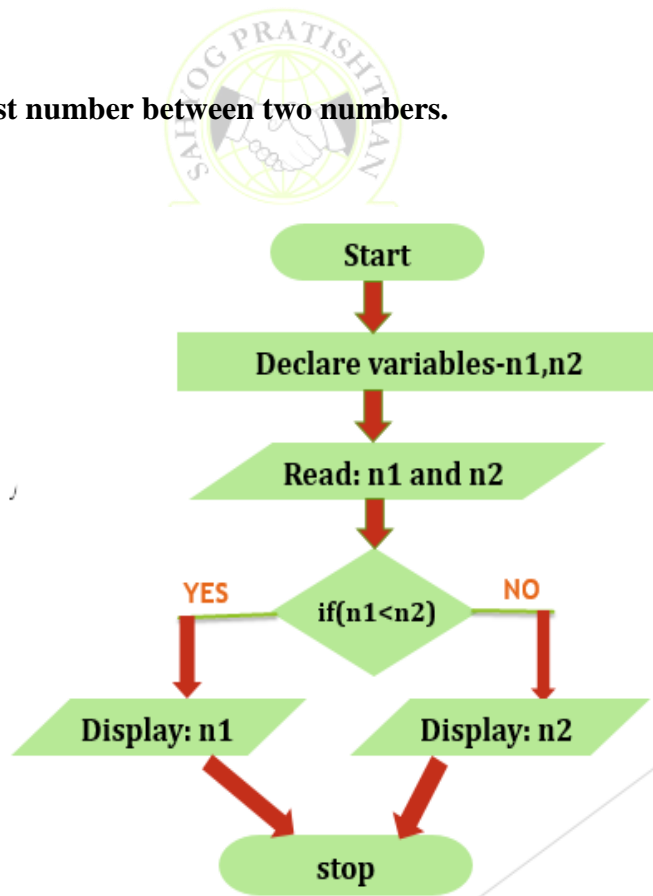
1. Design a flowchart to find the addition of two numbers.



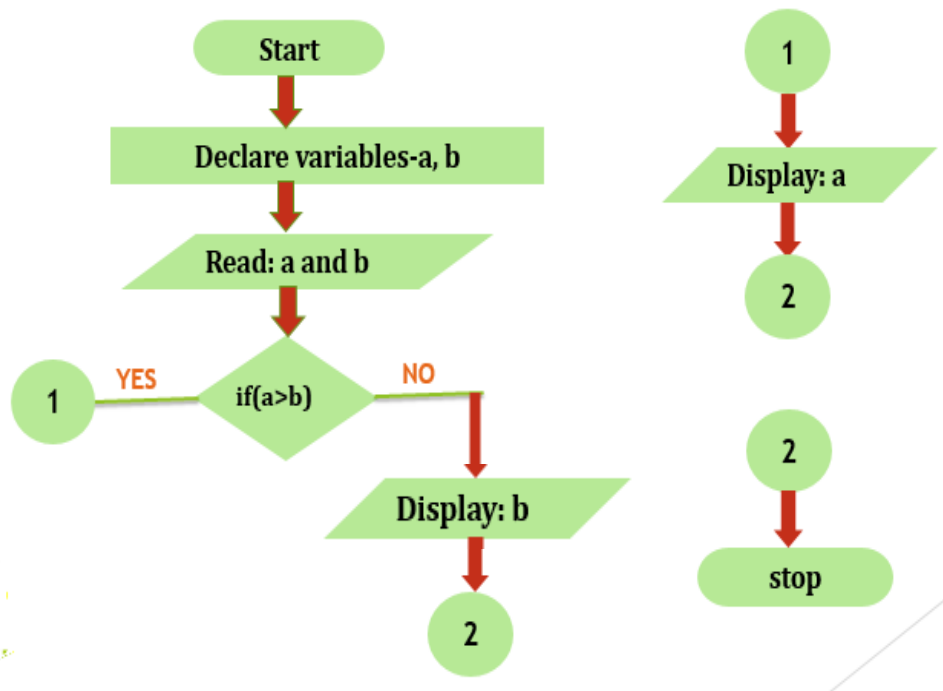
2. Flowchart to find the average of two numbers.



3. Flowchart to find the smallest number between two numbers.



4. Prepare a flowchart to check biggest number.(Use on page connector)



5. Prepare a flowchart to calculate square and cube of a number.(Using off page connector).

