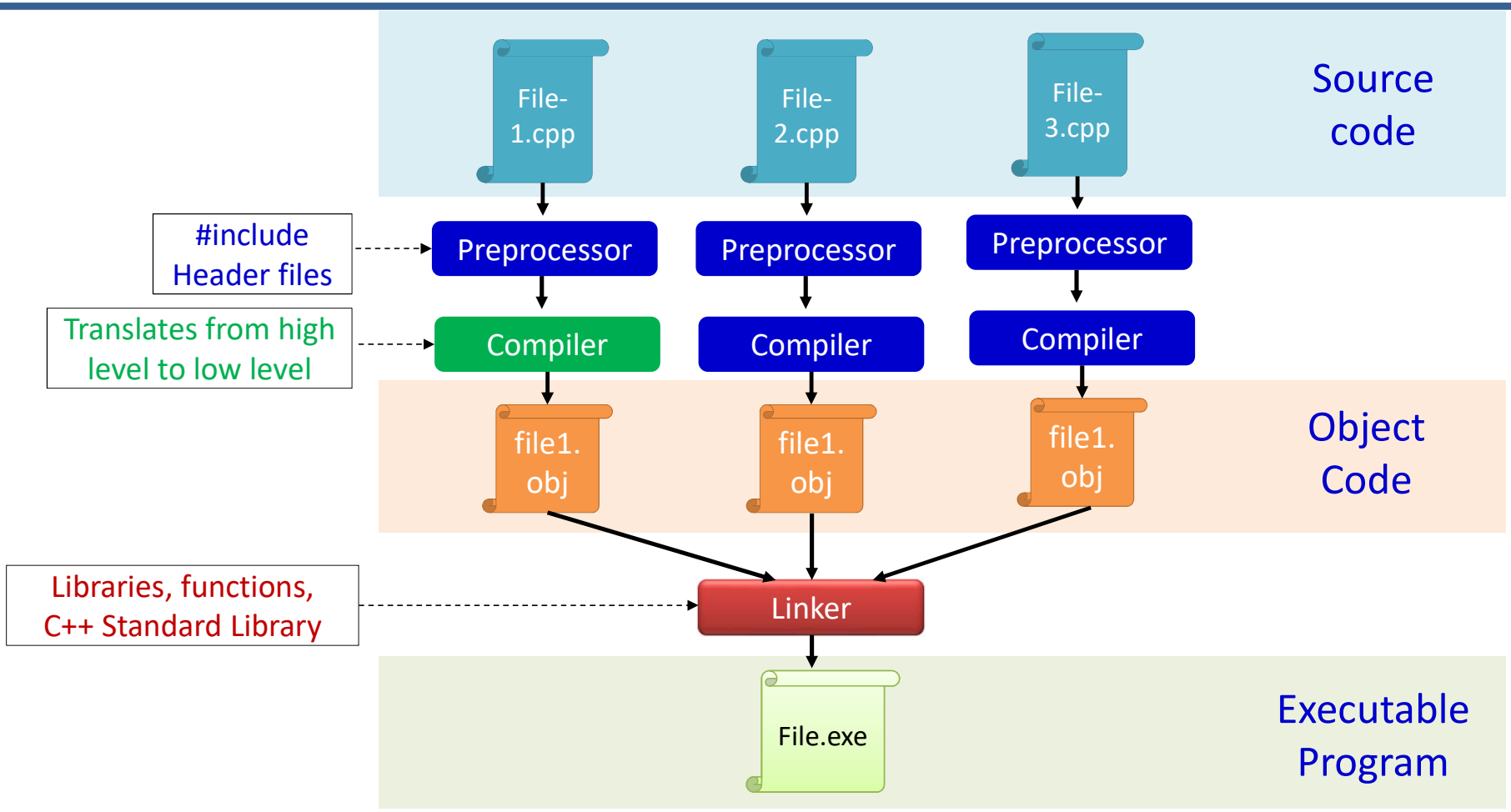
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

EEC-101
Programming with C++

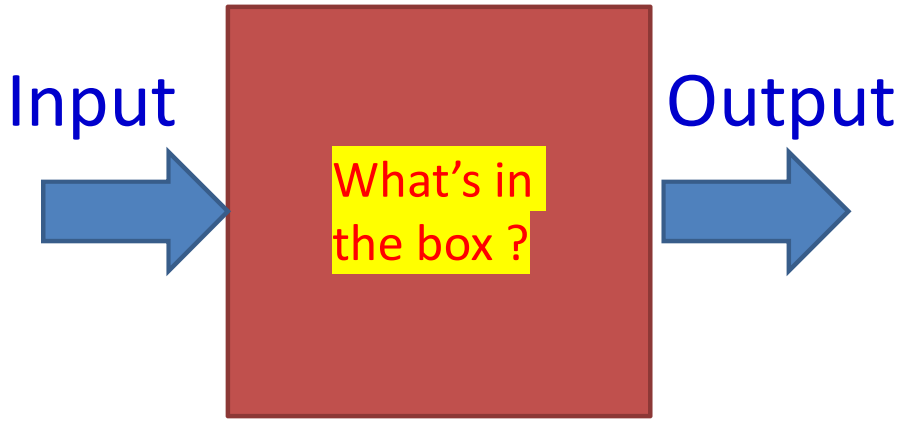
Module-1

Ramanuja Panigrahi


File-1.cpp
File-2.cpp
File-3.cpp
Source code
#include Header files
Preprocessor
Preprocessor
Preprocessor
Translates from high level to low level
Compiler
Compiler
Compiler
file1.obj
file1.obj
file1.obj
Object Code
Libraries, functions, C++ Standard Library
Linker
File.exe
Executable Program

About Subject

- Concepts of algorithm & flow charts;
- Input/output,
- constants, variables, expressions
- operators;
- Naming conventions and styles;
- Conditions and selection statements;
- Looping and control structures (while, for, do-while, break and continue);
- File I/O, Header files, String processing;
- Pre-processor directives such as #include, #define, #ifdef, #ifndef;
- Compiling and linking.



Algorithm: A step-by-step instruction for solving some problems.

Code: It is just the implementation of an algorithm in a computer.

Algorithm

- An algorithm is a procedure (a finite set of well-defined instructions) for accomplishing some task which, given an initial state, will terminate in a defined end-state.
- A sequence of instructions.
- It was created mathematician, Mohammed ibn-Musa al-Khwarizmi.
- Algorithms can be expressed in any language

algorithm

[**al**-*guh*-*rith*-*uhm*] Phonetic (Standard) IPA

noun

1 *Mathematics.* a set of rules for solving a problem in a finite number of steps, such as the Euclidean algorithm for finding the greatest common divisor.

2 *Computers.* an ordered set of instructions recursively applied to transform data input into processed data output, such as a mathematical solution, search engine result, descriptive statistics, or predictive text suggestions.

The study of algorithms is one of the foundations of computer science.

Algorithm Example

- Following a Recipe

- Recipes are a great example of an algorithm in everyday life. They illustrate a replicable set of steps to accomplish a specific goal (such as cooking Biriyani from scratch)
- Recipes are designed to create a duplicatable outcome, or to help individuals regardless of background be able to create a specific food by following a set of detailed instructions—just as an algorithm in computer science details steps to create replicable outcomes.

- 1 cup **white rice** (such as basmati or jasmine)
- 2 cups water

Instructions

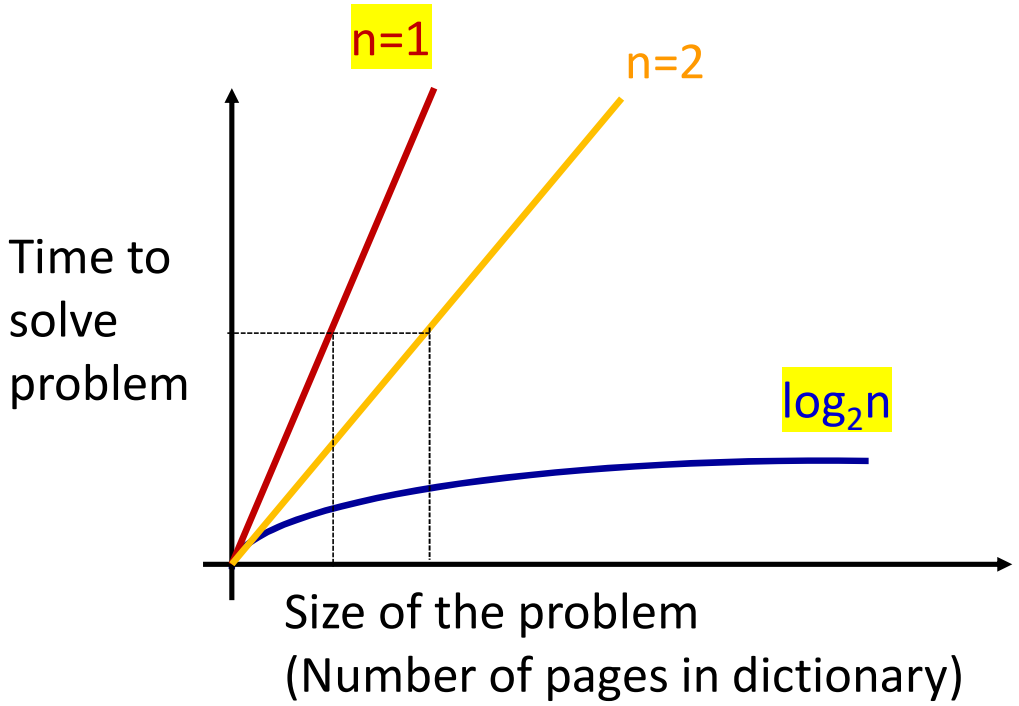
1. Add rice and water to a medium saucepan and bring to a boil over high heat. Once boiling, lower heat to a simmer and cover. Ensure it's simmering and not boiling or the rice can cook too quickly.
2. Simmer until water is completely absorbed and rice is tender – about 15-25 minutes (will depend on size and freshness of rice). Drain off any excess water if there is any (there shouldn't be).
3. I like to turn off the heat, remove the lid, fluff with a fork, put the lid back on, and let rest for 10 minutes so the moisture redistributes to prevent mushy rice. This rice is delicious in things like **stir fries**, **sushi**, with **stews**, and more!

Algorithms-correctness, efficiency

- Write an algorithm to find a word in a dictionary and return the page number.

- **Example:** Find the page on which the word "people" is. Assume the dictionary is of 1000 pages?

  - One page at a time
  - Two pages at a time
  - Divide by two.
    1. Check the mid-page
    2. IF "People" is there, give the result
    3. Else IF People is before mid-page, discard everything after mid-page
    4. Else IF People is after mid-page, discard everything before mid-page
    5. Return to step 1


n=1
n=2
Time to solve problem
$\log_2 n$
Size of the problem
(Number of pages in dictionary)

In The Art of Computer Programming (Knuth), a famous computer scientist, Donald Knuth, defines an algorithm as a set of steps, or rules, with five basic properties:

1) **Finiteness** - An algorithm must start and stop. The rules an algorithm applies must also conclude in a reasonable amount of time. What "reasonable" is depends on the nature of the algorithm, but in no case can an algorithm take an infinite amount of time to complete its task. Knuth calls this property the finiteness of an algorithm.
2) **Definiteness** - The actions that an algorithm performs cannot be open to multiple interpretations; each step must be precise and unambiguous. Knuth terms this quality definiteness. An algorithm cannot iterate a "bunch" of times. The number of times must be precisely expressed, for example, 2, 1000000, or a randomly chosen number.
3) **Inputs** - An algorithm starts its computation from an initial state. This state may be expressed as input values given to the algorithm before it starts.
4) **Outputs** - An algorithm must produce a result with a specific relation to the inputs.
5) **Effectiveness** - The steps an algorithm takes must be sufficiently simple that they could be expressed "on paper"; these steps must make sense for the quantities used. Knuth terms this property effectiveness.

- Algorithms for making things will often be divided into sections;
  - the parts/components/ingredients (inputs) required to accomplish the task
  - actions/steps/methods (processing) to produce
  - the required outcome (output).

  - Two forms of Algorithm:
    - Pseudocode
    - Flow chart

Pseudocode

- Pseudocode (which means fake code because it's not really programming code) specifies the steps required to accomplish the task.
- Pseudocode is a type of structured English that is used to specify an algorithm.
- Pseudocode cannot be compiled nor executed, and there are no real formatting or syntax rules.

Algorithm to add two numbers

Step 1: Input first number as P
Step 2: Input second number as Q
Step 3: Set Sum=P+Q
Step 4: Print Sum
Step 5: End.

Why Pseudocode?

- The programming process is a complicated one. You must first understand the program specifications.
- Then you need to organize your thoughts and create the program.
- • You must break the main tasks that must be accomplished into smaller ones in order to be able to eventually write fully developed code.
- Writing Pseudocode will save you time later during the construction & testing phase of a program's development.

How to Write Pseudocode?

- There are six basic computer operations
  1. A computer can receive information
     - Read (information from a file)
     - Get (information from the keyboard)
  2. A computer can put out information
     - Write (information to a file)
     - Display (information to the screen)
  3. A computer can perform arithmetic
     - Use actual mathematical symbols or the words for the symbols
     - Example:
       - Add number to total
       - Total = total + number

basic computer operations

- There are six basic computer operations
  4. A computer can assign a value to a piece of data
     - to give data an initial value
       - Initialize, Set
     - assign a value as a result of some processing
       - =
       - x=5+y;
     - To keep a piece of information for later use
       - Save, Store

Example

- Finding the average of any three numbers.
  - We might usually specify the procedure of solving this problem as “add the three numbers and divide by three”.
  - Here, Read (or Ask) and Write (or Say) are implied. However, in an algorithm, these steps have to be made explicit.
  - Thus a possible algorithm is

Example

- Step 1 Start
- Step 2 Read values of X, Y, Z
- Step 3 S = X + Y + Z
- Step 4 A = S / 3
- Step 5 Write value of A
- Step 6 Stop

Example

- Finding square and cube of a number.

  - Step 1 Start
  - Step 2 Read value of N
  - Step 3 S = N * N
  - Step 4 C = S * N
  - Step 5 Write values of S, C
  - Step 6 Stop

Basic computer operations (continued…)

5. A computer can compare two pieces of information and select one of two alternative

Flow Control by Choice

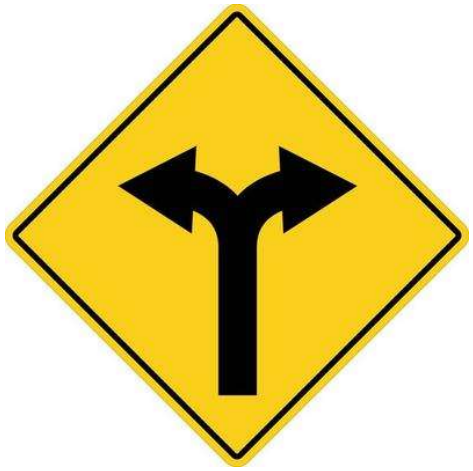
- IF condition THEN

  some action

ELSE

  alternative action

ENDIF



Example

- Finding the bigger of two numbers.

A = 5
B = 10

A = 100
B = 10

- Step 1 Start
- Step 2 Read A, B
- Step 3 If A > B,
  then BIG = A,
  otherwise
  BIG = B
- Step 4 Write BIG
- Step 5 Stop

Flow Control by Choice

Example

- Finding the smallest of three numbers.

A = 1 C = 3
B = 5

- Step 1 Start
- Step 2 Read A, B, C
- Step 3 If A < B,
  then If A< C
  Small=A
  otherwise
  Small=C
  otherwise
  then If B< C
  Small=B
  otherwise
  Small=C
- Step 4 Write Small
- Step 5 Stop

Basic computer operations (continued…)

6. A computer can repeat a group of actions

- Example:

WHILE condition (is true)

    some action

ENDWHILE

FOR a number of times

    some action

ENDFOR

Example

- Print “Hello” 5 times

  - Step 1 Start
  - Step 2 Write “Hello”
  - Step 3 Write “Hello”
  - Step 4 Write “Hello”
  - Step 5 Write “Hello”
  - Step 6 Write “Hello”
  - Step 7 Stop

```
Step 1      Start
Step 2      For 5 times
                Write “Hello”
            End for
Step 3      Stop
```

- Step 1 Start
- Step 2 Read n
- Step-3 For n times

  Write “Enter your name”

  Read Name

  Write “ Hello” Name

  End for
- Step 4 Stop

- Step 1 Start
- Step 2 Read n
- Step-3 For n times

  Call Function Intro

  End for
- Step 4 Stop

```
Function Intro
{
    Write “Enter your name”
    Read Name
    Write “ Hello” Name
}
```

- The pseudo code by itself doesn't provide enough information to be able to write program code.

- Data Dictionaries are used to describe the data used in the Pseudo Code.

- The standard data types used in Pseudo Code are Integer, Double, String, Char, and Boolean.

- The flowchart is a diagram that visually presents the flow of data through processing systems.

- This means by seeing a flow chart one can know the operations performed and the sequence of these operations in a system.

- A flowchart is a graphical representation of an algorithm.

  "A picture is worth a thousand words“

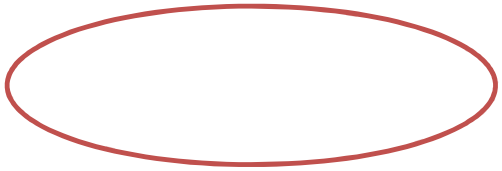
Flow Chart

- These flowcharts play a vital role in the programming of a problem and are quite helpful in understanding the logic of complicated and lengthy problems.
- Once the flowchart is drawn, it becomes easy to write the program in any high-level language

- A flowchart can therefore be used to:
  - Define and analyze processes
  - Build a step-by-step picture of the process for analysis, discussion, or communication
  - –Define, standardize or find areas for improvement in a process

Flowchart Symbols

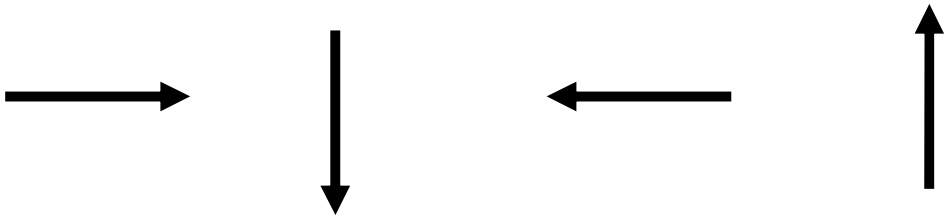
- Start and end symbols



- Represented as lozenges, ovals, or rounded Rectangles
- Usually containing the word "Start" or "End" or another phrase signaling the start or end of a process, such as "submit inquiry" or "receive product."

Flowchart Symbols

- Arrows



- Showing what's called "flow of control" in computer science.
- An arrow coming from one symbol and ending at another symbol.
- Represents that control passes to the symbol the arrow points to.

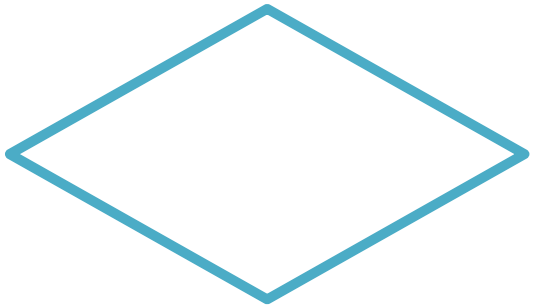
- Input/ Output

- Represented as a parallelogram.
- Examples: Get X from the user; display X

- Processing steps

- Represented as rectangles.
- Examples: “Add 1 to X"; "replace identified part"; "save changes" or similar.

- Conditional or decision

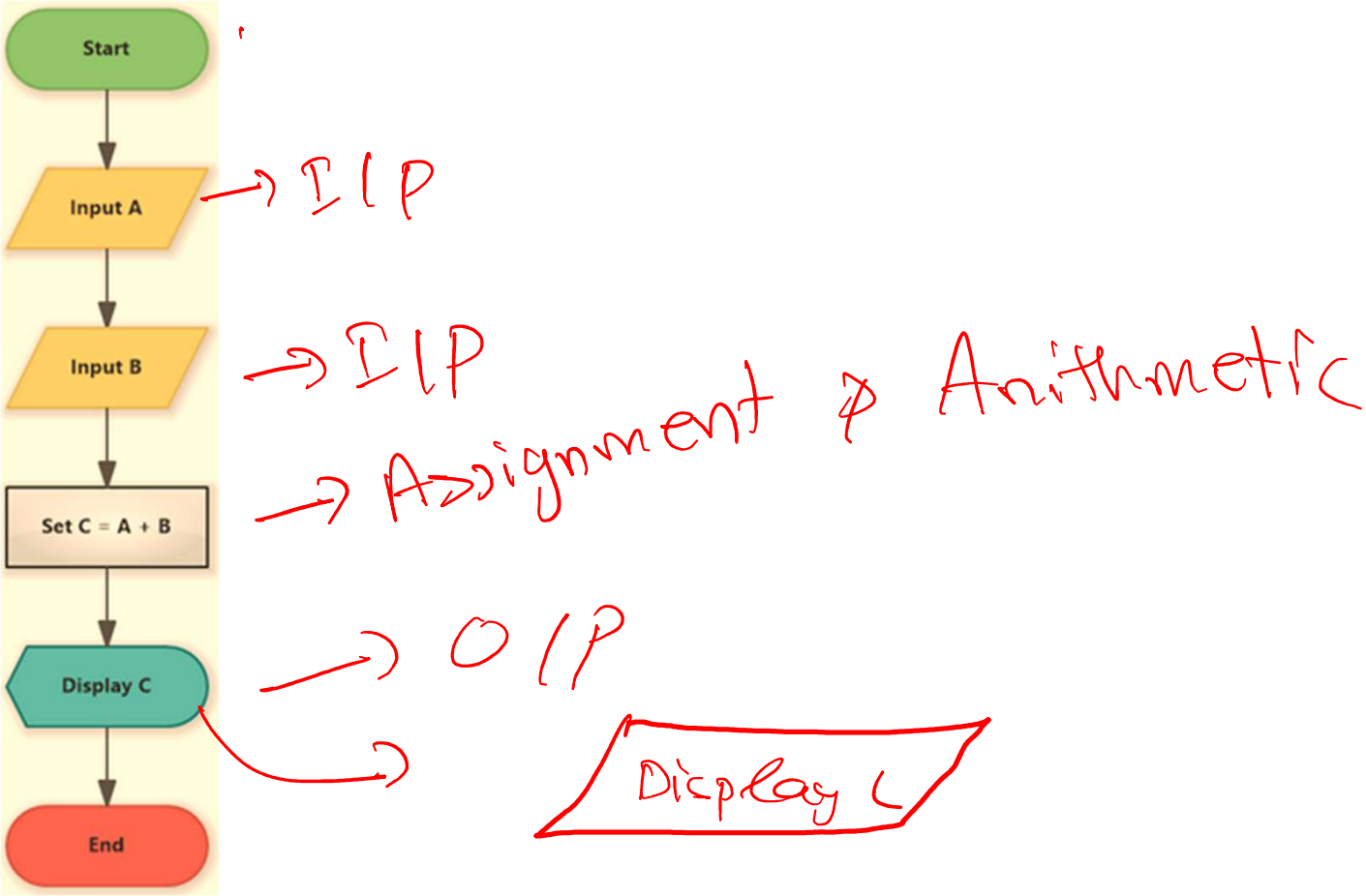


- Represented as a diamond (rhombus).
- These typically contain a Yes/No question or True/False test.

- Display

- Indicates a process flow step where information is displayed to a person (e.g., PC user, machine operator).

Flowchart to add two numbers



Flowchart Guidelines

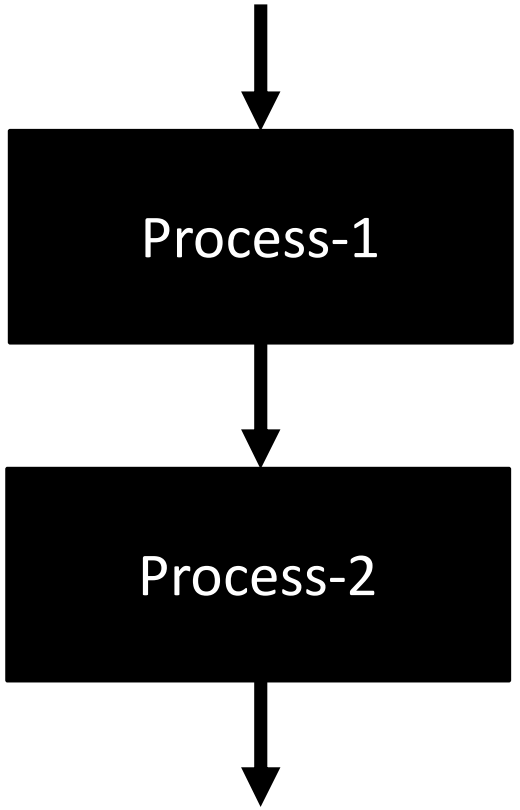
- Every flow chart has a START symbol and a STOP symbol.
- The flow of sequence is generally from the top of the page to the bottom of the page. This can vary with loops that need to flow back to an entry point.
- Use arrow-heads on connectors where flow direction may not be obvious.
- A flow chart should have no more than around 15 symbols (not including START and STOP).

Advantages of Using Flowcharts

1. Communication: Flowcharts are a better way of communicating the logic of a system to all concerned.
2. Effective analysis: With the help of a flowchart, the problem can be analyzed in a more effective way.
3. Proper documentation: Program flowcharts serve as good program documentation, which is needed for various purposes.
4. Efficient Coding: The flowcharts act as a guide or blueprint during the systems analysis and program development phase.
5. Proper Debugging: The flowchart helps in the debugging process.
6. Efficient Program Maintenance: The maintenance of the operating program becomes easy with the help of a flowchart. It helps the programmer to put efforts more efficiently on that part.

Basic Flow Control Structure

1. Sequence
   - Steps that execute in sequence are represented by symbols that follow each other top to bottom or left to right.
   - Top to bottom is the standard.

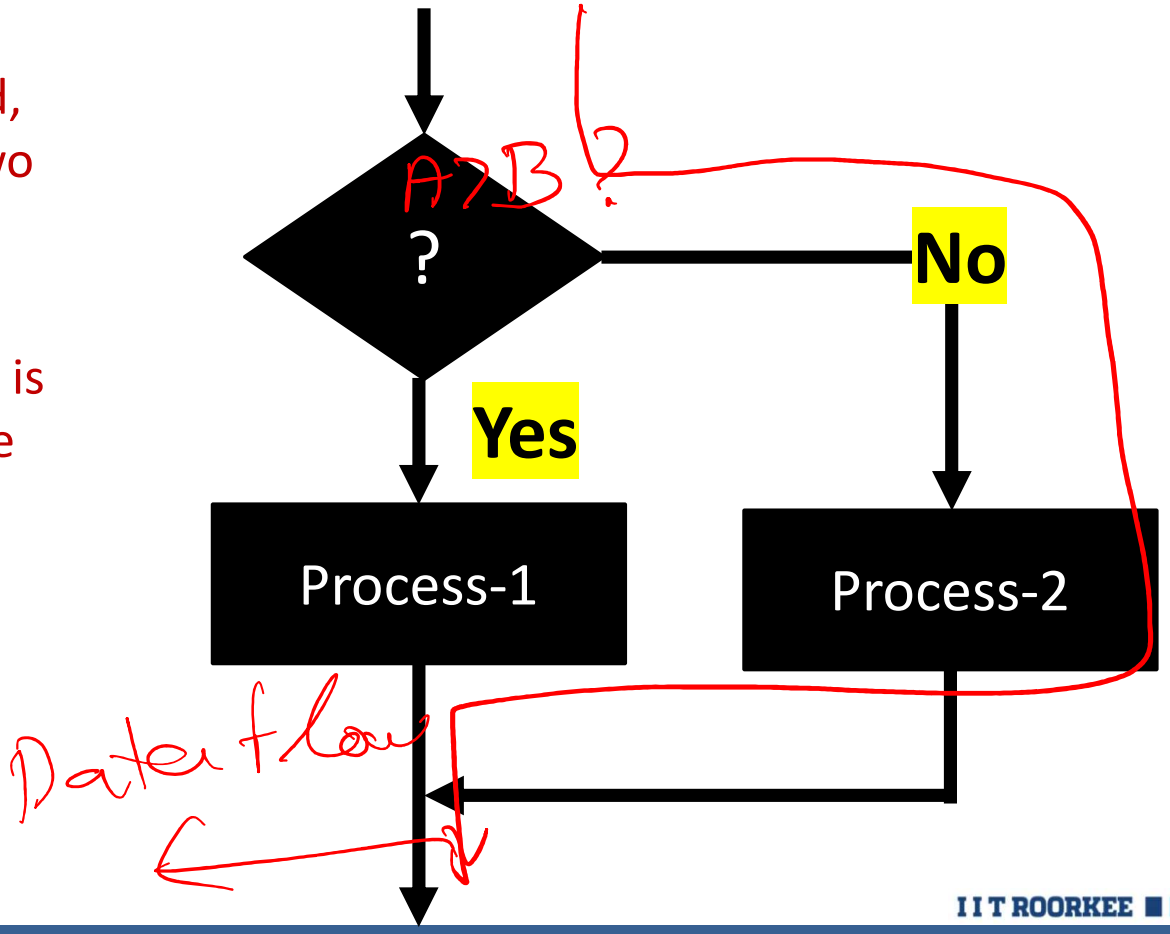


Basic Flow Control Structure

2. Selection
   - Once the condition is evaluated, the control flows into one of two paths.
   - Once the conditional execution is finished, the flows rejoin before leaving the structure.

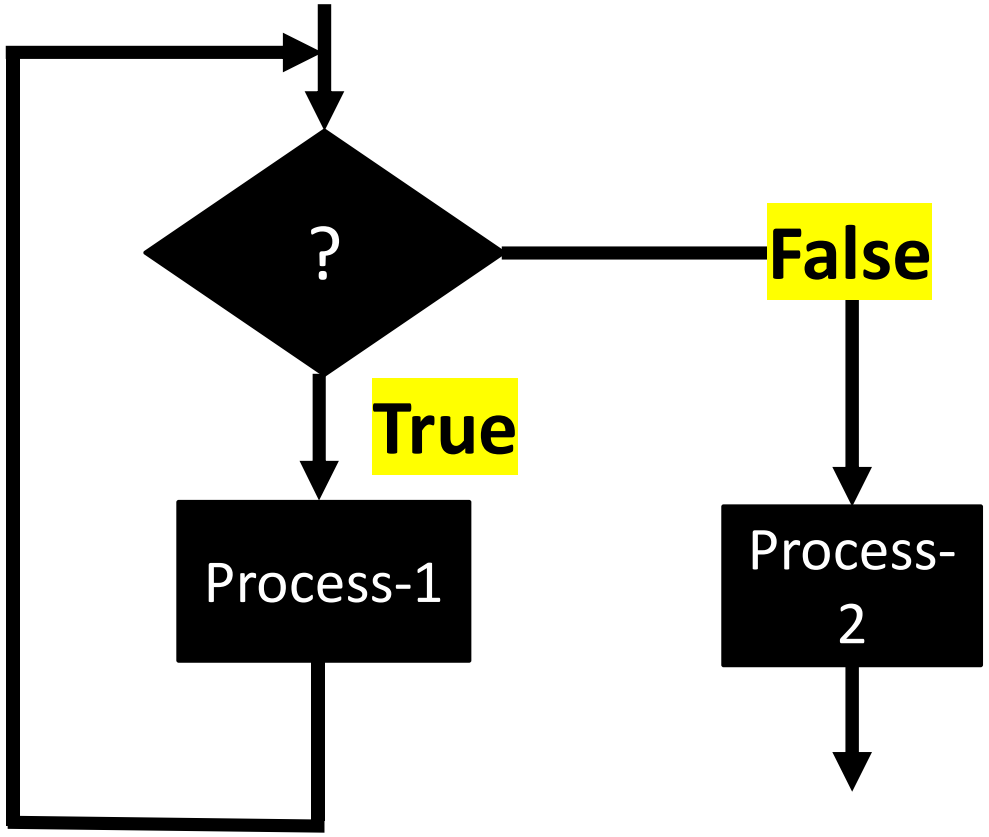
A = 3

B = 5



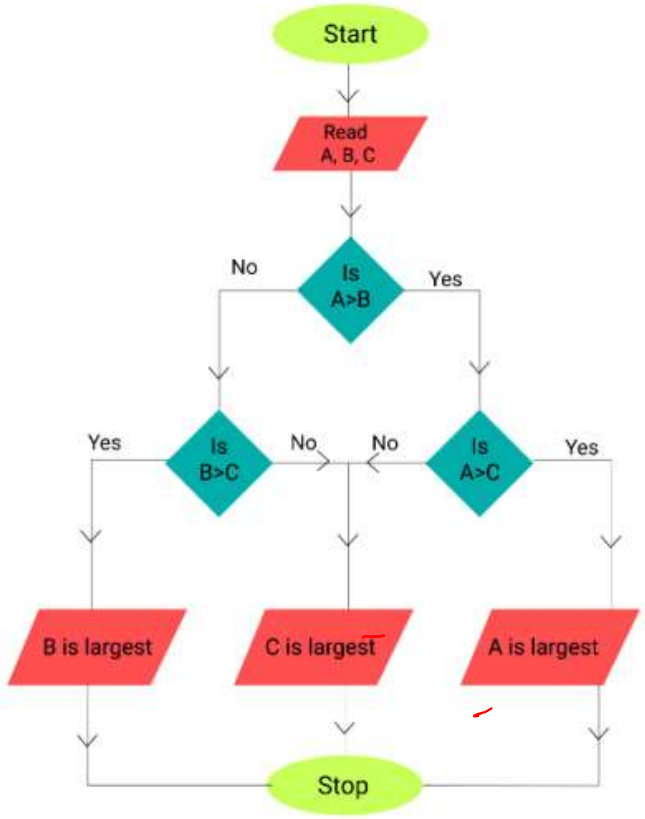
Basic Flow Control Structure

2. Loop
   - Either the processing repeats or the control leaves the structure.
   - Notice that the return line joins the entry line before the question

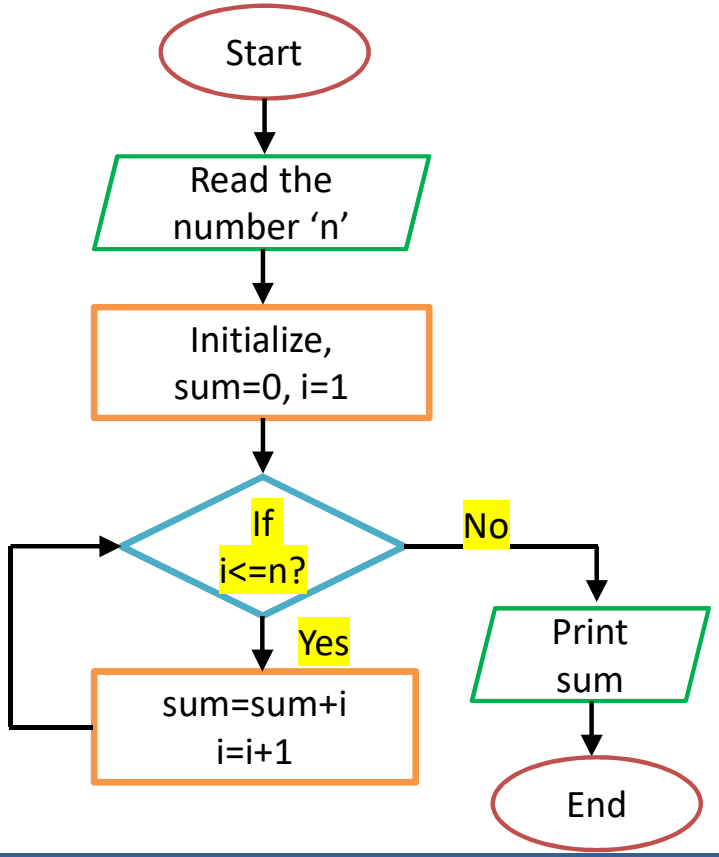


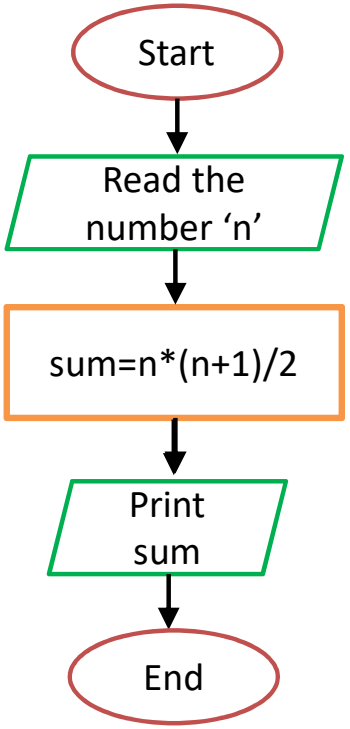
Example

Draw a flow chart to identify the largest of three numbers



- Draw a flowchart to print the sum of first ‘n’ natural





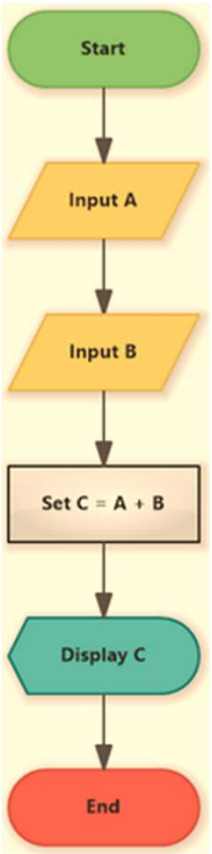
What is a program?

- A computer program is a sequence of instructions (written in a programming language) that, when executed on a computer, implements the algorithm.

From BlackBox to Program in C++

Algorithm to add two numbers

Step 1: Input first number as P
Step 2: Input second number as Q
Step 3: Set Sum=P+Q
Step 4: Print Sum
Step 5: End.



Flowchart to add two numbers

```
int main() {

  int first_number, second_number, sum;

  cout << "Enter first integers: ";
  cin >> first_number;
  cout << "Enter second integers: ";
  cin >>  second_number;
  // sum of two numbers in stored in variable sumOfTwoNumbers
  sum = first_number + second_number;

  // prints sum
  cout << first_number << " + " <<  second_number << " = " << sum;

  return 0;
}
```

A C++ program to add two numbers

Homework

- Algorithm & Flowchart to find sum of series $1 - X + X^2 - X^3 + \cdots + X^N$. Where X and N are user inputs.