

**GOVERNMENT POLYTECHNIC FOR GIRLS  
AHMEDABAD**

**Computer Department**

# COMPUTER PROGRAMMING (CODE: 3310701)

- This Course intends to develop programming skills in the students, using a popular structured programming language 'C'. The students will learn step by step procedure (i.e. flowcharting & Algorithm) of any program development process.
- The programming skills thus acquired using 'C' language can be used for acquiring necessary programming skill to work with advance level programming languages which in turn will be helping in developing programs for the scientific, research and business purposes.

# COMPETENCIES

- The course content should be taught and implemented with the aim to develop the following competencies.

**Develop Simple Programs using ‘C’ Language**

# TEACHING AND EXAMINATION SCHEME

Teaching Scheme (In Hours)			Total Credits (L+T+P)	Examination Scheme				
L	T	P		Theory Marks		Practical Marks		Total Marks
			C	ESE	PA	ESE	PA	200
3	0	4	7	70	30	40	60	

**Legends:** L-Lecture; T – Tutorial/Teacher Guided Theory Practice; P - Practical; C – Credit;  
ESE - End Semester Examination; PA - Progressive Assessment.

# UNIT 1

## FLOWCHART & ALGORITHM

- **Major Learning Outcomes**

- 1.1. Draw flow chart to solve given problem logically.
- 1.2. Develop Algorithm to solve given program.

# TOPICS AND SUB-TOPICS

## Flowchart

- Definition and Importance of flowchart.
- Symbols of Flowchart.
- Flow lines, Terminals, Input/output, Processing, Decision, Connection off-page connectors
- Guidelines for preparing Flowchart.
- Flowchart structure  
Sequence, selection, repetition.
- Limitation of flowchart

## Algorithm

- Developing and writing algorithm using pseudo codes

# FLOWCHART

## Definition

A pictorial or graphical representation of the sequence of any problem or process to solve by programming language.

### A Flowchart

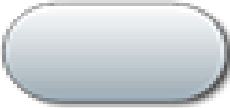

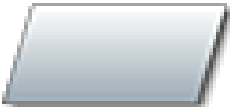


- shows logic of an algorithm
- emphasizes individual steps and their interconnections
- e.g. control flow from one action to the next

# IMPORTANCE OF FLOWCHART

- **Better Communication** : A flowchart is a pictorial representation of a program. Hence, it is easier for a programmer to explain the logic of a program to some others.
- **Proper program documentation** : Program documentation involves collecting, organizing, storing and maintaining a complete historical record of programs, and the other documents associated with a system.
- **Efficient coding** : Once a flowchart is ready, programmers find it very easy to write the corresponding program, because the flowchart acts as a road map for them.
- **Systematic debugging** : A flowchart is very helpful in detecting, locating and removing mistakes ( bugs) in a program in a systematic manner, because programmers find it easier to follow the logic of the program in flowchart form.
- **Systematic testing** : Testing is the process of confirming whether a program will successfully do all the jobs for which it has been designed under the specified constraints.



# SYMBOLS OF FLOWCHART

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 rectangle represents a process.
	Decision	A diamond indicates a decision.

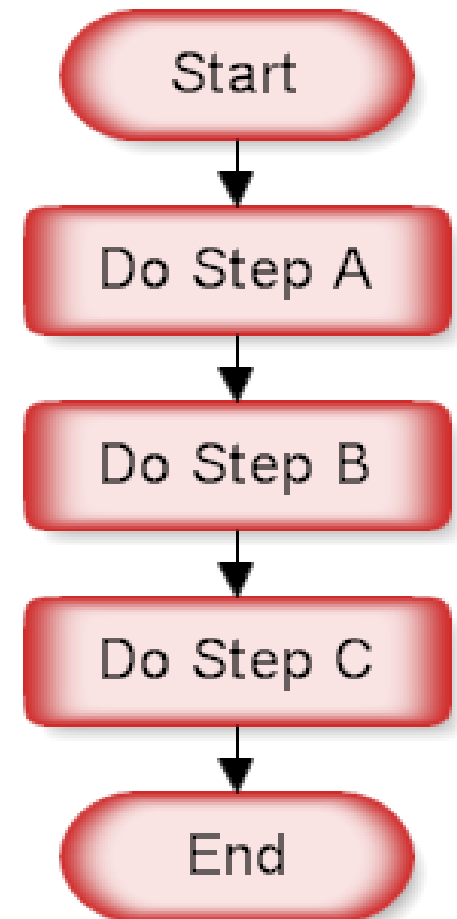
# GUIDELINES FOR PREPARING FLOWCHART

- In drawing a proper flowchart, all necessary requirements should be listed out in logical order.
- The flowchart should be neat, clear and easy to understand.
- The flowchart is to be read left to right or top to bottom.
- A process symbol can have only one flow line coming out of it.
- For a decision symbol, only one flow line can enter it, but multiple lines can leave it to denote possible answers.
- The terminal symbols can only have one flow line in conjunction with them.
- All boxes of the flowchart are connected with arrows.
- Draw a rough sketch of the system.

# FLOWCHART STRUCTURE

## 1. Sequence

The steps or process that are executed in sequence are represented by process symbols one after another either top to bottom or left to right in the flowchart.

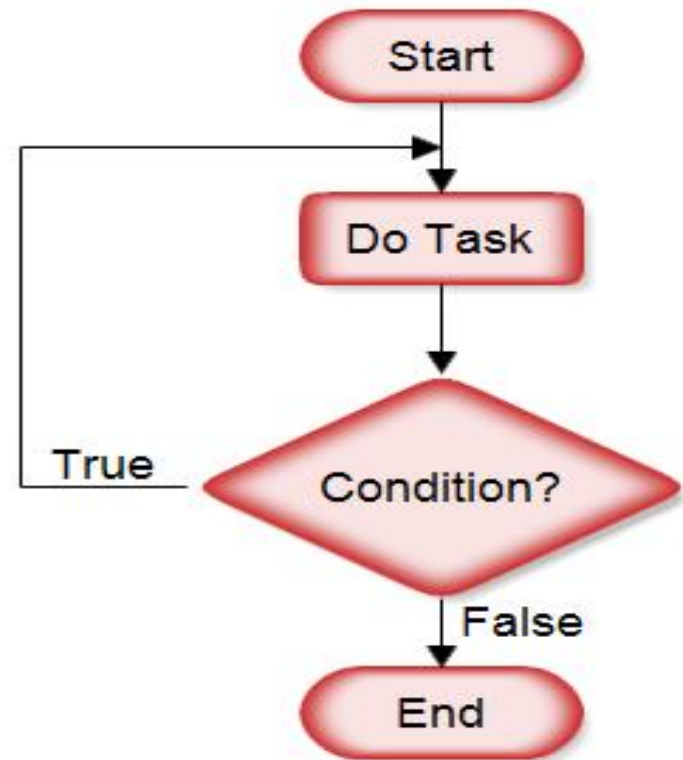


# FLOWCHART STRUCTURE

## 2. Repetition

It is used when a program needs to repeatedly process one or more instructions until some condition is met, at which time the loop ends. Many programming tasks are repetitive, having little variation from one item to the next.

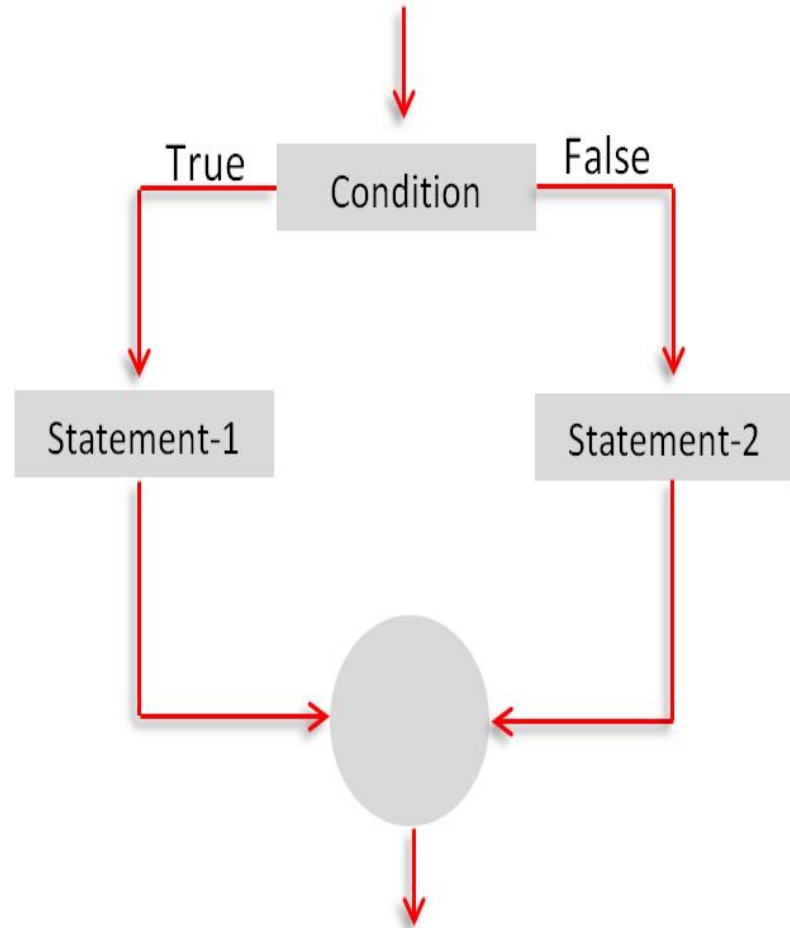
## Do While Loop



# FLOWCHART STRUCTURE

## 3. Selection

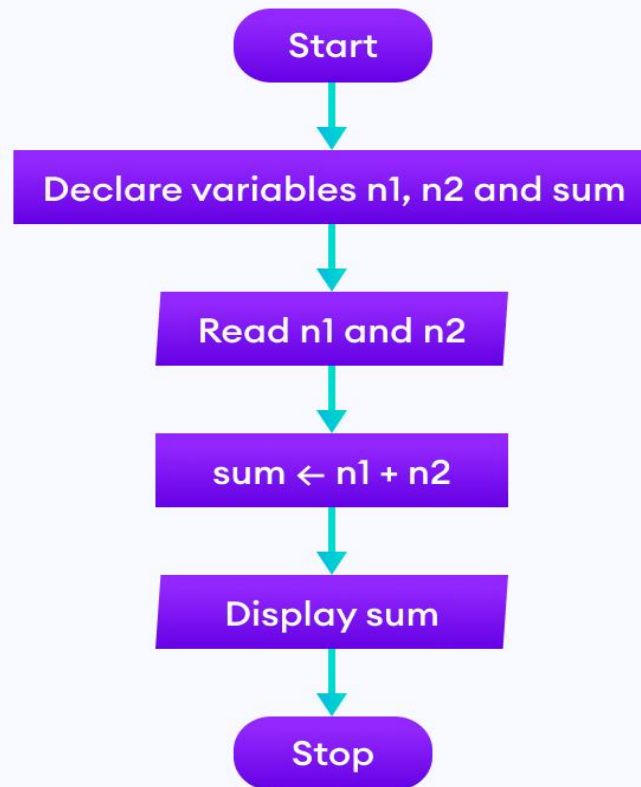
It is also known as a conditional structure, a selection structure is a programming feature that performs different processes based on whether a Boolean condition is true or false. Selection structures use relational operators to test conditions.



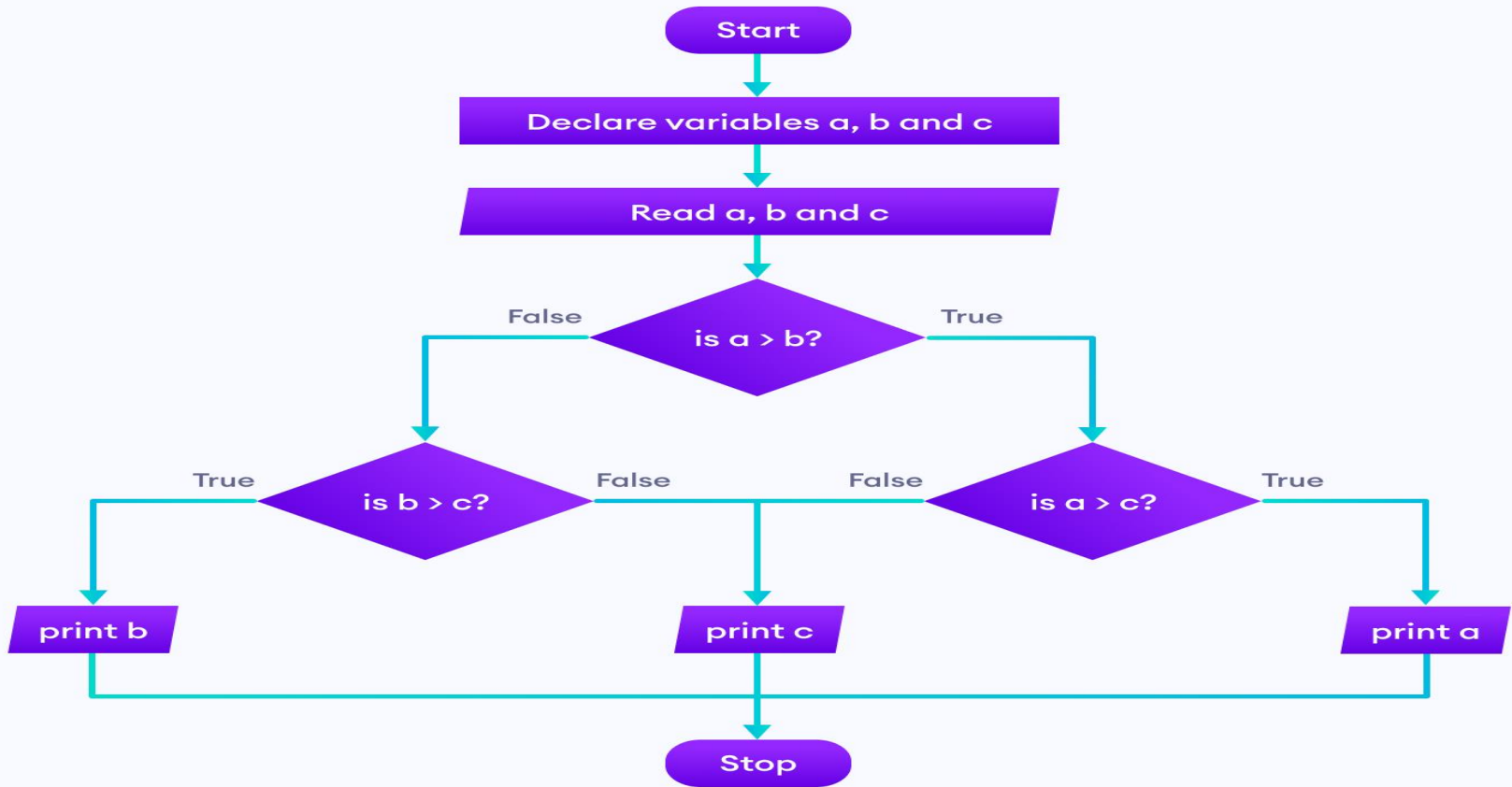
# LIMITATION OF FLOWCHART

- Flowcharts are very time consuming and laborious to draw with proper symbols and spacing, especially for large complex programs.
- Owing to the symbol –string nature of flowcharting any changes or modifications in the program logic will usually require a completely new flowchart.
- There are no standards determining the amount of detail that should be included in the flowchart.

# DRAW FLOWCHART FOR ADDITION OF TWO NUMBERS



**DRAW THE FLOWCHART TO FIND THE LARGEST AMONG THREE DIFFERENT NUMBERS ENTERED BY THE USER.**





## EXAMPLES

1. Draw the flowchart to find out number is even or odd.
2. Draw the flowchart to calculate area of circle.
3. Draw the flowchart to calculate area of rectangle.
4. Draw the flowchart to find average of two numbers.
5. Draw the flowchart to find square of a number.

# ALGORITHM

- A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.
- A typical programming task can be divided into two phases:
- Problem solving phase
  - produce an ordered sequence of steps that describe solution of problem
  - this sequence of steps is called an algorithm
- Implementation phase
  - implement the program in some programming language

# EXAMPLE 1: WRITE AN ALGORITHM TO PRINT “HELLO”

**Step 1:** start

**Step 2:** Print “HELLO”

**Step 3:** stop

**EXAMPLE 2: WRITE AN ALGORITHM TO CALCULATE AREA OF CIRCLE.**

**Step 1:** start

**Step 2:**Input r

**Step 3:**area  $\leftarrow (3.14*r*r)$

**Step 4:**Print area

**Step 5:** stop

# EXAMPLE 3: WRITE AN ALGORITHM TO FIND OUT NUMBER IS EVEN OR ODD.

**Step 1:** Start

**Step 2:** Read Number

**Step 3:** Check If  $\text{Number} \% 2 == 0$  Then

Print N is an Even Number.

Else

Print N is an Odd Number.

**Step 4:** stop

**EXAMPLE 4: WRITE AN ALGORITHM TO DETERMINE A STUDENT'S FINAL GRADE AND INDICATE WHETHER IT IS PASSING OR FAILING. THE FINAL GRADE IS CALCULATED AS THE AVERAGE OF FOUR MARKS.**

**Step 1:** start

**Step 2:** Input M1,M2,M3,M4

**Step 3:**  $\text{GRADE} \leftarrow (M1+M2+M3+M4)/4$

**Step 4:** if (GRADE <50 then)

    Print "fail"

    else

    Print "pass"

**Step 5:** stop

**EXAMPLE 5: WRITE AN ALGORITHM TO FIND OUT FACTORIAL OF GIVEN NUMBER.**

**Step 1:**Start

**Step 2:**Read the number.

**Step 3:**Intialization

$i=1, \text{Fact}=1$

**Step 4:**Repeat step 4 to 6 until  $i \leq \text{number}$

**Step 5:** $\text{Fact}=\text{Fact} * i$

**Step 6:** $i=i+1$

**Step 7:**print Fact

**Step 8:**Stop

**EXAMPLE 6: WRITE AN ALGORITHM TO GENERATE FIBONACCI SERIES.**

**Step 1:**Start

**Step 2:** Declare variable x,y,no,z

**Step 3:** Initialize variable x=0, y=1

**Step 4:** Read no from user

**Step 5:** Print x and by

**Step 6:** Repeat until no $\geq$ x

z=x+y

print z

x=y, y=z

**Step 7:**Stop



**EXAMPLE 7: WRITE AN ALGORITHM TO CHECK WHETHER ENTERED NUMBER IS PRIME OR NOT.**

**Step 1:**Start

**Step 2:** Read number n

**Step 3:** Set f=0

**Step 4:** For i=2 to n-1

**Step 5:** If n mod i=0 then

**Step 6:** Set f=1 and break

**Step 7:** If f=0 then

    print 'The given number is prime'

    else

    print 'The given number is not prime'

**Step 8:**Stop

## EXAMPLES

1. Write an algorithm to find average of two numbers.
2. Write an algorithm to calculate area of rectangle.
3. Write an algorithm to find square of a number.
4. Write an algorithm to calculate simple interest.
5. Write an algorithm to check entered number is Positive or negative.

Thank You

Thank You