SNS COLLEGE OF TECHNOLOGY

Coimbatore-36.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE NAME : 19CST101 – PROGRAMMING FOR PROBLEM SOLVING

I YEAR/ I SEMESTER

UNIT – I INTRODUCTION TO PROBLEM SOLVING TECHNIQUES

Topic: Algorithms & Building Blocks Of Algorithm

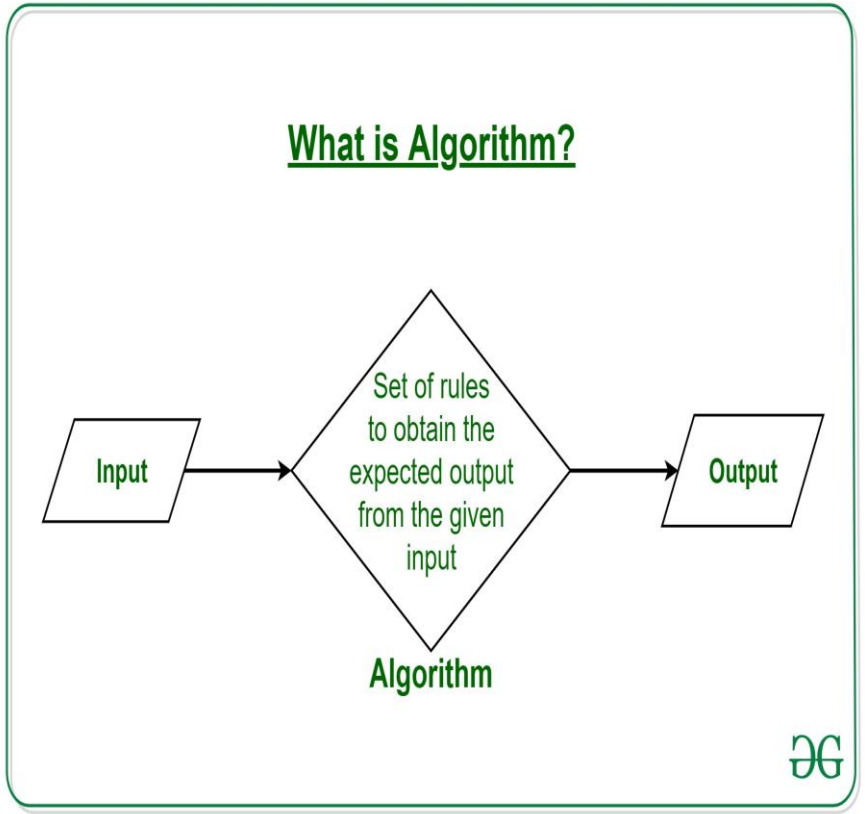
Mr. N. Selvakumar
Assistant Professor
Department of Computer Science and Engineering

Algorithms

Algorithm is a sequence of instructions that describe a method for solving a problem. It is a step by step procedure for solving a problem

Properties of Algorithms

- Should be written in simple English
- ach and every instruction should be precise and unambiguous
- Instructions in an algorithm should not be repeated infinitely
- Algorithm should conclude after a finite number of steps
- Should have an end point
- Derived results should be obtained only after the algorithm terminates



Qualities of a good algorithm

1. Time
2. Memory
3. Accuracy

Algorithms

Example: C program for Print the **"WELCOME TO SNSCT"**

Program:

```
1. #include <stdio.h>
//where the execution of program begins
1. Int main()
2. {
3. Printf(" WELCOME TO SNSCT");
4. Return 0;
5. }
```

Output:

WELCOME TO SNSCT

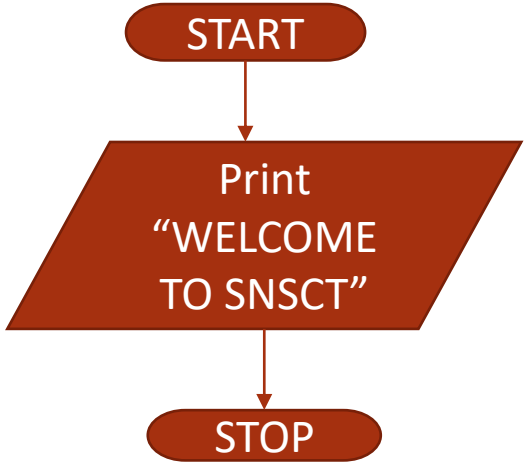
Algorithm:

Step 1: Start

Step 2: Print "WELCOME TO SNSCT"

Step 3: Stop

Flowchart:



Building Blocks of Algorithms

Algorithms can be constructed from basic building blocks namely,

1. Statements:
2. State
3. Control Flow
4. Functions

Building Blocks of Algorithms

1.Statements:

Statement is a single action in a computer.

1. Input Data
2. Process Data
3. Output Data

2.State:

Transition from one process to another process under specified condition with in a time is called state

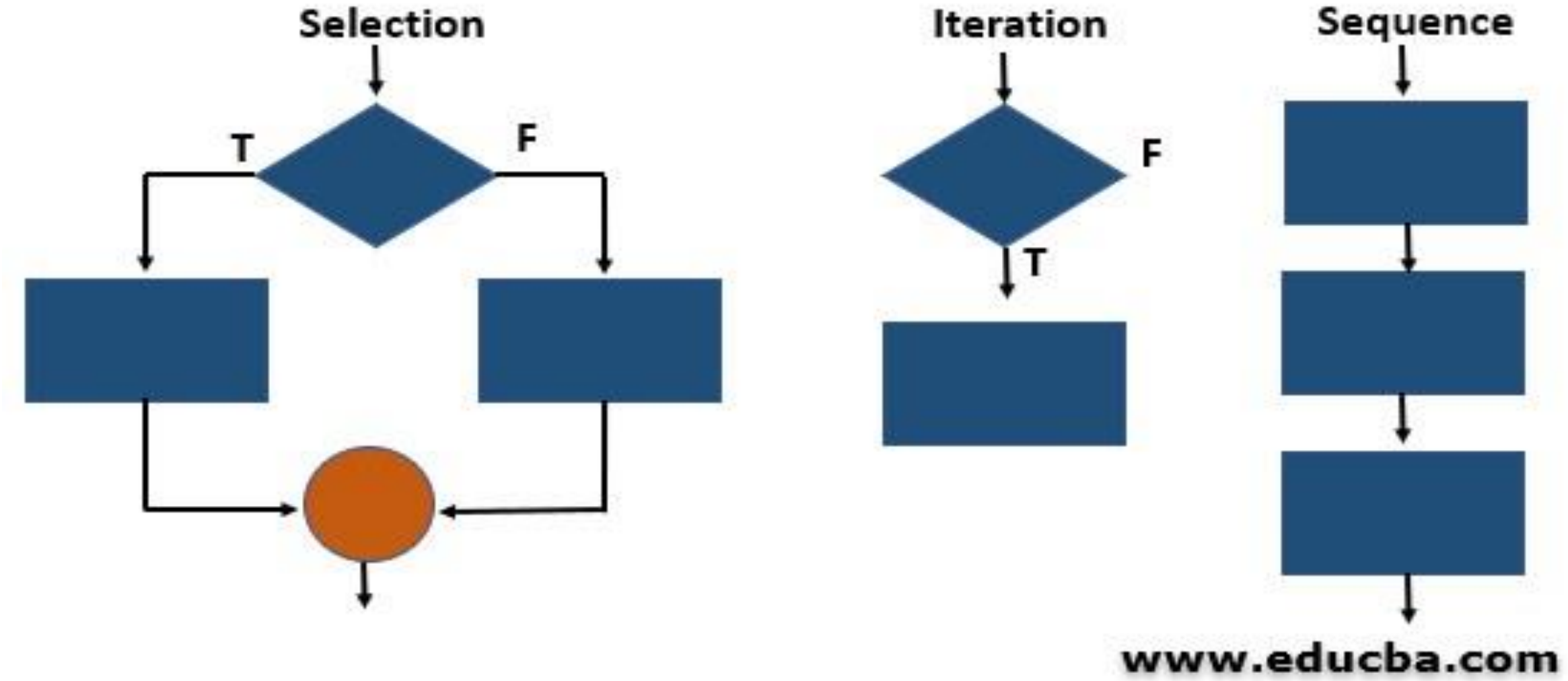
3.Control flow:

The process of executing the individual statements in a given order is called control flow

The control can be executed in three ways

1. Sequence
2. Selection
3. Iteration

Building Blocks of Algorithms



Building Blocks of Algorithms

1.Sequence: All the instructions are executed one after another is called sequence execution

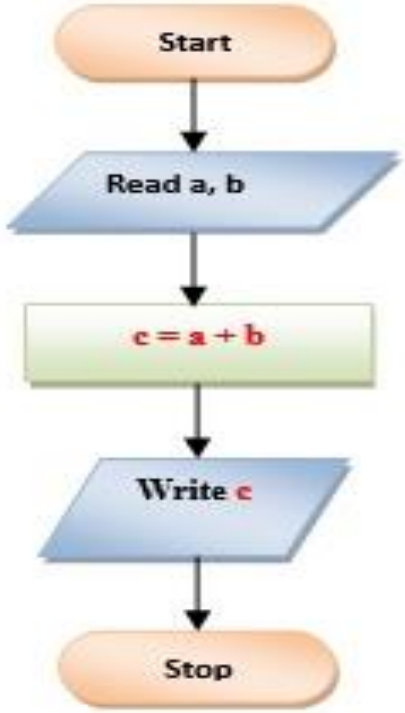
Example: Algorithm for Addition of TWO NUMBERS

To find sum of two numbers

Algorithm

1. Start
2. Read a, b
3. c = a + b
4. Print or display c
5. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{
    int a, b, c;

    printf("Enter value of a: ");
    scanf("%d", &a);

    printf("Enter value of b: ");
    scanf("%d", &b);
    c = a+b;

    printf("Sum of given two numbers is: %d", c);

return 0;
}
```

Building Blocks of Algorithms

2.Selection: A selection statement causes the program control to be transferred to a specific part of the program based upon the condition. If the conditional test is true, one part of the program will be executed, otherwise it will execute the other part of the program.

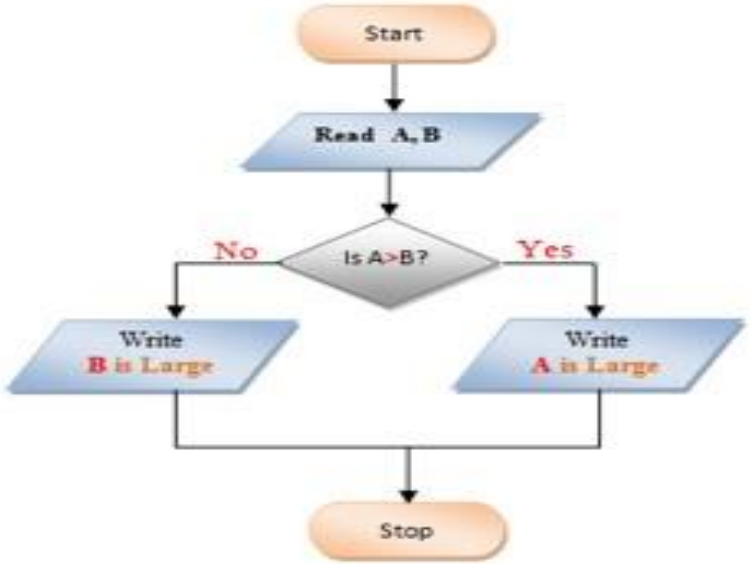
Example: Algorithm for Greatest of TWO NUMBERS

Greatest of two numbers

Algorithm

1. Start
2. Read A,B
3. If A > B then
   Print A is large
   else
   Print B is large
4. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{

    int A, B;

    printf("Enter values of A, B: ");
    scanf("%d %d", &A, &B);

    if (A>B)
    printf("A is Larger");
    else
    printf("B is Larger");

    return 0;
}
```

Building Blocks of Algorithms

3.Iteration: In programs, certain set of statements are executed again and again based upon conditional test. It executed more than one time. This type of execution is called looping or iteration.

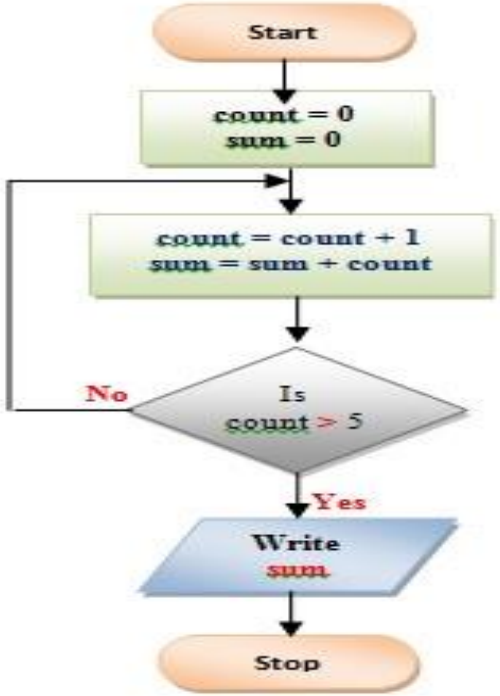
Example: Algorithm for sum of FIRST FIVE NATURAL NUMBERS

Find the Sum of First Five Natural Numbers

Algorithm

1. Start
2. Initialize count = 0, sum = 0
3. count = count + 1
4. sum = sum + count
5. Repeat steps 3,4 until count > 5
6. Print sum
7. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{

    int count, sum;
    sum = 0;

    for (count = 1; count<=5; count++)
    {
        sum = sum +count;
    }

    printf("Sum of 1st 5 numbers is: %d", sum);
    return 0;
}
```

Building Blocks of Algorithms

4.Functions: Function is a sub program which consists of block of code(set of instructions) that performs a particular task.

For complex problems, the problem is been divided into smaller and simpler tasks during algorithm design.

Benefits of Using Functions:

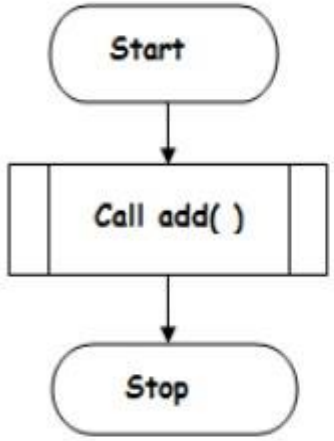
- Code reuse
- Reduction in line of code
- Easy to debug and test
- Easy to debug and test

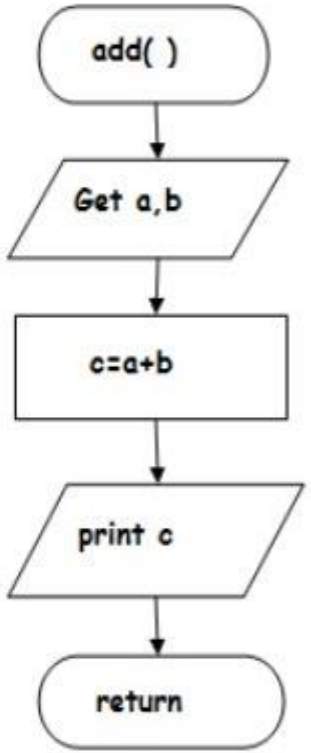
Main function()

Step 1: Start

Step 2: Call the function add()

Step 3: Stop





sub function add()

Step 1: Function start

Step 2: Get a, b Values

Step 3: add c=a+b

Step 4: Print c

Step 5: Return


Thank
you