

CSE101-Lec 2

Program structure of C program.

Various programming tools like flow chart and algorithms.

Algorithm

- Algorithm is defined as “ the finite set of steps, which provide a chain of action for solving a problem”
- It is step by step solution to given problem.
- Well organized, pre-arranged and defined textual computational module

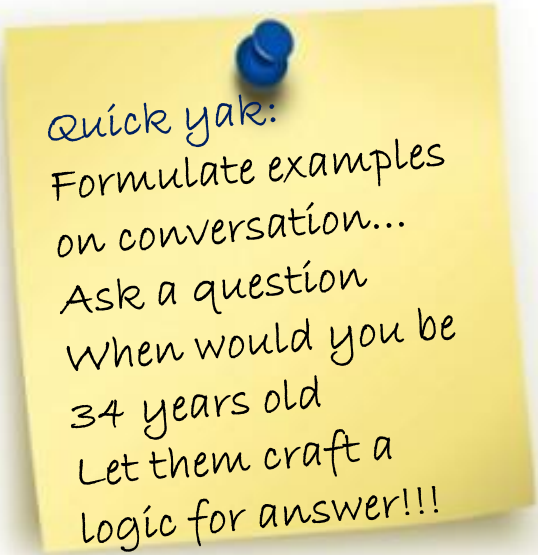
Characteristics of good Algorithm

1. **Correctness** - terminates on ALL inputs (even invalid inputs!) and outputs the correct answer.
2. **Simplicity** - each step of the algorithm performs one logical step in solving the problem.
3. **Precision** - each step of the algorithm is unambiguous in meaning.
4. **Comprehensibility** - the algorithm is easy to read and understand.
5. **Abstraction** - presents the solution steps precisely and concisely without referring to low-level (program code) details.
6. **Efficient** - Gives results rapidly based on the problem size; does not waste any space or time.
7. **Easy to Implement** - relatively easy to translate into a programming language.

Steps to create an Algorithm

1. Identify the Inputs

- What data do I need?
- How will I get the data?
- In what format will the

A yellow sticky note is pinned to the right side of the slide with a blue pushpin. It contains handwritten text in black ink.

quick yak:
Formulate examples
on conversation...
Ask a question
When would you be
34 years old
Let them craft a
logic for answer!!!

2. Identify the Outputs

- What outputs do I need to return to the user?
- What format should the outputs take?

Steps to create an Algorithm

3. Identify the Processes

- How can I manipulate data to produce meaningful results?
- Data vs. Information

4. Break the Solution to steps

By breaking the solution to the steps we can easily understand the logic of program

Example of Algorithm

To establish a telephone communication

- Step 1: Dial a phone number
- Step 2: Phone rings at the called party
- Step 3: Caller waits for the response
- Step 4: Called party picks up the phone
- Step 5: Conversation begins between them
- Step 6: After the conversation, both disconnect the call



Algorithm: Add 2 Numbers

Problem: To add two numbers

- Step1. Start.
- Step2. Take the two numbers.
- Step3. Add them.
- Step4. Print the result.
- Step5. Stop.

- Problem :To print the greatest number among two numbers.
- Step1:START
- Step2:Enter the two numbers num1 and num2;
- Step3:compare two numbers

If(num1>num2)

Print num1 is greatest

Else


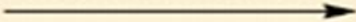


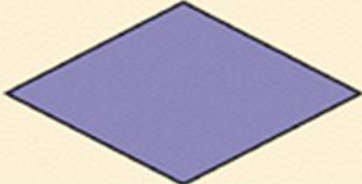
PrintNum2 is the greatest

Step4:STOP

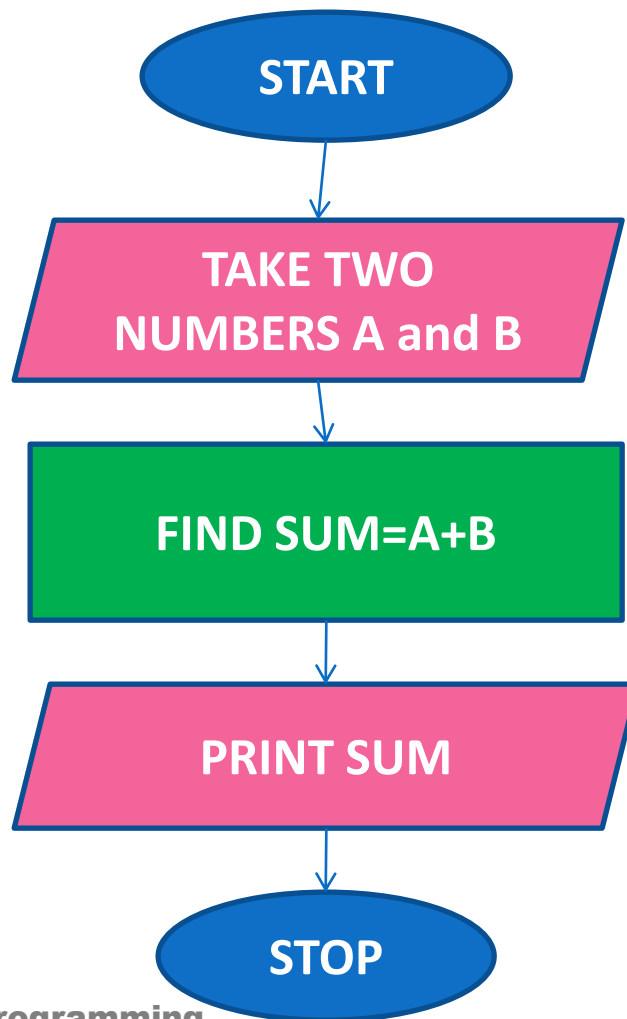
Flow Chart

- Flow Chart is pictorial representation of an algorithm.
- Whatever we have done in algorithm we can represent it in picture.
- It is easy to understand.
- Shows the flow of the instruction

Flow Chart Symbols

Name	Symbol	Use in flowchart
Oval		Denotes the beginning or end of a program.
Flow line		Denotes the direction of logic flow in a program.
Parallelogram		Denotes either an input operation (e.g., INPUT) or an output operation (e.g., PRINT).
Rectangle		Denotes a process to be carried out (e.g., an addition).
Diamond		Denotes a decision (or branch) to be made. The program should continue along one of two routes (e.g., IF/THEN/ELSE).

Flow Chart: Add 2 Numbers



quick yak:
Ask students to
draw a flow chart
for going to a
movie...



Program in C: Add 2 Numbers

```
#include<stdio.h>
int main()
{
    int a=4;
    int b= 2;
    int sum;
    sum = a+b;
    printf("Sum is: %d", sum);
}
```

Sum is: 6

Program in C

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int pr_per_kg = 45;
    float no_of_kg = 7.5;
    float t_pr;
    t_pr = pr_per_kg * no_of_kg;
    printf("%f", t_pr);
    getch();
}
```

337.5



Structure of C program

≡ File Edit Search Run Compile Debug Project Options Window Help

```
[■] P_1_1.C 1=[↑]
//Problem 1.1: If the price of one kg mango is 45 Rs. then find the price
//of 7.5 kg mangoes.

#include<stdio.h> // including stdio.h header file
#include<conio.h> // including conio.h header file

// main function, execution of program starts from here
int main()
{ // start of function block

    // body of main function
    int pr_per_kg = 45; // declaring variable of int type
    float no_of_kg = 7.5; // declaring variable of float type
    float t_pr; // declaring variable of float type
    t_pr = pr_per_kg * no_of_kg; // writing formula as Expression
    printf("%f",t_pr); // displaying the result
    getch(); // getch() function to hold the screen

} // end of function block
```

19:1

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

Explanation

- A simple C program consists of
 - Comments (optional)
 - //
 - /* */
 - Including header files
 - #include<header file name>
 - Functions
 - main function as special function
 - Other user defined functions (optional)
- Let's discuss these in detail..

Comments

File Edit Search Run Compile Debug Project Options Window Help

```
//Problem 1.1: If the price of one kg mango is 45 Rs. then find the price  
//of 7.5 kg mangoes.
```

```
#include<stdio.h> // including stdio.h header file
```

```
// main function  
int main()  
{  
    // body of main function  
    int pr_per_kg = 45; // declaring variable of int type  
    float no_of_kg = 7.5; // declaring variable of float type  
    float t_pr;  
    t_pr = pr_per_kg * no_of_kg; // writing formula as expression  
    printf("Total price is: %f", t_pr); // displaying the result  
    getch(); // getch() function to hold the screen  
}
```

- Two forward slashes '//' (double forward slashes), are used to write single line *comment*
- The next combination '/*.....*/' (forward slash with asterisk) is used for commenting multiple lines
- These comments are not being executed by compiler
- ** comment* can appear anywhere in a program where a white space can appear

19:1

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

Comments

Enhances readability of program

Real life example

Fuel consumption:
16.0 L/100 km

Carbon fiber
engine bonnet



Front
tyre: 255/35
ZR19

C code example

```
//Prog. Name: addition of integers
//Another format
/*Prog. Name: addition of integers
Student Name: Chandra Prakash
UID: 11121415
Section: M4571 */
#include<stdio.h>
#include<conio.h>
int main()
{
}
```

Header files

- The next two lines are command for including header files

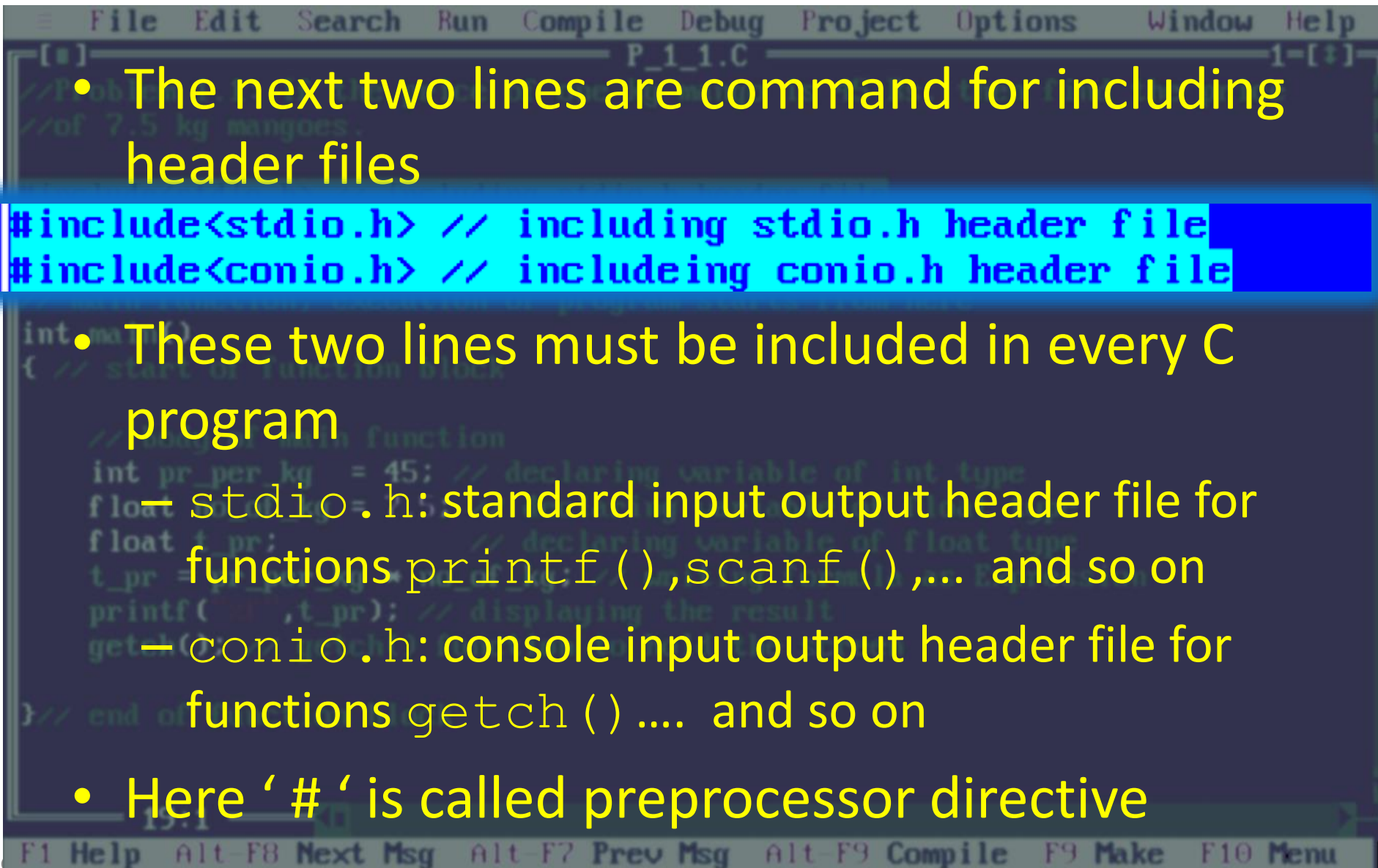
```
#include<stdio.h> // including stdio.h header file
#include<conio.h> // includeing conio.h header file
```

- These two lines must be included in every C program

— `stdio.h`: standard input output header file for functions `printf()`, `scanf()`, ... and so on

— `conio.h`: console input output header file for functions `getch()` ... and so on

- Here '`#`' is called preprocessor directive



Header files

Real life example



**Ferrari
car as
output**

C code example

```
//Sample program
#include<stdio.h> //header file for printf()
#include<conio.h> //header file for getch()
int main()
{
    //stdio.h is providing printf() function
    printf("Car is under process");
    //conio.h is providing getch() function
    getch();
}
```



Output:
Car is under process



Next Class: Components of C Identifier and Keywords Data Type

cse101@lpu.co.in