

Paper 102: Programming & Problem solving through C

Lecture-07:Unit-I C Fundamentals

System softwares

- **Assembler**

- It is a translator that converts assembly language code in the form of directly executable object code (i.e in machine language)

- **Interpreter**

- The interpreter converts source language instructions into executable code in a step by step manner.
- One instruction is converted, executed then the next instruction is taken up for processing.(i.e. line by line)

- **Compiler**

- It converts the entire source language program into executable code.
- The output of a compiler is either in the form of **directly executable object code(in machine language)**, or in the form of an **assembly program** to be run through the assembler.

System softwares

- **Linker**

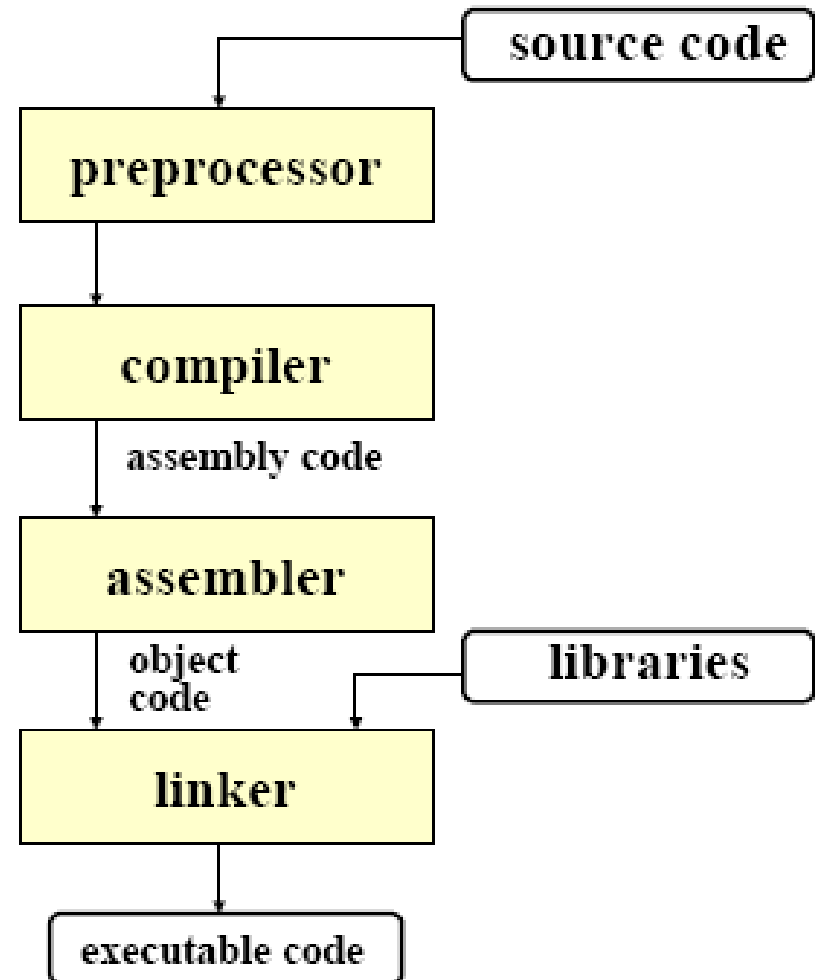
- A linker perform the important task of linking together all the object modules.
- This need arises if the program consist of several object modules.
- They must be linked together to execute as a single program.

- **Loader**

- The task of loading the linked object modules is performed by the *loader*.
- It is of two types:
 - **Absolute loader**:- load the executable code into the memory location specified in the object module.
 - **Relocating loader**:-load the object code in memory locations which are decided at load time.(i.e, any memory location)

The C compilation model

- ❑ The Preprocessor accepts source code as input and
 - removes comments
 - extends the code according to the preprocessor directives included in the source code (lines starting with #)
- ❑ The Compiler takes the output of the preprocessor and produces assembly code
- ❑ The Assembler takes the assembly code and produces machine code (or object code)
- ❑ The Linker takes the object code, joins it with other pieces of object code and libraries and produces code that can be executed



Algorithms

- The word algorithm comes from the name of a Persian mathematician

Abu ja`far Mohamed ibn Musa al Khowarizmi

- It refers to a method that can be used by a computer for describing the solution of a problem
- Criteria for all algorithms:
 - Input:-Zero or more quantities are externally supplied
 - Output:-At least one quantity is produced
 - Definiteness:- Each instruction is clear and unambiguous
 - Finiteness:- The algorithm should terminate after a finite number of steps
 - Correct:-For every input instance it halts with the correct output

Example of an algorithm

Algorithm : Average

This algorithm computes the average of n numbers.

The variables used are: n, number, count, sum: type integer

average: type real

Step 1: [Input the number of data items]

read n

Step 2: [continue processing if n is positive]

if $n > 0$

perform step 3
onwards

else

exit

Step 3: [Initialize]

sum=0

count=0

Step 4: [read and add the numbers]

repeat while $\text{count} < n$

read number

sum=sum + number

count=count + 1

Step 5: [Compute the average]

average=sum/n

Step 6: [finished]

Exit

Pseudocode

- Pseudocode consist of English-like statements describing an algorithm.
- It uses simple phrases and avoid cryptic symbols

Example of Pseudocode

```
1. If student's grade is greater than or equal to 60
    Display "passed"
else
    Display "failed"
```

Convert temperature from Fahrenheit to Centigrade

```
1. Enter the temperature in F
2. Calculate  $5/9 (F-32)$  and assign the result to C
3. Display F and C
4. Ask if the user wants to enter another value
5. Check if the answer is yes then
    Repeat step 1 to 5
Else
    stop
```

Find average of three subjects

```
1. assign total to zero
2. Enter the marks in three subject
3. Add the three marks in total
4. Assign the average to the total divided by 3
5. Display the marks, total and average
```

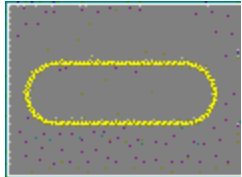

Flowchart

- It is a pictorial representation of an algorithm
- It shows the logic of the algorithm and the flow of control.
- It uses symbols to represent specific actions and arrows to indicate the flow of control.

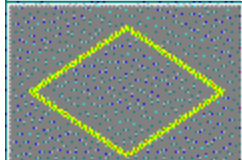
Flowchart symbols

Basic Symbols

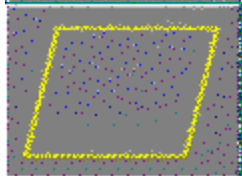
Of the many Symbols available, these *6 Basic Symbols* will be used most:



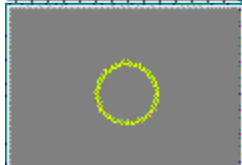
Start/Stop



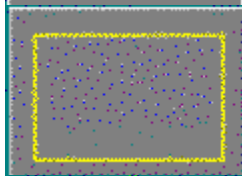
Question, Decision (Use in Branching)



Input/Output



Connector (connect one part of the flowchart to another)

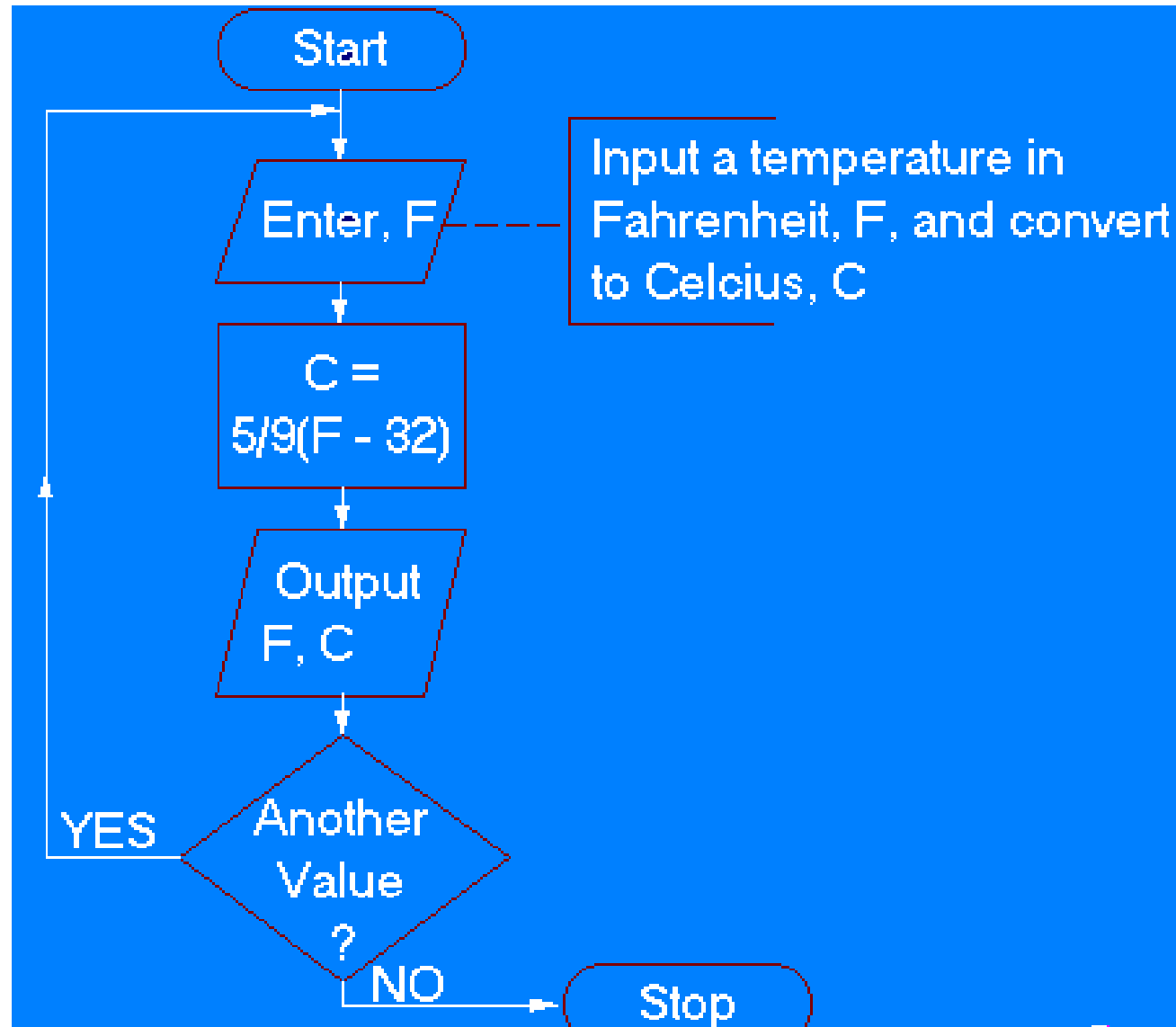


Process, Instruction



Comments, Explanations, Definitions

Flowchart example



Class assignment

- Write the Pseudocode, Algorithm and draw the Flowchart to input the monthly attendance of N students, total class taken, and marks scored in the monthly test. Compute the percentage of attendance. If percentage is less than 75%, 5 marks should be deducted from the marks scored in the monthly test. Display a monthly attendance and marks report of all N students in a proper format.