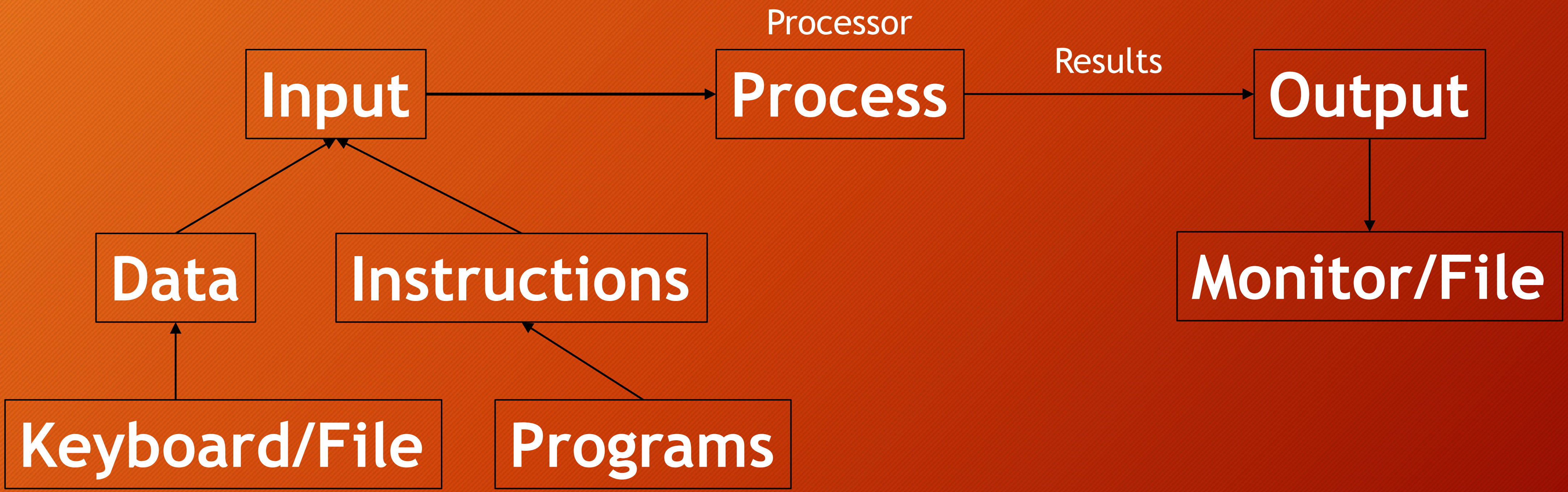
Lecture 1A

What Problem Can Be Solved By Computer



PRE-PROGRAMMING PHASE

- **Analyzing The Problem**
  - Understand and analyze the problem to determine whether it can be solved by a computer.
  - Analyze the requirements of the problem.
  - Identify the following:
    - Data requirement.
    - Processing requirement or procedures that will be needed to solve the problem.
    - The Output.

PRE-PROGRAMMING PHASE

- All These requirements can be presented in a Problem Analysis Chart (PAC) using IPO Chart

Data	Processing	Output
Given in the problem or provided by the user	List of processing required or procedures.	Output requirement.

PRE-PROGRAMMING PHASE

- Problem 1: **Payroll Problem**
  - Calculate the salary of an employee who works by hourly basis. The formula to be used is

Salary = Hour works * Pay rate

Data	Processing	Output
Hours_work Payrate	Salary = Hours_work * payrate	Salary

Problem 2

Write a Problem Analysis Chart (PAC) to convert the length in inches to centimeters where an inch = 2.54 cm.

Data	Processing	Output
Length in Inches	Centimeters = 2.54 x inches	Length in centimeters

Problem 3

- Write a Problem Analysis Chart (PAC) to find an area of a circle where area = pi * radius * radius

Data	Processing	Output
radius	area = 3.14 x radius x radius	area

Problem 4

- Write a Problem Analysis Chart (PAC) to find sum of two numbers

Data	Processing	Output
NUM1, NUM2	SUM = NUM1 + NUM2	SUM

Writing the steps/instructions to solve the Problem

- Creating a blueprint:
    - Like an architect draws a blueprint before beginning construction on a building, similarly the programmer draws logic of the program before writing a program.
- Tool Used:
    - Algorithms
    - Flowcharts

PRE-PROGRAMING PHASE

- **Writing the Algorithm (Pseudocode)**
  - Pseudocode means an imitation computer code.
  - It is a set of instructions (descriptive form) to describe the logic of a program. Thus, It can be used in place of symbols or a flowchart to describe the logic of a program.
  - Pseudocode is close to the actual programming language.
  - Using the Pseudocode, the programmer can start to write the actual code.

Properties of Algorithms

- An algorithm should always have a clear starting and stopping point.
- Inputs and outputs should be defined precisely.
- Each steps in algorithm should be clear and unambiguous.
- Algorithm should be most effective among many different ways to solve a problem.
- An algorithm shouldn't have computer code. Instead, the algorithm should be written in such a way that, it can be used in similar programming languages.

Write an algorithm to add two numbers entered by user.

- Step 1: START
- Step 2: READ/INPUT values num1 and num2.
- Step 3: COMPUTE sum = num1 + num2
- Step 4: DISPLAY/PRINT sum
- Step 5: END

A fruit vendor selling apples wants to know how much sales was done by him in a day

A fruit vendor selling apples wants to know how much sales was done by him in a day

In programming, algorithm are the set of well defined instruction in sequence to solve a program.

Algorithm:

- *STEP 1: Start*
- *STEP 2: Read price, quantity*
- *STEP 3: Sale = price x quantity*
- *STEP 4: Print Sale*
- *STEP 5: End*

An algorithm should always have a clear starting and stopping point.

An algorithm must have finite nsteps numbered to show the sequence of instructions

Write algorithms for the following problems:

1. Write algorithm for Problem 1, 2 and 3 discussed in the class earlier
2. Convert the distance in miles to kilometers where 1.609 kilometers per mile.
3. To compute area of a square.
4. To compute area of a rectangle
5. To compute the percentage of a student based on his score in 5 subjects scored out of 100 for each subject
6. Read name of the user and display a welcome message to the user

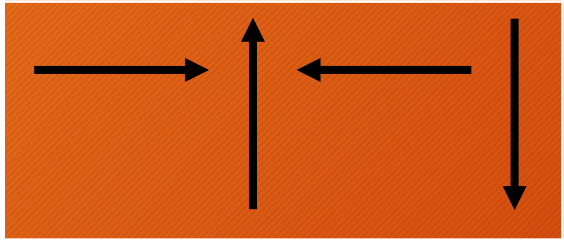
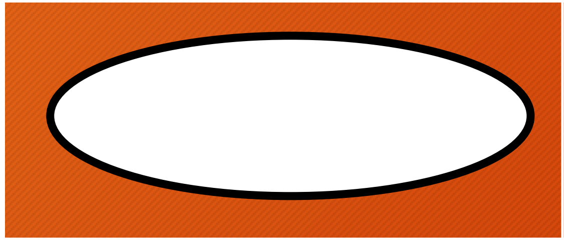
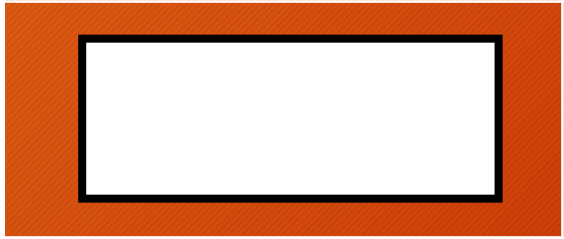
PRE-PROGRAMING PHASE

- **Program Flowcharts**
  - Flowchart is the graphic representations of the individual steps or actions to implement a particular module.
  - Flowchart is independent of any programming language.
  - Flowchart is the logical design of a program.
  - It is the basis from which the actual program code is developed.

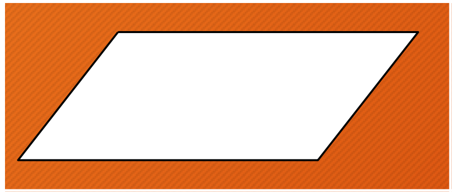
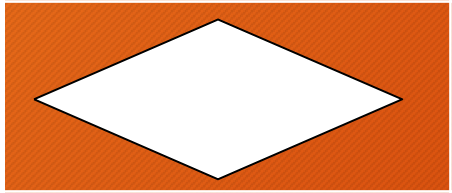
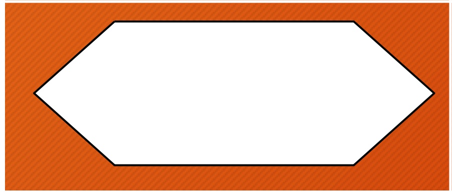
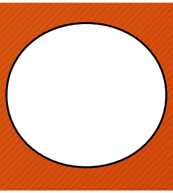
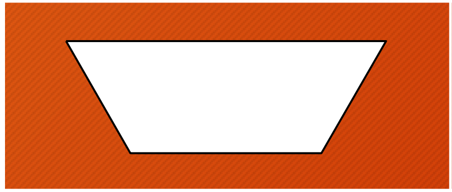
PRE-PROGRAMMING PHASE

- Flowchart serves as documentation for computer program.
- The flowchart must be drawn according to definite rules and utilizes standard symbols adopted internationally.
- The International Organization for Standardization (IOS) uses the symbols shown below

PRE-PROGRAMMING PHASE : Flowchart Symbols

Symbol	Function
	Show the direction of data flow or logical solution.
	**Indicate the beginning and ending of a set of actions or instructions (logical flow) of a module or program.**
	**Indicate a process, such as calculations, opening and closing files.**

PRE-PROGRAMMING PHASE

	**Indicate INPUT to the program and OUTPUT from the program.**
	**Use for making decision. Either True or False based on certain condition.**
	Use for doing a repetition or looping of certain steps.
	**Connection of flowchart on the same page.**
	Connection of flowchart from page to page.

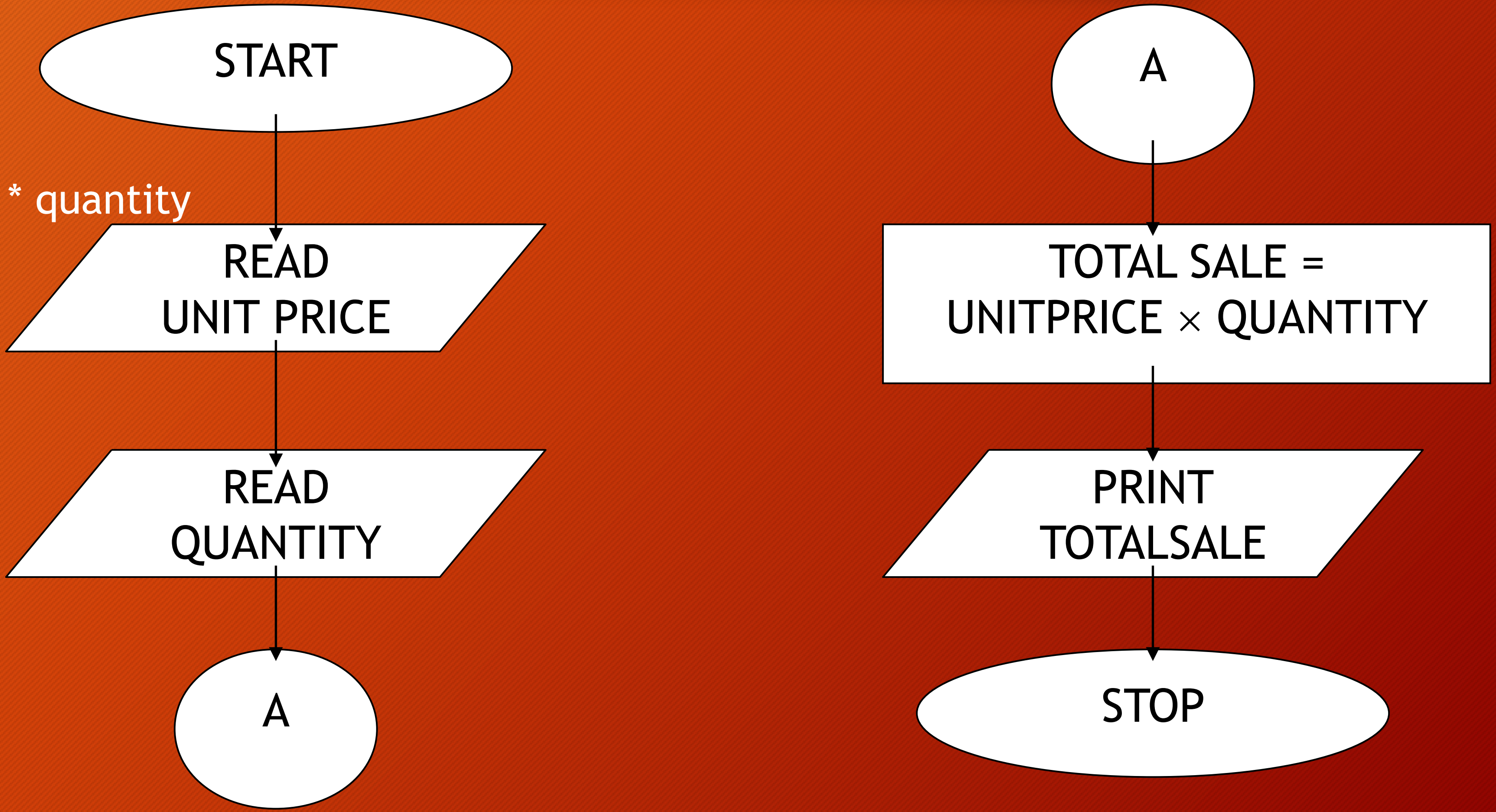
PRE-PROGRAMMING PHASE

- **Example 2.3 : Sale Problem**
  - Draw a flowchart for a problem that to read two numbers.
  - The first number represents the unit price of a product and the second number represents the quantity of the product sold. Calculate and print the total sale.
  - Solution: Stepwise Analysis of the Sale Problem
    - Start of processing
    - Read the unit price
    - Read the quantity
    - Calculate total sale
    - Print total sale
    - Stop the processing

PRE-PROGRAMMING PHASE

Compute total sale of a fruit vendor who sells apples.

- Step 1: START
- Step 2: READ unitprice
- Step 3: READ quantity
- Step 3: COMPUTE sales = price * quantity
- Step 4: DISPLAY sales
- Step 5: STOP



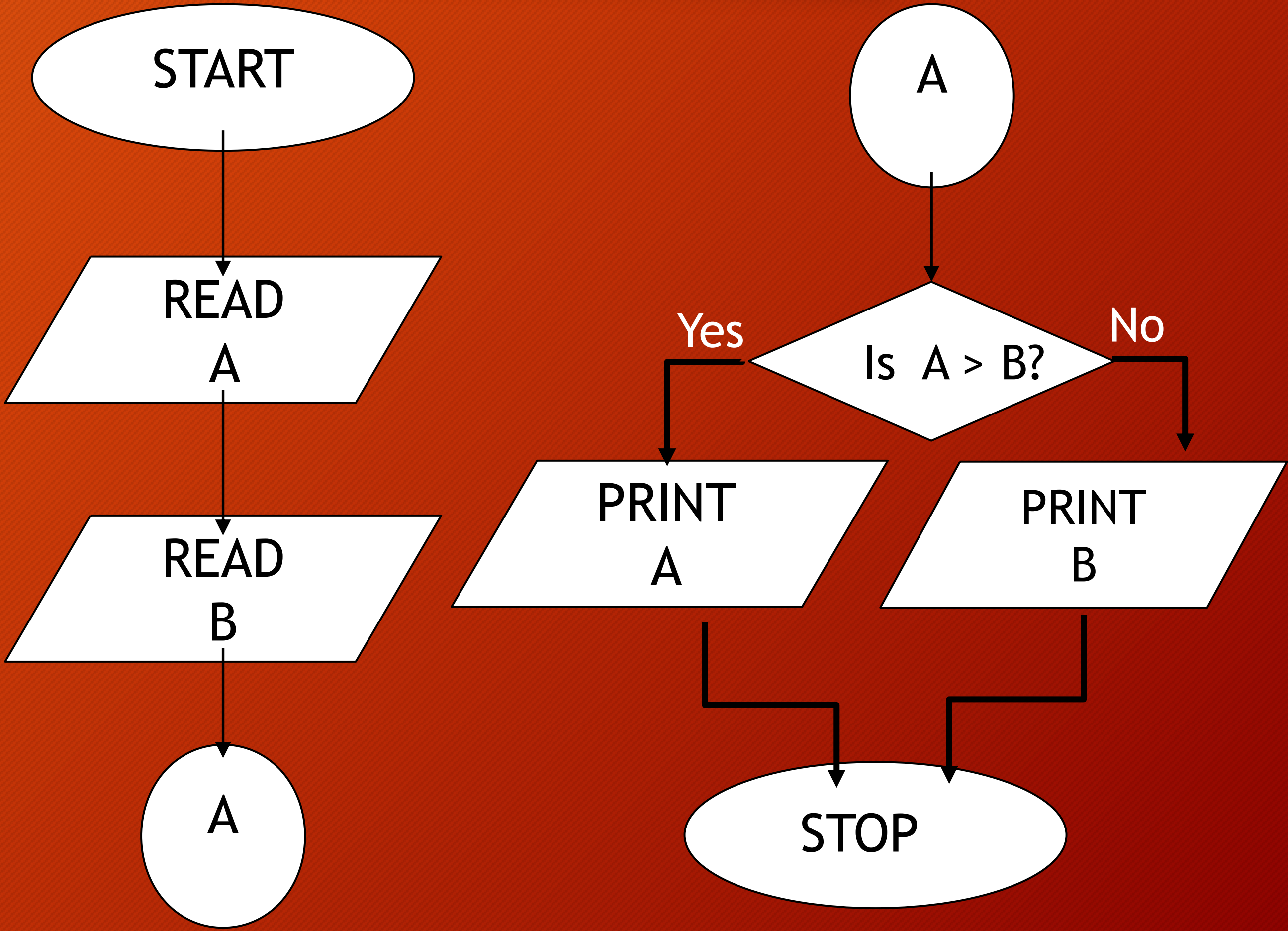
Draw a Flowchart to find the largest among two numbers entered by user.

Draw a Flowchart to find the largest among two numbers entered by user.

- Step 1: START
- Step 2: DECLARE variables a and b.
- Step 3: READ variables a and b.
- Step 4: If a>b

```
        Then
                DISPLAY a
        else
                DISPLAY b
```

- Step 5: STOP



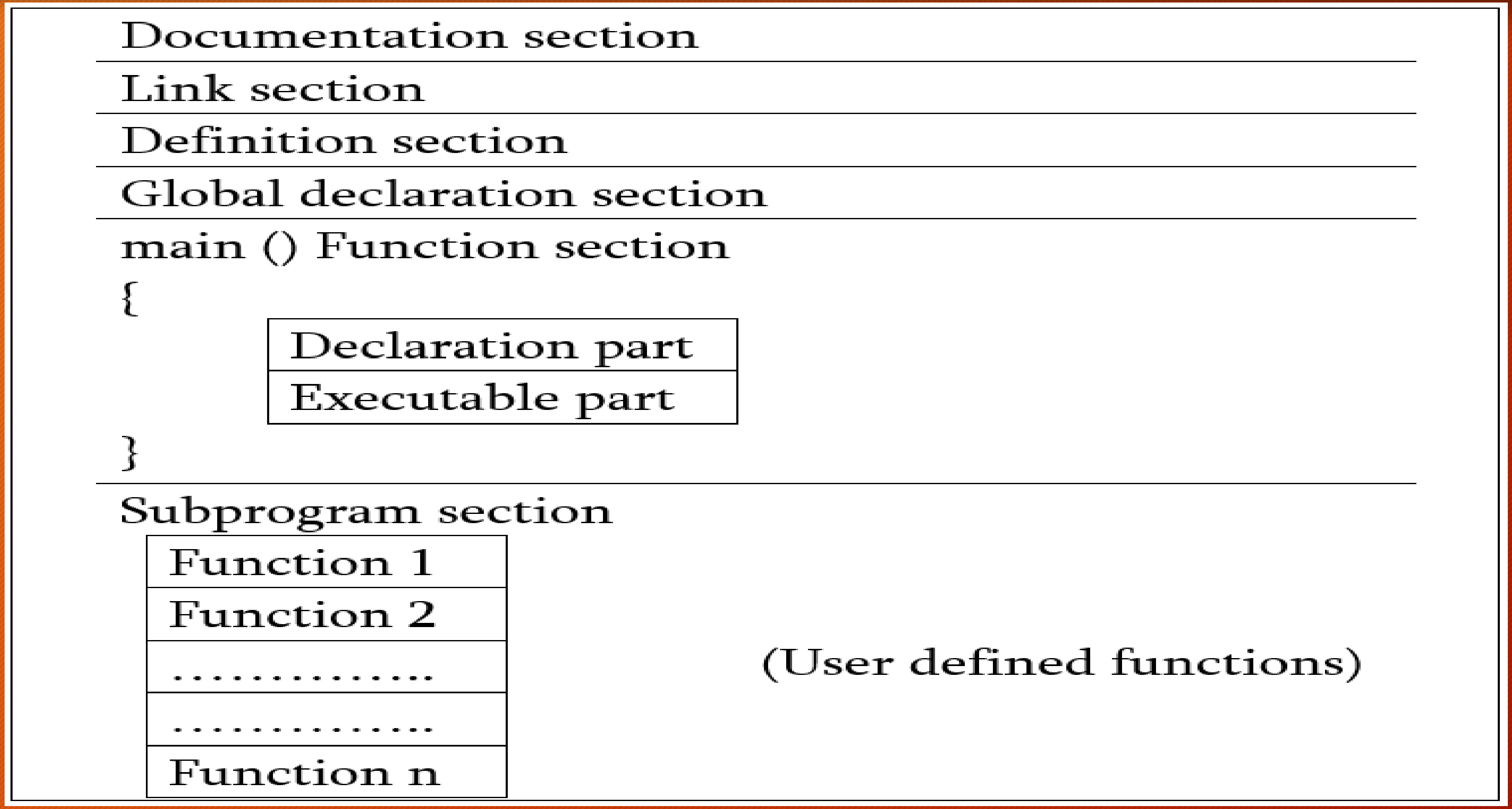
Basics of ‘C’

General Aspect of ‘C’

- C was originally developed in the 1970s, by Dennis Ritchie at Bell Telephone Laboratories, Inc.
- C is a
  - **High/Middle level**
    - C contains certain additional features that allows it to be used at a lower level , acting as bridge between machine language and the high level languages.
  - **General-purpose**
  - **Structured**

programming language.

Basic Structure of C Program



Documentation section

Link section

Definition section

Global declaration section

main () Function section

{

Declaration part

Executable part

}

Subprogram section

Function 1

Function 2

....

....

Function n

(User defined functions)

Following is the basic structure of a C program.

Documentation	Consists of comments, some description of the program, programmer name and any other useful points that can be referenced later.
Link	Provides instruction to the compiler to link function from the library function.
Definition	Consists of symbolic constants.
Global declaration	Consists of function declaration and global variables.

main( )
{
}
	Every C program must have a main() function which is the starting point of the program execution.

Subprograms	User defined functions.