Introduction

Course Objectives

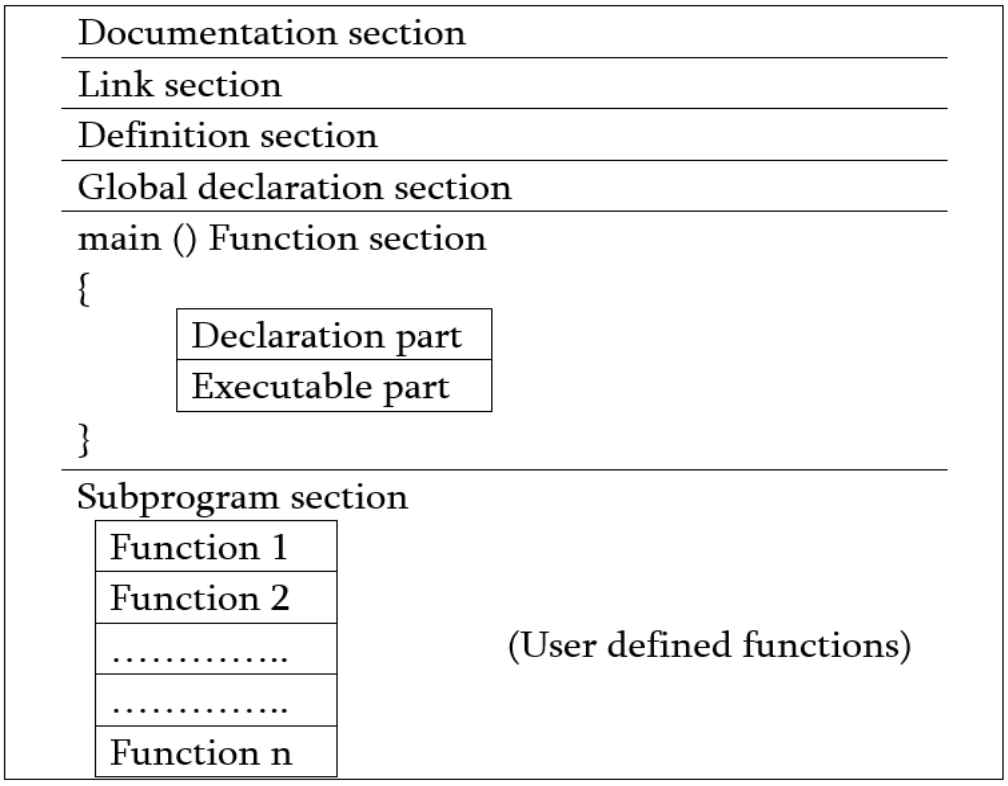
- The aim of the course is to provide exposure to students on developing programs for solving real-world applications
- Given a computational problem, the learners identify and abstract the programming task involved and also get to learn the syntax and semantics of the C language
- In this course, the learners get hands-on experience to write programs, edit, compile, debug and verify the results

Basic Aspects of 'C'

At Bell Laboratories, Dennis Ritchie was developed the programming language C. Initially, this language programming language was developed for writing application software; nowadays it becomes such popular and basic language. This language is very high level structured and general-purpose programming language.

C programming language can be easily used on different types of computers. There are other programming languages which are C++ and Java are based on C language that means you can learn these languages easily in the future. In recent days, C language has been widely used on the UNIX operating system. C language is built for performance and memory management: embedded systems, operating systems, real-time systems, communication systems.

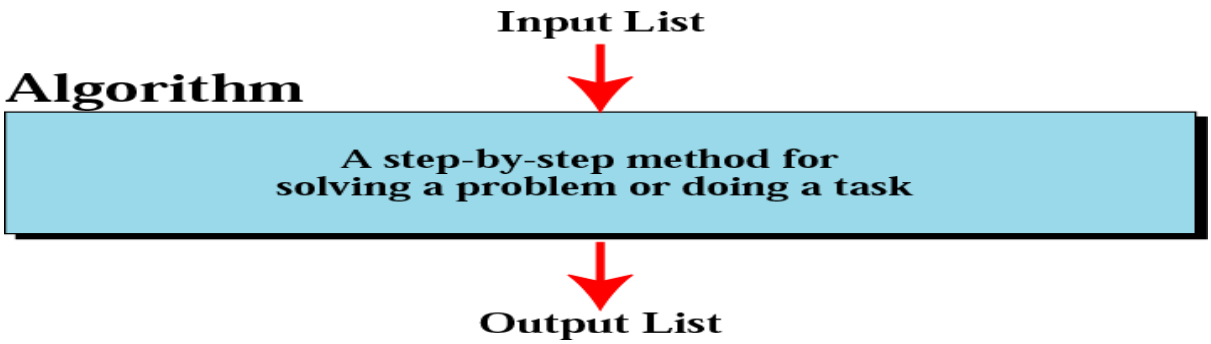
Basic Structure of C Program



Sample Program:

```
/* Simple program in C */

#include<stdio.h>

int main()

{
  printf("\n Introduction to Programming in C ");
  return 0;

}
```

Algorithm:



- An algorithm is a step-by-step process which is used in solving mathematical or computational problems
- An algorithm can be expressed in any language including natural language, programming language, pseudocode
- It can be visualized using a flowchart
- Tools used to represent algorithms: flowchart and pseudocode

Example 1: Algorithm to add two numbers

Step 1: Start
Step 2: Declare variables num1, num2 and sum
Step 3: Read values num1 and num2
Step 4: Add num1 and num2 and assign the result to sum
sum←num1+num2
Step 5: Display sum
Step 6: Stop

Example 2: Find the average of three numbers

step 1: Get the input of three real numbers and store in a, b, c

step 2: Calculate sum <- a + b + c

step 3: Calculate average <- sum / 3

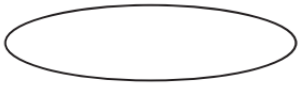
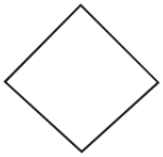
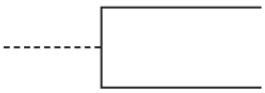
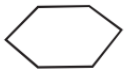
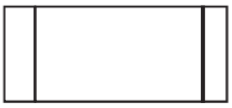
step 4: Print sum and average

Flowchart:

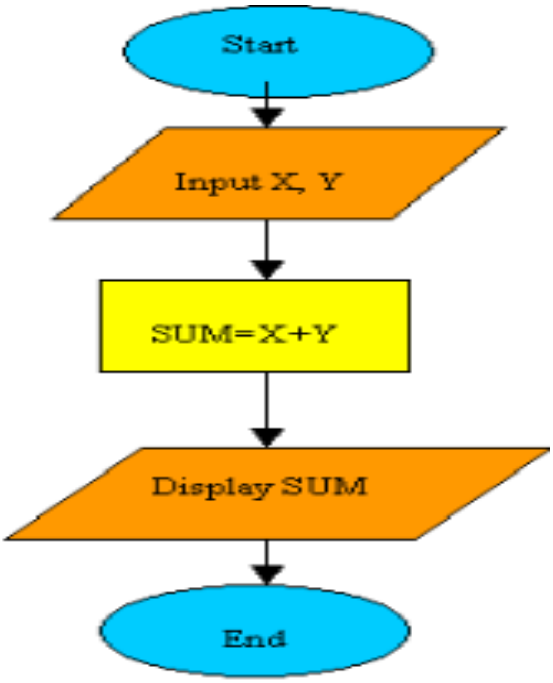
- A flowchart gives a step-by-step procedure in sequential order and present the workflow of an algorithm
- To specify the method of solving a problem
- Flow chart Communicate ideas and solutions
- A flowchart is a graphical representation of the problem-solving process
- To plan the sequence of a computer program

The most commonly used symbols to construct a flow chart.

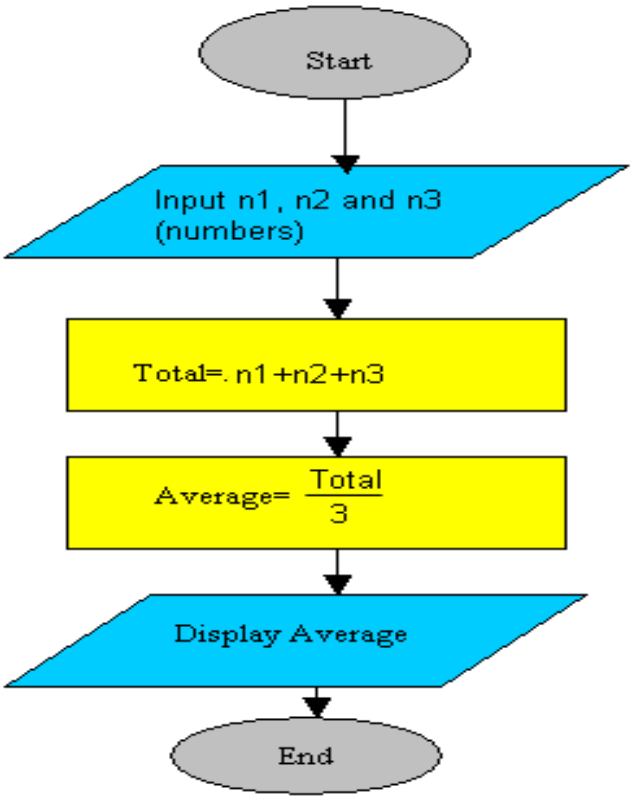
The following shapes are often used in flowcharts:

	An oval indicates beginning or end of a program.
	A parallelogram is a point where there is input to or output from the program.
	A rectangle indicates the assignment of avalue to a variable, constant, or paramter. the assigned value can be the reult of a computation. The computation would also beincluded in the rectagle.
	A diamond indicates a point where a dicision is made.
	An open-ended rectangle contains comment statements. The comment is connected to the program flow via adashed line.
	A hexagon indicates the beginning of a repitition.
	The double-lined rectangle indicates the use of an algorithm specified outside the program, such as asubroutine.
	Circles can be used to combine flow lines.
	Arrows indicate the direction and orderof program execution.

Example 1: To find the sum of two numbers



Example 2: To find the average of three numbers



Compile and Execute Program

Text Editor and C Compiler

STEPS

- Create the program
- Compile the program
- Link the program with functions that are needed from the C library
- Run the program

Compile and execute the program

✓ Program Name: sample.c

✓ GNU C Compiler: gcc sample.c

✓ Execution by default: a.out

REFERENCES

1. Byron S Gottfried, “Programming with C”, Schaum’s Outlines, Second Edition, Tata McGraw-Hill, 2006.

2. Reema Thareja, “Programming in C”, Oxford University Press, Second Edition, 2016

3. Kernighan, B.W and Ritchie,D.M, ―The C Programming language, Second Edition, Pearson Education, 2006.