Unit No.	Course Content	No. of Hours
1.	**Problem Solving Concepts:** Introduction to Algorithms and Flowcharts, Exchanging the values of Two Variables, Counting, Summation of a Set of Numbers, Factorial Computation, Sine Function Computation, Generation of the Fibonacci Sequence. **Overview of C:** Basic structure of a C program, executing a C Program, Character Set, C Tokens, Keywords and Identifiers, Constants, Variables, data types, Declaration of variables, Assigning values to variables, Defining Symbolic Constants, Formatted Input and Formatted Output.	08
2.	**Operators:** Introduction, Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operators, Conditional Operator, Bitwise Operators, Conditional Operator, Special Operators: Comma and 'sizeof' operator. **Expressions:** Arithmetic Expressions, Evaluation of Expressions, Precedence of Arithmetic Operators, Type Conversions in Expressions, Operator Precedence and Associativity.	06
3.	**Control statements:** Introduction, Decision Making with IF Statement, Simple IF Statement, the IF...ELSE Statement, Nesting of IF Statements, The ELSE…IF Ladder, The Switch statement, Use of 'goto' statement. **Loops:** The WHILE Statement, The DO – While Statement, the FOR Statement, Nesting of Loops, Jumps in Loops: break & continue statement. exit statement.	08

Programming for Problem Solving (2020CS110)
Credits: 3: 0: 0

Hours: 39 Hours (13 Weeks * 3 Hours= 39 Hours)

Tutor:
Dr. Srinath.S, Associate Professor, Dept. of Computer Science and Engineering,
Sri Jayachamrajendra College of Engineering, JSS S&TU, Mysuru- 6

Email: srinath@sjce.ac.in

CIE – Continuous Internal Evaluation – 50 Marks
(5 Events, Weightage 10 Marks each, so 5*10 = 50 Marks)
Event 1 – Test 1 (Scheduled at the College Level)
(Eligibility to take up Test 1: First block attendance should be >=60%)

(First block is from day 1 till one day before the commencement of first test)

Event 2 – Quiz 1 (Will be scheduled by the tutor)

Event 3 – Test 2 (Scheduled at the College Level)
(Eligibility to take up Test 2: Second block attendance should be >=60%)
(Second block is from day 1 after first test till one day before the commencement of second test)

Event 4 – Quiz 2 (Will be scheduled by the tutor)
Event 5 – Test 3 (Scheduled at the College Level)

Eligibility to take up SEE:

- **Should have scored minimum of 50% marks in the CIE, i.e 25 Marks**
- **Should maintain 85% attendance**

SEE – Semester End Examination - 50 Marks
Examination for 100 Marks, will be reduced to 50Marks
(Minimum score to pass is 40 out of 100)

Total Marks: 100 (50 CIE + 50 SEE)
>=90 and above – S grade
>=75 and <90 - A grade

SEE QPP
Two parts
All the question is for 10 Marks : (10 * 10 =100 Marks)

Part A (all are compulsory)

Compulsory
1 (Unit 1)
2 (Unit 2)
3 (Unit 3)
4 (Unit 4)
5 (Unit 5)

Part B (will have internal choice)

6 or 7 (Unit 1) : answer any one.
8 or 9 (Unit 2) : answer any one
10 or11 (Unit 3) : answer any one
12 or 13 (Unit 4) : answer any one

14 or 15 (unit 5) : answer any one.

Programming for Problem Solving (2020CS110): 3 Credits Marks: 100

Tutor:
Dr. Srinath.S, Associate Professor, Dept. of Computer Science and Engineering,
Sri Jayachamrajendra College of Engineering, JSS S&TU, Mysuru- 6

UNIT – 1 (Syllabus)

Introduction to Problem Solving:
Introduction to computer concepts: Block diagram of a computer, Define: Hardwar, Software, Operating System, Text Editor.

Computer Language classification: MLL, ALL, HLL, Assembler, Compiler, Interpreter
Introduction to algorithm, different symbols used in flowchart and converting an algorithm to flowchart.

Overview of C:
History of C, Importance of C, Basic structure of a C program, executing a C Program, Character Set, C Tokens, Keywords and Identifiers, Constants, Variables, data types, Declaration of variables, assigning values to variables, Defining Symbolic Constants, Formatted Input and Formatted Output

---

Introduction to Problem Solving:

Introduction to Computer Concepts:

Define Computer:

Computer is a calculating machine, which accepts data through input device, processes it using processing device and give the result through output device. Computer is an electro-mechanical machine. It consists of electronic parts (IC'S) electrical (motors, fan) and Mechanical parts (Outer cover, keyboard).

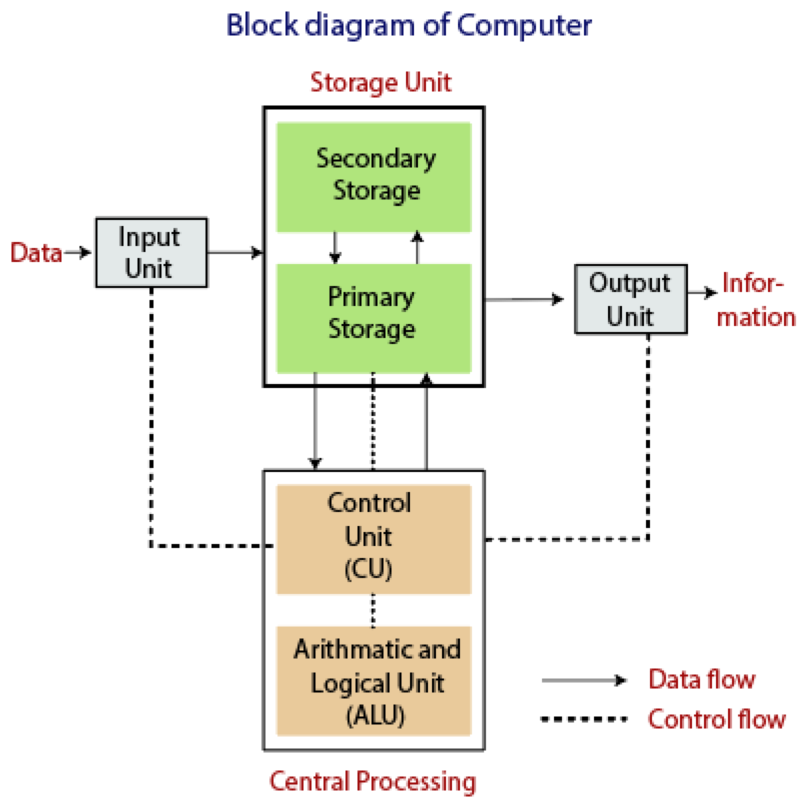
Characteristics of Computer:

The following are the characteristics of a computer:

1. Speed: Computer can process the data at an extremely high speed.
2. Accuracy: Computer will produce accurate result.
3. Reliability: It is the measurement of performance of computer.

4. Versatile: Computer can be used for different purpose.
5. Storage capacity: computer can store large amount of data.

Block Diagram of a Computer:



Computer has five fundamental units:

1. Input unit: Through which data will be fed into the computer

   Example: Keyboard, Mouse, Microphone, Joystick, Web camera

   Keyboard is the standard input unit.
2. Storage unit: It is used to store the data/ Information.

   Two types:

   (i) Primary Storage Unit / Primary Memory

   Example: Random Access Memory (RAM)

   (ii) Secondary Storage Unit/ Secondary Memory

   Example: Hard Disk Drive (HDD)/ Solid State Device (SSD)

3. Control Unit: Controls all other units.
4. Arithmetic and Logical Unit: Does the computation.

   Note: Control Unit and Arithmetic and Logical unit put together called as

Central Processing Unit or CPU. A single chip CPU is called Microprocessor.

5. Output Unit: Through which result will be displayed to the external world
   Example: Monitor, Speaker, Projector, Printer
   Monitor is the standard output unit.
   Output generated using printer is called Hardcopy output.

Hardware: All the physical parts of the computer is called hardware. You can touch and feel. Example: Keyboard, Mouse, outer cover, IC's

Software: Programs written for the computer is called software. Ex: MS-Word, C program, Windows 10 OS

Operating System: It is the interface between user of the computer and computer hardware.

Text Editor: It is an application program which allows user to type text and edit it.

Computer Languages:

Machine Level Language (MLL) : Binary language it use 1 and 0. It is the only language understood by the computer.

Assembly Level Language (ALL) : It uses short notations like add, sub, mul and div etc.
It cannot be understood by the computer. Hence ALL needs to be converted into MLL
Assembler: is the program, which converts ALL to MLL and vice versa.

Example for ALL: ALL 8085, ALL 8086

High Level Language (HLL) : It is English like language. It is easy to understand by the programmer.
It cannot be understood by the computer.
HLL needs to be converted into MLL.

Compiler or Interpreter will be used to convert HLL to MLL and vice versa.

Compiler: converts HLL to MLL in one step.

Interpreter: Converts HLL to MLL line by line.

Example: C,C++, Java

Algorithm: It is the step-by-step procedure to solve a given problem. It will accept input, processes and gives output. It is simple to represent and **it is independent of programming language.**

Example:

I. Write an algorithm to exchange the contents of two variable.

```
1. [Start]
2. [Input two numbers]
   Read(x,y)


3. [Process two swap two numbers]
   t <- x
   x <- y
   y <- t

4. [Output – to print the swapped numbers]
   Write (x,y)
5. [Stop]
```

II. Write an algorithm to find the area of a triangle given base and height.

```
1. [Start]
2. [Input – Base and height of the triangle]
   Read(b,h)
3. [Process – to compute the area of the triangle]
   a <- 0.5*b*h
4. [Output the area]
```

```
    Write (a)
5.  [Stop]
```

III. Write an algorithm to find the area of a triangle given its 3 sides.

```
1.  [Start]
2.  [Input – 3 sides of the triangle]
    Read(a,b,c)
3.  [Process – to compute half perimeter]
    s<- (a+b+c)/2.0
4.  [Process to compute the area ]
    a  <- sqrt( s * (s-a) * (s-b) * (s-c) )
5.  [Output the area]
    Write (a)
6.  [Stop]
```

IV. Write an algorithm to compute the area of a circle

```
1.  [Start]
2.  [Input – radius of the circle]
    Read(r)
3.  [Initialize]
    p <- 3.14
4.  [Process – to compute the area of the circle]
    a <-  (p * r *r)
5.  [Output the area]
    Write (a)
6.  [Stop]
```

V. Write an algorithm to check whether the given number is odd or even

```
1.  [Start]
2.  [Input a number]
    Read(n)
```

```
3. [Compute the reminder]
   r  <-  n modulo 2
4. [ process to check whether the given number is odd or even using
   reminder value]
   If (r = 0)
           Write (“Given number is Even”)
   Else
           Write (“Given number is Odd”)
5. [Stop]
```

VI. Write an algorithm to compute the sum of first ‘n’ natural numbers

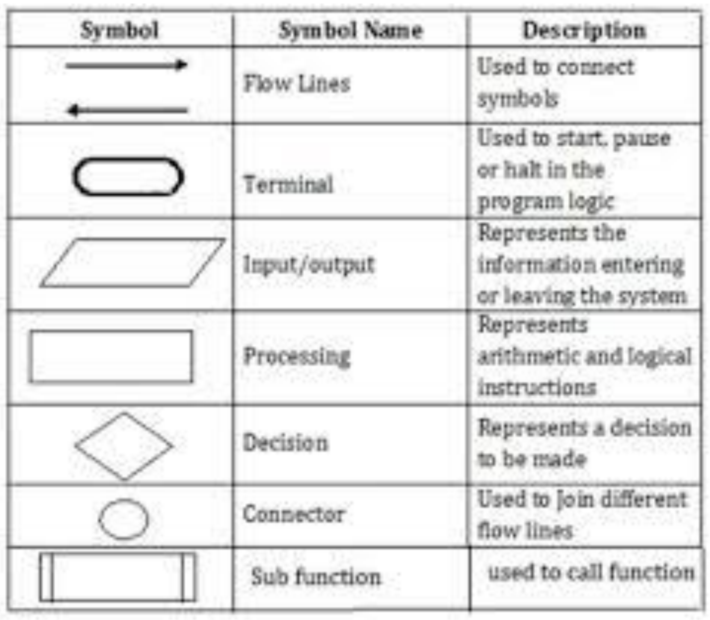
```
1. [Start]
2. [Input a number]
   Read(n)
3. [Initialize]
   sum < - 0


4. [compute the sum of first ‘n’ numbers ]
   for  i <- 1 to n   step 1
           sum <- sum + i
   end for
5. [Display the sum]
   Write(sum)

6. [Stop]
```

VII. Write an algorithm to compute the factorial of the given number

```
1. [Start]
2. [Input a number]
       Read(n)
3. [Initialize]
       fact < - 1
```

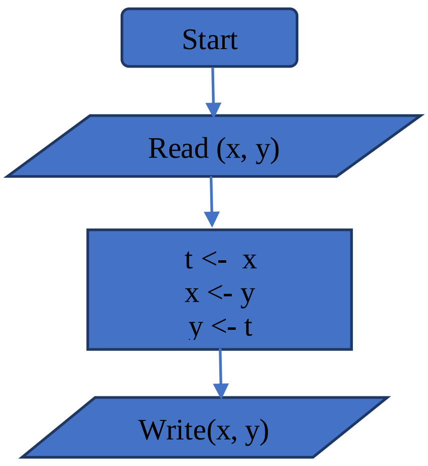
```
4.  [compute the factorial of 'n' ]
        for  i <- 1 to n   step 1
                fact <- fact* i
        end for

5.  [Display the sum]
        Write(fact)

6.  [Stop]
```

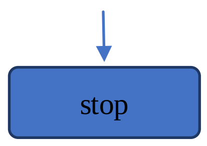
Flow chart: It is the pictorial representation of an algorithm. All the steps are drawn in the form of different shapes. Commonly used symbols for flow chart are :



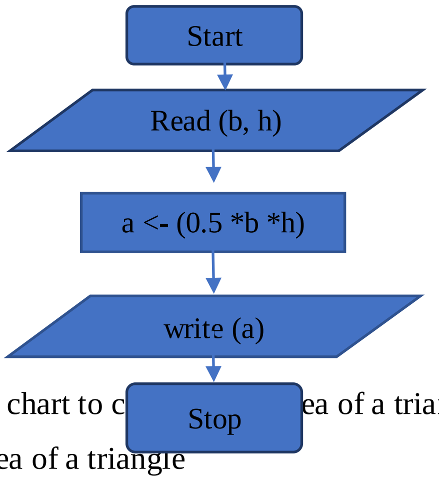
Symbol	Symbol Name	Description
→ ←	Flow Lines	Used to connect symbols
⬭	Terminal	Used to start, pause or halt in the program logic
▱	Input/output	Represents the information entering or leaving the system
▭	Processing	Represents arithmetic and logical instructions
◇	Decision	Represents a decision to be made
○	Connector	Used to join different flow lines
▯	Sub function	used to call function

I. Write a flowchart to exchange contents of two variables:

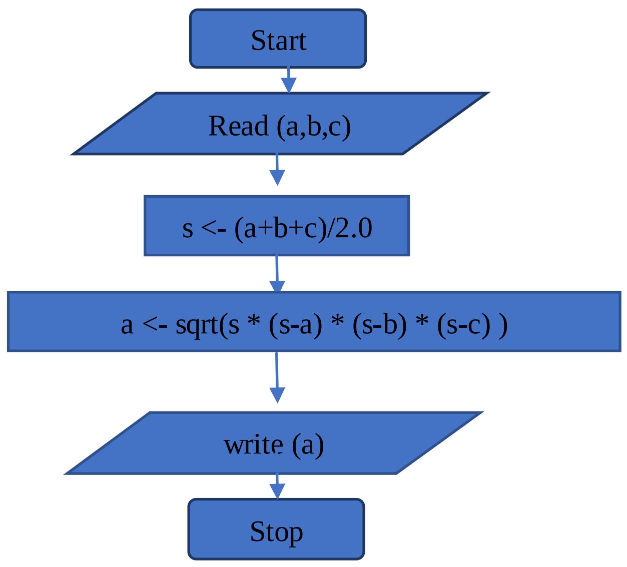




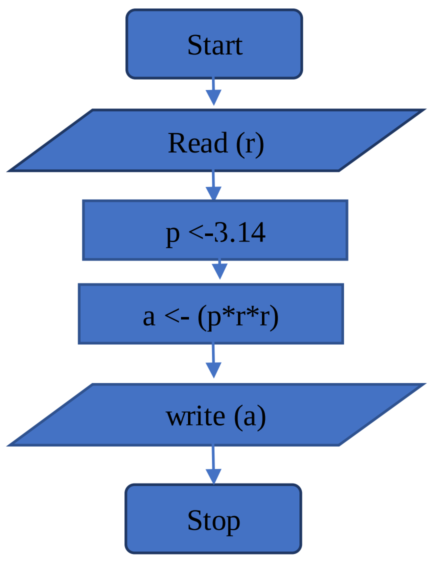
2. Write a flowchart to compute the area of a triangle



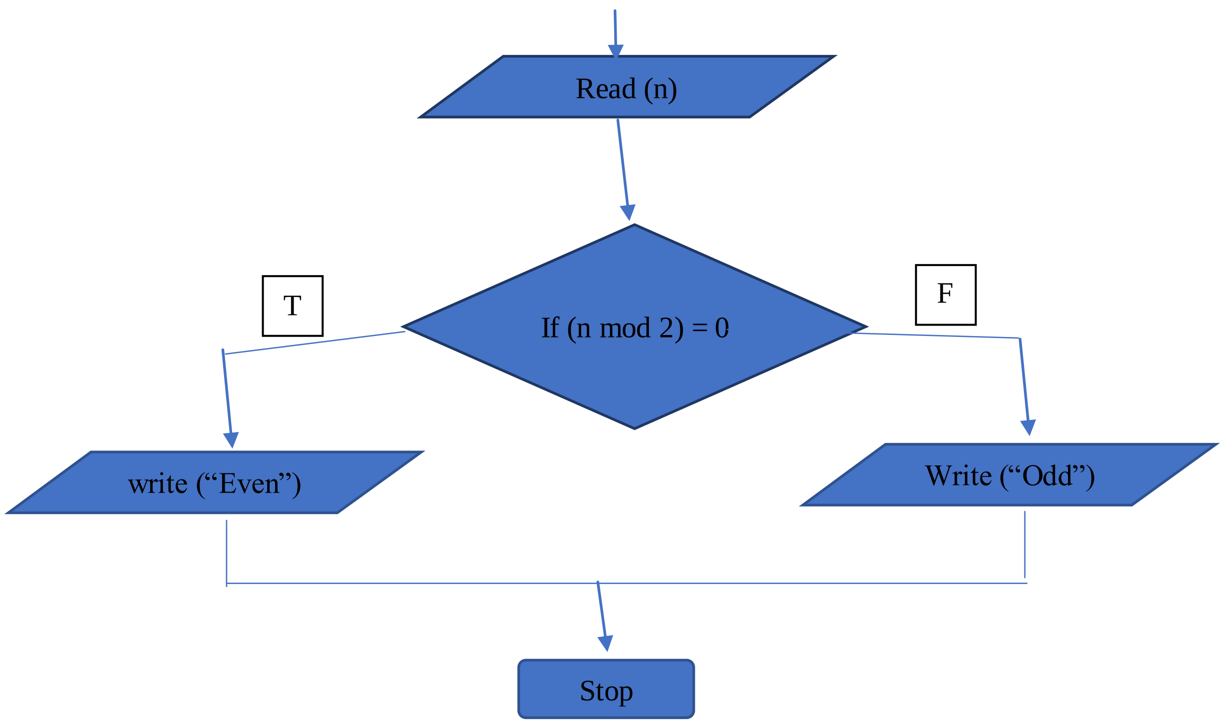
3. Write a flow chart to c ea of a triangle given 3 sides.
compute the area of a triangle



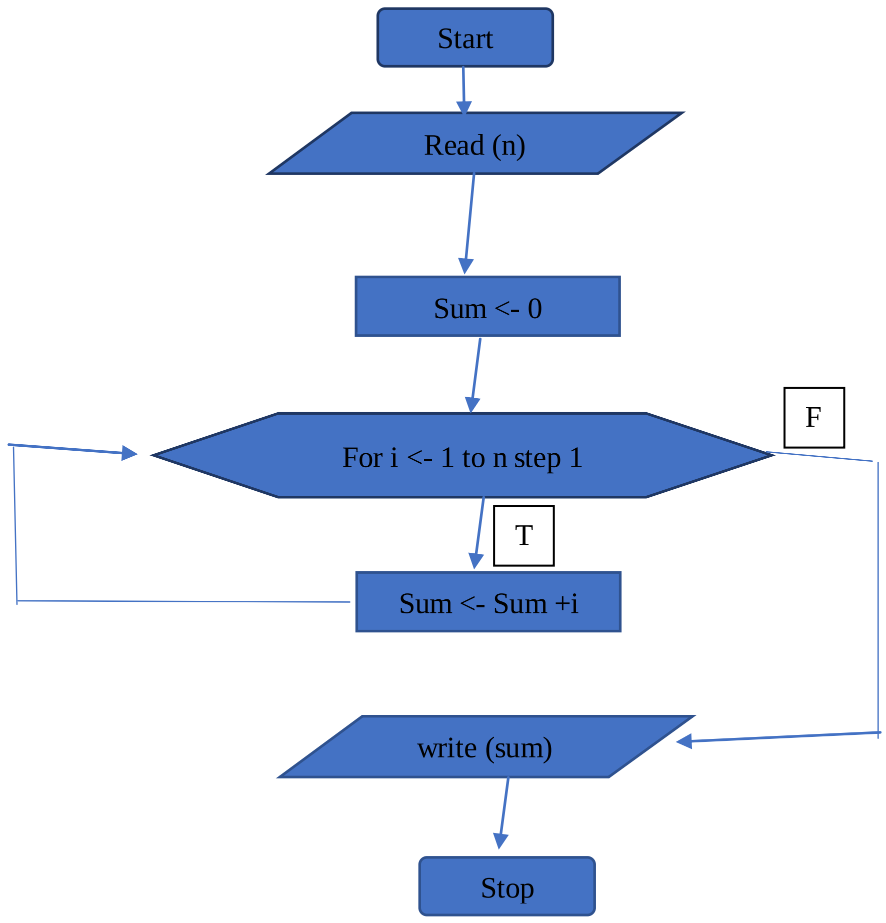
4. Write a flowchart to find the area of a circle



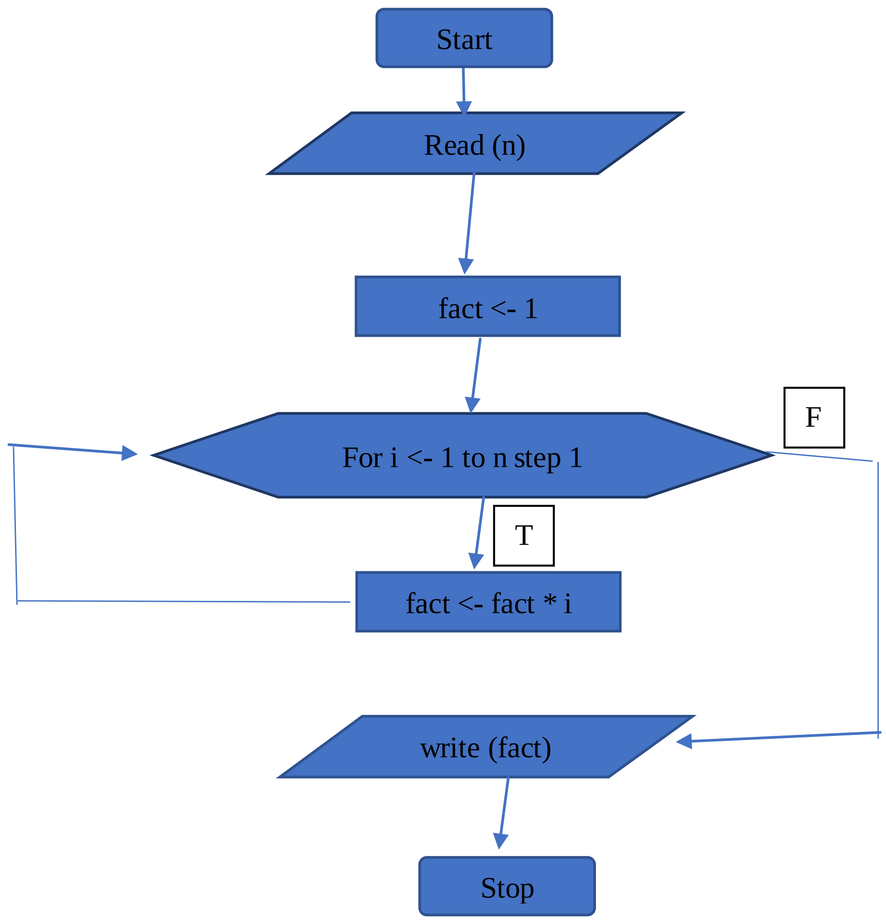
5.Write a flowchart to check whether the given number is odd or even



6.Write the flow chart to compute the sum of first ‘n’ natural numbers



7.Write a flowchart to compute the factorial of a given number



Overview of C:

Syllabus:

Basic structure of a C program, executing a C Program, Character Set, C Tokens, Keywords and Identifiers, Constants, Variables, data types, Declaration of variables, Assigning values to variables, Defining Symbolic Constants, Formatted Input and Formatted Output.

Basic Structure of a C Program:

C program can be viewed as a collection of functions. A C program may contain one or more sections as shown in the figure.

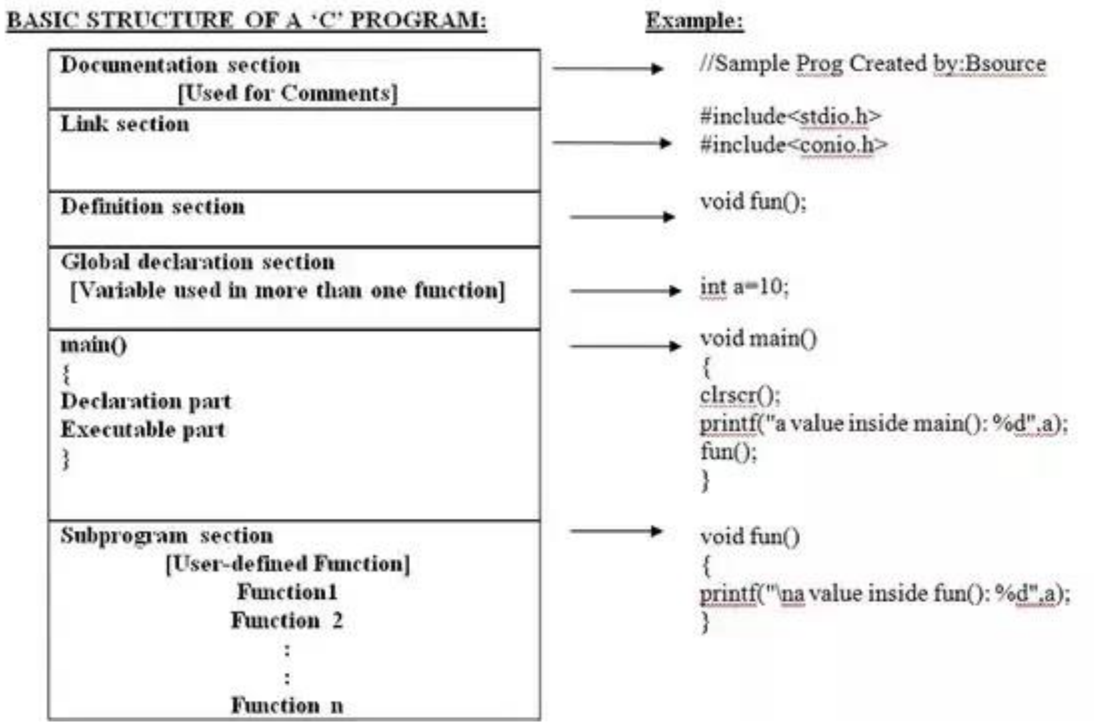
The document section consists of set of **comment line**s giving the name of the program. It can contain information about the program, author, date of writing the program etc. It is very popularly known as comment statement. It is enclosed between /* … and …..*/

The link section provides instructions to the compiler to link functions from the system library.

In this section, line begins with # symbol.

Some variables which are used in one or more functions are placed in the global declaration

section.
Every C program must have one main() function. This section contains two parts, declaration section and executable part. The declaration part declares all the variables used in the executable part. This is followed by executable part. These two parts are placed between opening and the closing braces. The program execution begins at the opening braces and ends at the closing braces. All the statements in the declaration and executable section ends with semicolon (;).
The subprogram section contains all the user defined functions that are called in the main function. Used defined functions are usually placed immediately after the main function.
=



```
//Sample Prog Created by:Bsource

#include<stdio.h>
#include<conio.h>

void fun();

int a=10;

void main()
{
clrscr();
printf("a value inside main(): %d",a);
fun();
}

void fun()
{
printf("\na value inside fun(): %d",a);
}
```

Executing a C Program:

Executing a C program involves a series of steps. These are:

1. Creating the program
2. Compiling the program
3. Linking the program with functions that are needed from the C library
4. Executing the program.

Character Set:

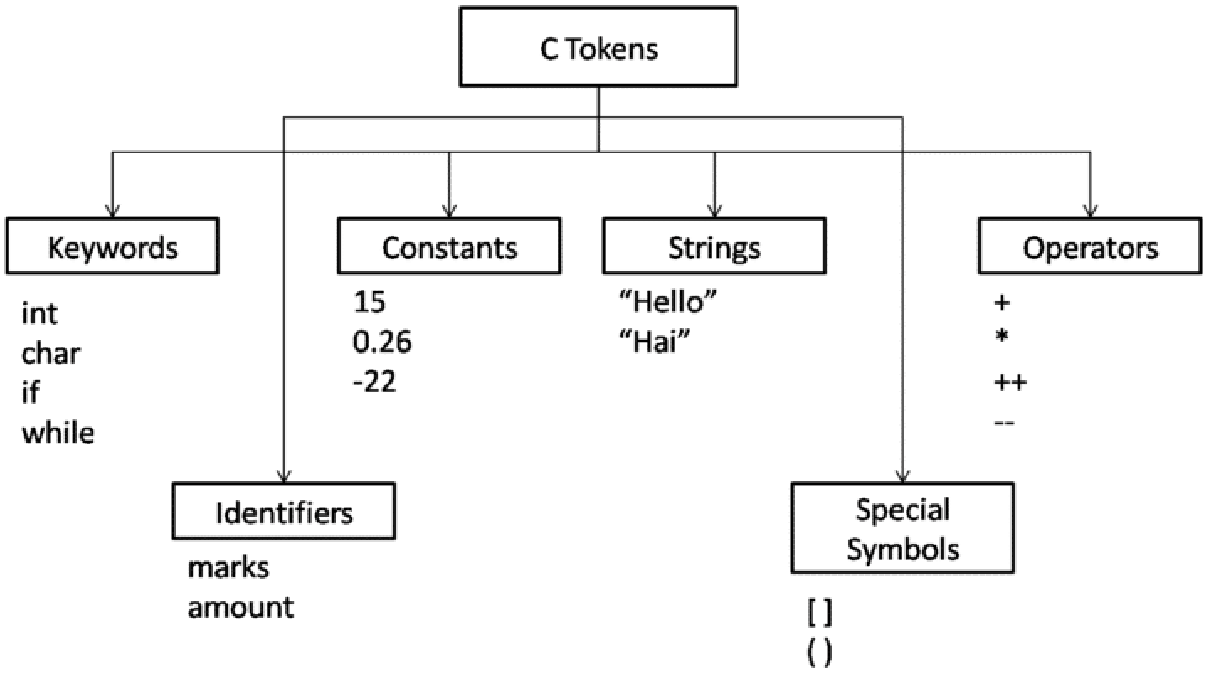
The characters in C are grouped into the following:

1. Letters : A…Z and a…z

2. Digits : 0-9
3. Special characters : Example: #,&,;…
4. White spaces.

C Tokens:

C has six types of tokens as shown in figure below:



Keywords

Keyword is also called as reserved words.

All keywords have fixed meaning.

Keywords cannot be used as Identifier.

ANSI 'C' has 32 keywords

Example: if, else, int, float

Identifier:

Identifier refer to the names of variables, functions and arrays.

These are user defined names and consists of sequence to letters and digits.

The only special character allowed is underscore (_).

First character of the identifier cannot be digit.

Length of the identifier can be upto 31 characters.

It cannot contain white space (blank space)

Keywords cannot be used as identifier.

Both upper case and lowercase letters are allowed.

Example for valid identifier:

X, x, x_y, s1, add_sum, s100, intx

Example for Invalid identifier:

1x : invalid because it starts with number

int : invalid because ‘int’ is a reserved word

x@: invalid because the special character @ not allowed.

Constants:

Constants in C refer to fixed values that do not change during the execution of a program.

C supports different types of constant:

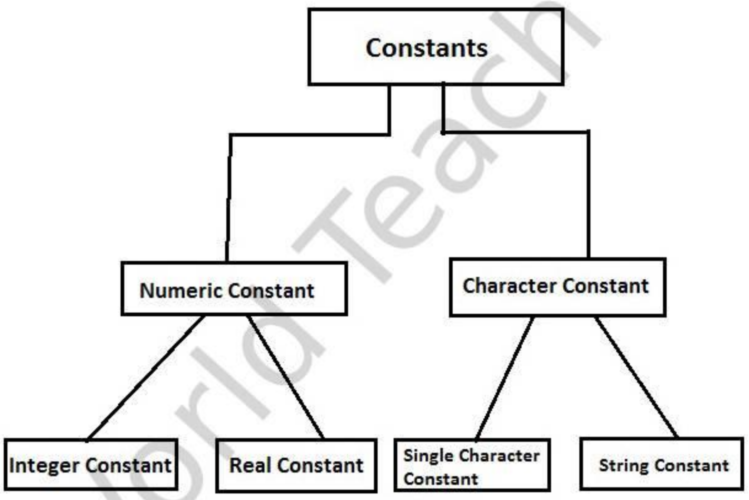


Fig. Basic types of c constants

Integer constant: Refers to a sequence of digits. Example: +78, -321

Real Constant: Numbers containing fractional part. Ex: 4.5,-0.76

A real constant may also be expressed in exponential notation. It will use ‘E’ or ‘e’ . It indicates that the number of ‘e’ will be raised to power 10.

Example: 3.18e3, -1.2E-1

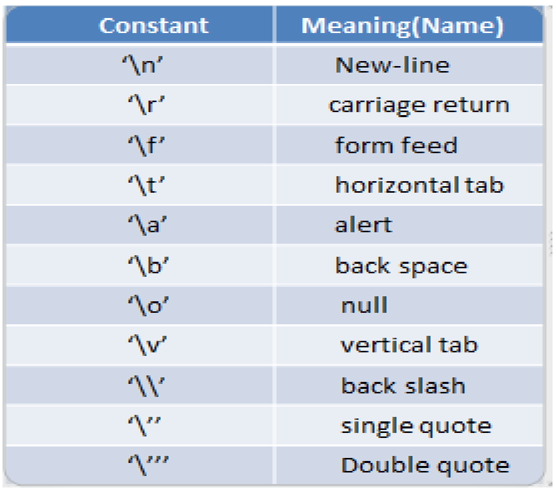
Single Character constant:

It contains a single character enclosed in a pair of single quote marks. Example: ‘a’,’#’,’5’

String Constant: is a sequence of characters enclosed in double quotes.

Example: “JSS”, SJCE1”, “#74”

Backslash characters:



Constant	Meaning(Name)
‘\n’	New-line
‘\r’	carriage return
‘\f’	form feed
‘\t’	horizontal tab
‘\a’	alert
‘\b’	back space
‘\o’	null
‘\v’	vertical tab
‘\\’	back slash
‘\’’	single quote
‘\”’	Double quote

Variables: A Variable is a data name that may be used to store a data value.

A variable can take different values at different times during execution.

(Rules to create a variable name: refer rules for identifier)

Data types:

ANSI C supports 3 classes of data types:

1. Primary data type: int, float, char, double and void
2. Derived data type: Arrays, Pointers
3. User Defined data type: typedef

```
Ex: typedef int xyz;
       xyz  a,b,c;
```

Primary data type

1. Int –	2 bytes	Range -32768 to +32768
2. Float -	4 bytes	Range- 3.4e-38 to 3.4e+38
3. Char	1 byte	Range- -128 to +127
4. Double	8 bytes	Range – 1.7e-308 to 1.7e+308
5. Void	0 byte	

Assigning values to variables:

Value can be assigned to variable at the time of declaration.

Example: int x=10;

Define Symbolic Constant:

Certain unique constants can appear at multiple locations in a program. Then it can be defined as symbolic constant.

Example:

```
#define PI 3.14
```

Format Specifiers:

Formatted Input Statement:

Data Type		Format
Integer	Integer	%d
	Short	%d
	Short unsigned	%u
	Long	%ld
	Long assigned	%lu
	Hexadecimal	%x
	Long hexadecimal	%lx
	Octal	%O (letter 0)
	long octal	%lo
Real	float,double	%f, %lf, %g
Character		%c
String		%s

Formatted Input statement:

Use format specifier and accept the data.

Example:

```
scanf("%d",&a);
scanf("%d%d",&a,&b);
scanf("%d%f",&x,&y);
```

Formatted output statement:

Use format specifier to display the result through standard output device monitor.

Example:

```
pritnf("%d",x)
pritnf("Sum = %d\n",s);
printf(" Square of %d = %d\n",n,s);
```

--------------------------End of unit 1 ---