

# I

Name \_\_\_\_\_

Roll No. \_\_\_\_\_ Year 20 \_\_\_\_\_ 20 \_\_\_\_\_

Exam Seat No. \_\_\_\_\_

ELECTRONICS GROUP | SEMESTER - II | DIPLOMA IN ENGINEERING AND TECHNOLOGY

## A LABORATORY MANUAL FOR 'C' PROGRAMMING LANGUAGE (22218)

### Pointers



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

int main()
{
    int a,b;
    register int c;
    clrscr();
    printf("Enter first number\n");
    scanf("%d",&a);
    printf("Enter second number\n");
    scanf("%d",&b);
    c=a+b;
    printf("The sum of %d and %d is %d",a,b,c);
    getch();
    return 0;
}
```

Enter first number  
10  
Enter second number  
20  
The sum of 10 and 20 is 30



**MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI**  
(Autonomous) (ISO 9001 : 2015) (ISO / IEC 27001 : 2013)

## **VISION**

To ensure that the Diploma level Technical Education constantly matches the latest requirements of technology and industry and includes the all-round personal development of students including social concerns and to become globally competitive, technology led organization.

## **MISSION**

To provide high quality technical and managerial manpower, information and consultancy services to the industry and community to enable the industry and community to face the changing technological and environmental challenges.

## **QUALITY POLICY**

We, at MSBTE are committed to offer the best in class academic services to the students and institutes to enhance the delight of industry and society. This will be achieved through continual improvement in management practices adopted in the process of curriculum design, development, implementation, evaluation and monitoring system along with adequate faculty development programmes.

## **CORE VALUES**

MSBTE believes in the followings:

- Education industry produces live products.
- Market requirements do not wait for curriculum changes.
- Question paper is the reflector of academic standards of educational organization.
- Well designed curriculum needs effective implementation too.
- Competency based curriculum is the backbone of need based program.
- Technical skills do need support of life skills.
- Best teachers are the national assets.
- Effective teaching learning process is impossible without learning resources.



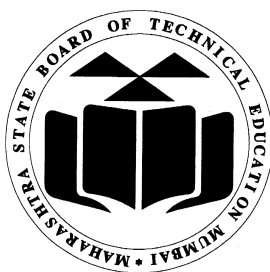
**A Laboratory Manual for**

# **'C' Programming Language**

**(22218)**

## **Semester-II**

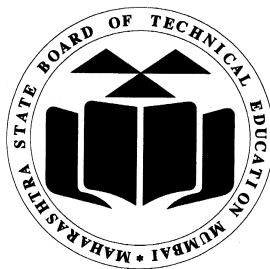
**(EJ/EX/EN/ET/EQ/DE)**



**Maharashtra State**  
**Board of Technical Education, Mumbai**  
**(Autonomous) (ISO 9001:2015) (ISO/IEC 27001:2013)**



Maharashtra State Board of Technical Education,  
(Autonomous) (ISO 9001 : 2015 ) (ISO/IEC 27001 : 2013)  
4th Floor, Government Polytechnic Building, 49, Kherwadi,  
Bandra ( East ), Mumbai - 400051.  
(Printed on December, 2017)



# MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION

## Certificate

This is to certify that Mr. / Ms. .... Roll  
No. ...., of First Semester of Diploma in.....  
..... of Institute,.....  
..... (Code: .....) has completed the term work satisfactorily  
in Subject **‘C’ programming Language (22218)** for the academic year  
20..... to 20..... as prescribed in the curriculum.

Place: .....

Enrollment No:.....

Date: .....

Exam. Seat No: .....

**Subject Teacher**

**Head of the Department**

**Principal**





## Preface

The primary focus of any engineering laboratory/field work in the technical education system is to develop the much needed industry relevant competencies and skills. With this in view, MSBTE embarked on this innovative 'I' Scheme curricula for engineering diploma programmes with outcome-based education as the focus and accordingly, relatively large amount of time is allotted for the practical work. This displays the great importance of laboratory work making each teacher; instructor and student to realize that every minute of the laboratory time need to be effectively utilized to develop these outcomes, rather than doing other mundane activities. Therefore, for the successful implementation of this outcome-based curriculum, every practical has been designed to serve as a '**vehicle**' to develop this industry identified competency in every student. The practical skills are difficult to develop through 'chalk and duster' activity in the classroom situation. Accordingly, the 'I' scheme laboratory manual development team designed the practicals to **focus** on the **outcomes**, rather than the traditional age old practice of conducting practicals to 'verify the theory' (which may become a byproduct along the way).

This laboratory manual is designed to help all stakeholders, especially the students, teachers and instructors to develop in the student the pre-determined outcomes. It is expected from each student that at least a day in advance, they have to thoroughly read through the concerned practical procedure that they will do the next day and understand the minimum theoretical background associated with the practical. Every practical in this manual begins by identifying the competency, industry relevant skills, course outcomes and practical outcomes which serve as a key focal point for doing the practical. The students will then become aware about the skills they will achieve through procedure shown there and necessary precautions to be taken, which will help them to apply in solving real-world problems in their professional life.

This manual also provides guidelines to teachers and instructors to effectively facilitate student-centered lab activities through each practical exercise by arranging and managing necessary resources in order that the students follow the procedures and precautions systematically ensuring the achievement of outcomes in the students.

Automation and embedded system design industry needs to build microcontroller based applications which are being developed using 'C'. In order to develop given application they have to build logic, develop algorithm and flow charts. This course is designed keeping in view of developing these skills and enhance programming knowledge of diploma students. It also helps to write codes for low level programming such as developing operating systems, drivers, and compilers; 'C' has been widely used as a general-purpose language to develop basic applications to solve engineering problems.

Although best possible care has been taken to check for errors (if any) in this laboratory manual, perfection may elude us as this is the first edition of this manual. Any errors and suggestions for improvement are solicited and highly welcome.

## **Programme Outcomes (POs) to be achieved through Practical**

Following programme outcomes are expected to be achieved out of the programme outcomes through the practicals of the course on:

- PO1. Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Electronics related problems.
- PO2. Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
- PO3. Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
- PO4. Engineering tools:** Apply relevant Computer programming / electronic technologies and tools with an understanding of the limitations.
- PO5. The engineer and society:** Assess social, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics.
- PO6. Environment and sustainability:** Apply Digital electronics engineering solutions also for sustainable development practices in social and environmental contexts using programming language.
- PO7. Ethics:** Apply ethical principles for commitment to professional ethics, responsibilities and norms of the practice also in the field of Electronics.
- PO8. Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.
- PO9. Communication:** Communicate effectively in oral and written form.
- PO10. Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

## Practical- Course Outcome matrix

<b>Course Outcomes (COs):</b> <ol style="list-style-type: none"> <li>Interpret the basic code of 'C'.</li> <li>Implement decision making in 'C' programming.</li> <li>Use Arrays and string in 'C' programming.</li> <li>Use functions in 'C' programs for modular programming approach.</li> <li>Use pointers to increase efficiency of programs.</li> <li>Implement basic concept of structure in 'C'.</li> </ol>							
S. No.	Title of the Practical	CO a.	CO b.	CO c.	CO d.	CO e.	CO f.
1	<b>Programs to display formatted output:</b> <ol style="list-style-type: none"> <li>Write a C program to display hexadecimal, decimal, octal format of entered number using %d, %c, %i, %f, %g, %u, %o, %s, %x</li> <li>Write algorithm and draw flow chart for following problems:                             <ol style="list-style-type: none"> <li>Addition of two numbers</li> <li>Exchange value of two variable</li> </ol> </li> </ol>	√	-	-	-	-	-
2	<b>Program using Logical and Bitwise operators:</b> <ol style="list-style-type: none"> <li>Display the message "Hello World", with your email_id using printf ( ) function.</li> <li>Logical operations: &amp; (AND) ,   (OR) for given values , Bitwise operations :&lt;&lt; (LEFT SHIFT), &gt;&gt; (RIGHT OPERATOR) for given values.</li> </ol>	√	-	-	-	-	-
3	<b>Evaluate scientific expressions using functions of header files in C:</b> <ol style="list-style-type: none"> <li>Write a program to display current time and date using <b>time.h</b> header file.</li> <li>Write a program to display addition of value of resistor R, Where,                             <ol style="list-style-type: none"> <li>R series =R1 + R2+R3 and</li> <li>R parallel =1/ R1 + 1/ R2+1/R3</li> </ol> </li> </ol>	√	-	-	-	-	-



S. No.	Title of the Practical	CO a.	CO b.	CO c.	CO d.	CO e.	CO f.
4	<b>Evaluate scientific problems using C.:</b> (a) Write a program to calculate inductive resistance ( $F_L$ ) with the help of given formula $F_L = 2 * \pi * f * L$ . Where $\pi$ , $f$ , $L$ are given data. (b) Write a program to calculate capacitive resistance ( $F_C$ ) with the help of given formula $F_C = 1/(2 * \pi * f * C)$ . Where $\pi$ , $f$ , $C$ are given data. <i>Note. Develop above programs using local variables, global variables and arithmetic operators.</i>	√	-	-	-	-	-
5	<b>Implement decision control statements in C using 'if' :</b> (a) Write a program to find whether given number is even or odd. (b) Write a program to find whether given number is Positive, negative or zero.	-	√	-	-	-	-
6	<b>Implement decision control statements in C using 'if-else' Statement:</b> a) Write a program to find the largest among n numbers using 'if- else'. b) Write a program to determine leap year using 'if-else'.	-	√	-	-	-	-
7	<b>Implement decision control statements in 'C' using 'nested if-else' :</b> a) Determine whether a string is palindrome. b) Find the greatest of the three numbers using conditional operators.	-	√	-	-	-	-
8	<b>Write implement switch case control statements in c.</b> Write a program to perform addition, subtraction; multiplication and division according to user's choice using switch case statement for given data.	-	√	-	-	-	-

S. No.	Title of the Practical	CO a.	CO b.	CO c.	CO d.	CO e.	CO f.
9	<b>Implement loop control statements in 'C' using 'for' loop:</b> a) Write a program to print the table for given no. in one column. b) Write a program to count the number of digit in a given number.	-	√	-	-	-	-
10	<b>Implement loop control statements in 'C':</b> (a) Find Fibonacci series for given number. (b) Write a program to produce the following output: <div style="text-align: center;">             1              2      3              4      5      6              7      8      9      10           </div>	-	√	-	-	-	-
11	<b>Print the Result sheet:</b> Conditions given are: marks $\geq 40\%$ pass, marks $< 40\%$ fail, marks $60 \geq$ first class, marks above 75 % distinction, marks $> 100$ and marks $< 0$ not valid.	-	√	-	-	-	-
12	<b>Processing elements of Array:</b> 1. Write a program to declare, modify and print elements of a given data array. 2. Write a program to find highest marks in a class of n students using array. 3.	-	-	√	-	-	-
13	<b>Write programs using array:</b> a) Write a program to copy of one array into second array for given data elements. b) Write a program to create an array by reversing the elements of the given array.	-	-	√	-	-	-

S. No.	Title of the Practical	CO a.	CO b.	CO c.	CO d.	CO e.	CO f.
14	<b>Write programs using multidimensional array:</b> (a) Write a program to sort numbers in ascending and descending in a given array. (b) Write a program to add two matrices of size 3*3 store additions in third matrix for given data elements.	-	-	√	-	-	-
15	<b>Perform operations on String using function:</b> (a) Write a program that accept a string from user and print that string. (b) Write a program that accept a string and compare it with existing string.	-	-	√	√	-	-
16	<b>Implement String function in C:</b> a) Write a program to accept and concatenate two strings. b) Write a program to find length of a string.	-	-	√	√	-	-
17	<b>Write c program to perform library function:</b> Develop Program to demonstrate: a) Use of all string handling functions. b) Use of few Mathematical functions	-	-	√	√	-	-
18	<b>Write C program to perform user defined function:</b> (a) Write a program to add two numbers using function. (b) Write a program to perform addition, subtraction, multiplication and division using switch case statement and user defined function for given data.	-	-	-	√	-	-

S. No.	Title of the Practical	CO a.	CO b.	CO c.	CO d.	CO e.	CO f.
19	<b>Write basic C programs using Pointer:</b> a) Write a program to use address operator (&) and pointer operator (*) for given data. b) Write a program to add two integer numbers using pointer	-	-	-	-	√	-
20	<b>Perform operation on array using pointer:</b> (a) Write a program to calculate the sum of elements of given array using pointer. (b) Write a program to access the array elements using pointer.	-	-	√	-	√	-
21	<b>Write c program to implement concept of call by value and call by reference in function:</b> a) Write a program to interchange given values of two variables using call by value mechanism. b) Write a program to interchange given values of two variables using call by reference mechanism.	-	-	-	√	√	-
22	<b>Implement concept of pointer using C:</b>  Write a C program to exchange given values of two variables using pointer.	-	-	-	-	√	-
23	<b>Implement structure in C:</b> Create structure DATE using 'C' having members' day, month, year and assign initial values to that structure.	-	-	-	-	-	√
24	<b>Implement operations carried on structure using C:</b> Write a program to create a structure for five students having data members like Roll No., Name, Class, and marks in three subjects and calculate the % of marks.	-	-	-	-	-	√

## **List of Industry Relevant Skills**

The following industry relevant skills of the competency **“Develop 'C' programs to solve broad-based electronic engineering related problems.”** are expected to be developed in you by undertaking the practicals of this laboratory manual.

1. Analyze problem definition
2. Develop algorithm and Draw Flow Chart.
3. Develop C programs for Real life applications.
4. Compile programs using Turbo C.
5. Debug and execute the program.
6. Ability to solve application level problems.

### **Brief Guidelines to Teachers**

1. Teacher should provide the guideline with demonstration of practical to the students with all features.
2. Teacher shall explain prior concepts to the students before starting of each experiment
3. Involve students in performance of each experiment.
4. Teacher should ensure that the respective skills and competencies are developed in the students after the completion of the practical exercise.
5. Teachers should give opportunity to students for hands on experience after the demonstration.
6. Teacher is expected to share the skills and competencies to be developed in the students.
7. Teacher may provide additional knowledge and skills to the students even though not covered in the manual but are expected the students by the industry.
8. Teachers are advised to prepare programming problems on similar guidelines as given below. Each student should develop and execute minimum TWO programs.
9. Finally give practical assignment and assess the performance of students based on task assigned to check whether it is as per the instructions.

### **Instructions for Students**

1. Listen carefully the lecture given by teacher about subject, curriculum, learning structure, skills to be developed.
2. Organize the work in the group and make record of all programs.
3. Students shall develop maintenance skill as expected by industries.
4. Student shall attempt to develop related hand-on skills and gain confidence.
5. Student shall develop the habits of evolving more ideas, innovations, skills etc. those included in scope of manual
6. Student shall refer technical magazines.
7. Student should develop habit to submit the practicals on date and time.
8. Student should well prepare while submitting write-up of exercise.
9. Attach /paste separate papers wherever necessary.



## Content Page

### List of Practicals and Progressive Assessment Sheet

S. No.	Title of the Practical	Page No.	Date of performance	Date of submission	Assessment marks (25)	Dated sign. of teacher	Remarks (if any)
1.	<b>Programs to display formatted output:</b> a) Write a C program to display hexadecimal, decimal, octal format of entered number using %d, %c, %i, %f, %g, %u, %o, %s, %x b) Write algorithm and draw flow chart for following problems: i. Addition of two numbers ii. Exchange value of two variable	1					
2.	<b>Program using Logical and Bitwise operators:</b> (a) Display the message "Hello World", with your email_id using printf ( ) function. (b) Logical operations: & (AND) ,   (OR) for given values , Bitwise operations :<< (LEFT SHIFT), >> (RIGHT OPERATOR) for given values.	10					
3.	<b>Evaluate scientific expressions using functions of header files in C:</b> (a) Write a program to display current time and date using <b>time.h</b> header file. (b) Write a program to display addition of value of resistor R, Where, i. R series = $R_1 + R_2 + R_3$ and ii. R parallel = $\frac{1}{R_1} + \frac{1}{R_2 + 1/R_3}$	17					

4.	<p><b>Evaluate scientific problems using C.:</b></p> <p>(a) Write a program to calculate inductive resistance (<math>F_L</math>) with the help of given formula <math>F_L = 2 * \pi * f * L</math>. Where <math>\pi</math>, <math>f</math>, <math>L</math> are given data.</p> <p>(b) Write a program to calculate capacitive resistance (<math>F_C</math>) with the help of given formula <math>F_C = 1/(2 * \pi * f * C)</math>. Where <math>\pi</math>, <math>f</math>, <math>C</math> are given data.</p> <p><i>Note. Develop above programs using local variables, global variables and arithmetic operators.</i></p>	24					
5.	<p><b>Implement decision control statements in C using 'if' :</b></p> <p>(a) Write a program to find whether given number is even or odd.</p> <p>(b) Write a program to find whether given number is Positive, negative or zero.</p>	32					
6.	<p><b>Implement decision control statements in C using 'if-else' Statement:</b></p> <p>a) Write a program to find the largest among n numbers using 'if- else'.</p> <p>b) Write a program to determine leap year using 'if-else'.</p>	40					
7.	<p><b>Implement decision control statements in 'C' using 'nested if-else' :</b></p> <p>a) Determine whether a string is palindrome.</p> <p>b) Find the greatest of the three numbers using conditional operators.</p>	48					
8.	<p><b>Write implement switch case control statements in c.</b></p> <p>Write a program to perform addition, subtraction; multiplication and division according to user's choice using switch case statement for given data.</p>	56					

9.	<b>Implement loop control statements in 'C' using 'for' loop:</b> a) Write a program to print the table for given no. in one column. b) Write a program to count the number of digit in a given number.	64					
10.	<b>Implement loop control statements in 'C':</b> (a) Find Fibonacci series for given number. (b) Write a program to produce the following output: <pre>               1             2   3           4   5   6         7   8   9  10           </pre>	72					
11.	<b>Print the Result sheet:</b> Conditions given are: marks $\geq 40\%$ pass, marks $< 40\%$ fail, marks $60 \geq$ first class, marks above 75 % distinction, marks $> 100$ and marks $< 0$ not valid.	80					
12.	<b>Processing elements of Array:</b> a) Write a program to declare, modify and print elements of a given data array. b) Write a program to find highest marks in a class of n students using array.	87					
13.	<b>Write programs using array:</b> a) Write a program to copy of one array into second array for given data elements. b) Write a program to create an array by reversing the elements of the given array.	95					
14.	<b>Write programs using multidimensional array:</b> (a) Write a program to sort numbers in ascending and descending in a given array. (b) Write a program to add two matrices of size $3 \times 3$ store additions in third matrix for given data elements.	102					

15.	<b>Perform operations on String using function:</b> (a) Write a program that accept a string from user and print that string. (b) Write a program that accept a string and compare it with existing string.	110					
16.	<b>Implement String function in C:</b> a) Write a program to accept and concatenate two strings. b) Write a program to find length of a string.	118					
17.	<b>Write c program to perform library function:</b> Develop Program to demonstrate: a) Use of all string handling functions. b) Use of few Mathematical functions	125					
18.	<b>Write C program to perform user defined function:</b> (a) Write a program to add two numbers using function. (b) Write a program to perform addition, subtraction, multiplication and division using switch case statement and user defined function for given data.	134					
19.	<b>Write basic C programs using Pointer:</b> a) Write a program to use address operator (&) and pointer operator (*) for given data. b) Write a program to add two integer numbers using pointer	142					
20.	<b>Perform operation on array using pointer:</b> (a) Write a program to calculate the sum of elements of given array using pointer. (b) Write a program to access the array elements using pointer.	149					

21.	<b>Write c program to implement concept of call by value and call by reference in function:</b> a) Write a program to interchange given values of two variables using call by value mechanism. b) Write a program to interchange given values of two variables using call by reference mechanism.	155					
22.	<b>Implement concept of pointer using C:</b> Write a C program to exchange given values of two variables using pointer.	163					
23.	<b>Implement structure in C:</b> Create structure DATE using 'C' having members' day, month, year and assign initial values to that structure.	170					
24.	<b>Implement operations carried on structure using C:</b> Write a program to create a structure for five students having data members like Roll No., Name, Class, and marks in three subjects and calculate the % of marks.	177					
<b>Total</b>							

\* To be transferred to proforma of CIAAN-2017.



## Practical No. 1: Programs to Display Formatted Output

### Problem Statement:

- a) Write a C program to display hexadecimal, decimal, octal format of entered number using %d, %c, %i, %f, %g, %u, %o, %s, %x
- b) Write algorithm and draw flow chart for following problems:
  - i. Addition of two numbers
  - ii. Exchange value of two variable

### I. Practical Significance

'C' Language provides very few basic data types. These data types using printf( ) function: it is standard for print formatting is used to display information required by the user and also prints values of variables having type specifiers for printf( ).

### II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Electronics related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
- **Engineering tools:** Apply relevant Computer programming / electrical technologies and tools with an understanding of the limitations.
- **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### IV. Relevant Course Outcome(s)

- Interpret the basic code of C.

### V. Practical Outcome (POs)

1. Write/ Compile / Execute simple 'C' program using Turbo 'C' compiler and basic data types.
2. Write C program for the evaluating arithmetic expression

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.




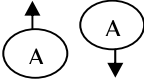
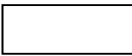

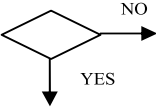
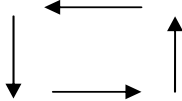

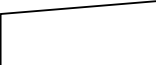
## VII. Minimum Theoretical Background

**Formatting Input/output statement:** C programming support two formatting functions `printf()` and `scanf()` out of which `printf()` convert data stored in the program into text stream for output to monitor. The minimum field width and precision/type specifiers are usually constants, however they may also provide by arguments to `printf()`.

Different type specifiers are also used with address operator (used with “%” sign) in `scanf()` statement.

Type specifier	Qualifying input
%c	For single characters
%d	For integer values
%f	For floating point numbers
%G or %g	For floating point numbers in shorter of exponential format
%i	for Integer
%o	for Octal numbers
%s	for sequence of (string of) characters
%u	for unsigned integer values
%x or %X	for hexadecimal values

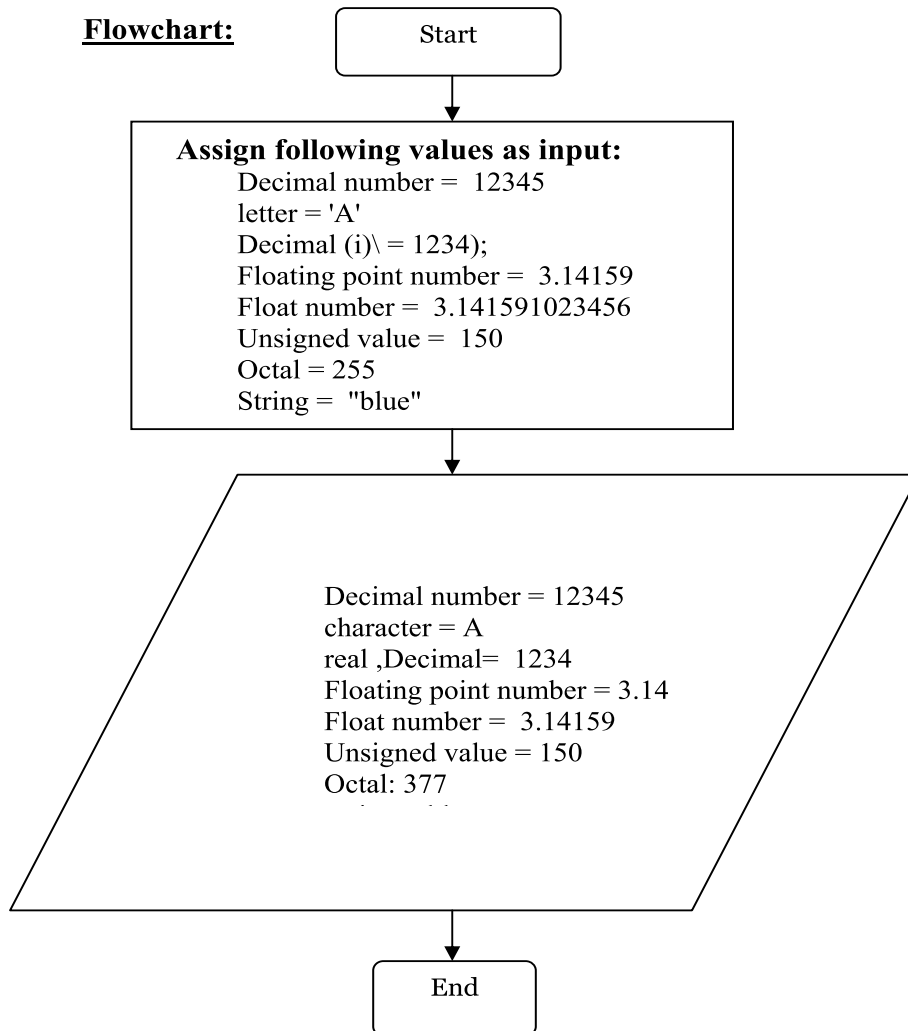
These are some standard graphics that are used in flowchart as following:

Symbol	Meaning	Symbol	Meaning
	Start/Stop		Connectors
	General Processing		Magnetic disk/Database
	Decision		Flow Lines
	Input / Output		Manual Input

### Algorithm:

1. List the data types user specify to print.
2. Select the format specifier for the respective data type.
3. Use proper syntax of `printf()` statement.
4. Check the required format of output.

**Flowchart:**



**C Program Code:**

```
#include<stdio.h>
#include<conio.h>
main()
{
    char letter;
    letter = 'A'; /* assign character value */
    clrscr();
    printf("Decimal number = %d\n", 12345);
    printf("character = %c\n", letter);
    printf("real ,Decimal= %i\n", 1234);
    printf("Floating point number = %3.2f\n", 3.14159);
    printf("Float number = %g\n", 3.141591023456);
    printf("Unsigned value = %u\n", 150);
    printf("Octal: %o\n", 255);
    printf("String = %s\n", "blue");
    printf("Hexadecimal: %x\n", 255);
    getch();
}
```

**Output of the source example:**

```
Decimal number = 12345  
character = A  
real ,Decimal= 1234  
Floating point number = 3.14  
Float number = 3.14159  
Unsigned value = 150  
Octal: 377  
String = blue  
Hexadecimal: ff
```

**VIII. Algorithm**

**IX. Flow Chart**

## **X. 'C' Program Code**

## **XI. Resources required**

<b>Sr. No.</b>	<b>Name of Resource</b>	<b>Specification</b>	<b>Quantity</b>	<b>Remarks</b>
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## **XII Precautions**

1. All C statements must end with a semicolon (;).
2. White space used in c to describe blanks and tabs.
3. Use proper format specifier for relevant data types in printf () statement.
4. Follow safety practices

## **XIII Resources used**

<b>S. No.</b>	<b>Name of Resource</b>	<b>Specification</b>
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	





[illegible]



### XVIII References / Suggestions for further Reading

1. [www.cprogrammingcodes.blogspot.in/2011/09/algorithms-and-flowchart.html](http://www.cprogrammingcodes.blogspot.in/2011/09/algorithms-and-flowchart.html) (as on 4/7/2017)
2. <https://www.programiz.com/article/flowchart-programming> (As on 4/7/2017)

### XIX Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### *List of Students /Team Members*

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No. 2: Program Using Logical and Bitwise Operators

### Problem Statement:

- (a) Display the message “Hello World”, with your email\_id using printf ( ) function.
- (b) Logical operations: & (AND) , | (OR) for given values ,  
Bitwise operations :<< (LEFT SHIFT), >> (RIGHT OPERATOR) for given values.

### I. Practical Significance

Operator is a symbol that specifies the Mathematical / Arithmetic, Logical or Relational operations to be performed using values of specified variables. It is a symbol that tells the computer to perform specific operation. C language supports different types of operators, which can be used with variables and constants to form expressions.

### II. Relevant Program Outcomes (POs)

1. **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Electronics related problems.
2. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
3. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
4. **Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.

### III. Competency and Practical skills

This practical develop the following skills

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### IV. Relevant Course Outcome(s)

- Interpret the basic code of C.

### V. Practical Outcome (POs)

Write a program to perform following operations:

1. Display the message “Hello World”, name, address, date of birth and email\_id using printf ( ) function.
2. Logical operations: & (AND) , | (OR) for given values,  
Bitwise operations :<< (LEFT SHIFT), >> (RIGHT OPERATOR) for given values.

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

## VII. Minimum Theoretical Background

C programming has wide range of operators to perform various operations. For better understanding of operators, these operators can be classified as:

Operator	Its types
Arithmetic operators	+ , - , * , / % (modulus)
Relational operators	< , > , == , <= , >= , !=
Logical operators:	&& ,    , !
Unary operators	- , ++ , --
Conditional operators/ ternary operator	( ? : ) For example: x = exp1 ? exp2 :exp3
Bitwise operators	&& (Bitwise AND),    (Bitwise OR), ~ (bitwise NOT), << , >>(Shift operator)

## VIII. Algorithm

**IX. Flow Chart:**

**X. 'C' Program Code**

## XI. Resources required

Sr. No.	Name of Resource	Specification	Qty.	Remarks
1	Hardware: computer system	Computer, (i3-i5 preferable), RAM minimum 2 GB and onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## XII Precautions

1. All C statements must end with a semicolon (;).
2. White space used in c to describe blanks and tabs.
3. Use proper format specifier for relevant data types in printf( ) statement.
4. Select proper operator with its syntax and precedence.

## XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## XIV Results (Output of the Program)

.....  
 .....  
 .....

## XV Conclusion(s)

.....  
 .....  
 .....

## XVI Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. State result of following:

Given a = 4, b=12

Expression	Specify your answer/output
a = b	b =
b = ++a	b =
a = b--	a =
a && b	
b << 4	b
x = (a>7) ? a : (a-2)	x =

2. Write algorithm and draw flow chart for following problems:

- iv. Addition of two numbers
- v. Exchange value of two variable

3. **Predict the output or error(s) for the following:**

```
void main()
{
    int i=-1,j=-1,k=0,l=2,m;
    m=i++&& j++&& k++||l++;
    printf("%d %d %d %d %d",i,j,k,l,m);
}
```

4. void main()

```
{
    int c=- -2;
    printf("c=%d",c);
}
```

5. void main()

```
{
    int i=10;
    i=!i>14;
    printf("i=%d",i);
}
```

**[Space for answers]**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

## XVII Exercise

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Bitwise operations :<< (LEFT SHIFT), >> (RIGHT OPERATOR) for given values.
2. Display the message “Hello World”, your name, address, date of birth and email\_id using printf ( ) function.

## XVIII References / Suggestions for further Reading

1. <https://www.programiz.com/c-programming/c-operators>
2. [https://en.wikipedia.org/wiki/Bitwise\\_operations\\_in\\_C](https://en.wikipedia.org/wiki/Bitwise_operations_in_C)

## XIX Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

### List of Students / Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	



### Practical No.3: Evaluate Scientific Expressions Using Functions of Header Files in C.

#### Problem Statement:

- (a) Write a program to display current time and date using **time.h** header file.
- (b) Write a program to display addition of value of resistor R, Where,
  - i.  $R_{\text{series}} = R_1 + R_2 + R_3$  and
  - ii.  $R_{\text{parallel}} = 1/R_1 + 1/R_2 + 1/R_3$

**Note.** (Please use \* symbol and it should be bold)

#### I. Practical Significance

C programming is capable of evaluating scientific expression for solving engineering problems. Flexibility and types of operators make it easier. Programmer can find time required to execute the program which will be helpful to determine and compare efficiency of same program using different approaches.

#### II. Relevant Program Outcomes (POs)

- 1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
- 2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
- 3. **Engineering tools:** Apply relevant Computer programming / electronic technologies and tools with an understanding of the limitations.
- 4. **The engineer and society:** Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics.

#### III. Competency and Practical skills

This practical is expected to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

- 1. Write algorithm and Draw Flow Chart.
- 2. Write and Save a simple C program for function using pointers.
- 3. Compile (Turbo C) the C program.
- 4. Debug and execute the program.
- 5. Follow ethical practices.

The basic process model used for these kinds of problems starts with identification of need, then goes through information gathering, idea generation, evaluation and selection steps.

#### IV. Relevant Course Outcomes(s)

Interpret the basic code of C.

#### V. Practical Outcome (POs)

- 1. Write a program to display current time and date using **time.h** header file.
- 2. Write a program to display addition of two numbers.
- 3. Write a program to display value of resistor R, Where,
  - i.  $R_{\text{series}} = R_1 + R_2 + R_3$  and
  - ii.  $R_{\text{parallel}} = 1/R_1 + 1/R_2 + 1/R_3$

**Note:-** Use math.h header file

## **VI. Relevant Affective domain related Outcome(s)**

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

## **VII. Minimum Theoretical Background**

Engineers use computers to solve a variety of problems ranging from the evaluation of a simple function to solving a system of nonlinear equations. C has become the language of choice for many engineers and scientists not only because it has powerful commands and data structures, but also because it can easily be used for system-level operations.

Math.h is a header file in the standard library of the C programming language designed for basic mathematical operations. Most of the functions involve the use of floating point numbers.

All functions that take or return an angle work in radians.

### **Example to count clock ticks:**

```
#include <time.h>
#include <stdio.h>
#include <dos.h>

int main(void)
{
    clock_t start, end;
    start = clock();

    delay(2000);

    end = clock();
    printf("The time was: %f\n", (end - start) / CLK_TCK);

    return 0;
}
```

## **VIII. Algorithm**

**IX. Flow Chart:**

**X. 'C' Program Code**

## XI. Resources required

Sr. No.	Name of Resource	Specification	Qty.	Remarks
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## XII. Precautions

1. Use appropriate data type to assign appropriate values of resistances.
2. White space used in c to describe blanks and tabs.
3. Use proper format specifier for relevant data types in printf( ) statement.
4. Use proper function from time .h (header file) with relevant parameters required in function.

## XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## XIV Result (Output of the Program)

.....  
 .....  
 .....

## XV Conclusion(s)

.....  
 .....  
 .....  
 .....

## XVI Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. Write any two in-built functions with syntax from **math.h** those can be used in above programs?
2. Write any three in-built functions with syntax from **time.h** those will be useful or helpful to analyze program?
3. Write expression using operators in C for finding Serial resonance?

- [Space for answers]**

[illegible]



### XVIII References / Suggestions for further Reading

1. Web address: [https://en.wikibooks.org/wiki/C\\_Programming/math.h](https://en.wikibooks.org/wiki/C_Programming/math.h) (As on 4/7/2017).
2. Book: C Application programs and projects, Dilip Mali, Pramod Vasambekar Penram International Publishing (India) Pvt. Ltd. Mumbai, ISBN(10): 81-87972-24-6

### XIX Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### *List of Students / Team Members*

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No.4: Evaluate Scientific Problems Using C

### Problem Statement:

- (a) Write a program to calculate inductive resistance ( $F_L$ ) with the help of given formula  $F_L = 2 * \pi * f L$ . Where  $\pi$ ,  $f$ ,  $L$  are given data.
- (b) Write a program to calculate capacitive resistance ( $F_C$ ) with the help of given formula  $F_C = 1/(2 * \pi * f C)$ . Where  $\pi$ ,  $f$ ,  $C$  are given data.

*Note. Develop above programs using local variables, global variables and arithmetic operators.*

### I. Practical Significance

Engineers use computers to solve a variety of problems ranging from the evaluation of a simple function to solving a system of nonlinear equations. C is accepted by many engineers and scientists not only because it has powerful commands and data structures, but also because it can easily be used for system-level operations, their flexibility and ease of operations make it easier.

### II. Relevant Program Outcomes (POs)

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **The engineer and society:** Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics.
4. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### IV. Relevant Course Outcomes

Interpret the basic code of C.

### V. Practical Outcome (POs)

The basic process model used for these kinds of problems starts with identification of need, and then goes through information gathering, idea generation, evaluation and selection steps.

1. Write a program to calculate inductive resistance ( $F_L$ ) with the help of given formula  $F_L = 2 * \pi * f L$ .  
Where  $\pi$ ,  $f$ ,  $L$  are given data.



2. Write a program to calculate capacitive resistance ( $F_C$ ) with the help of given formula  $F_C = 1/(2 * \pi * f * C)$ .

Where  $\pi$ ,  $f$ ,  $C$  are given data.

**Note.** Develop above programs using local variables, global variables and arithmetic operators.

#### VI. Relevant Affective domain related Outcome(s)

1. Solving open-ended problems
2. Follow safety measures
3. Follow ethical practices.

#### VII. Minimum Theoretical Background

##### Global variables:

They are declared outside function. Its scope is throughout the program. Its life span is also throughout entire program. They are automatically initialized to 0.

##### Local variables:

The declaration is placed after the operating curly brace { of any function including main( ), and before any function statement. Scope of a local variable is limited to the function in which it is declared. Life span of a local variable within the block in which it is declared.

##### Illustration of Global and Local variables used in C Program.

```
#include<stdio.h>

// Global variables
int A;
int B;

int Add()
{
    return A + B;
}

int main( )
{
    int answer; // Local variable
    A = 5;
    B = 7;
    answer = Add();
    printf("%d\n",answer);
    return 0;
}
```

## **VIII. Algorithm**

## **IX. Flowchart**

## **X. 'C' Program Code**

### XI. Resources required

Sr. No.	Name of Resource	Specification	Qty.	Remarks
1	Hardware: computer system,	Computer, (i3-i5 preferable), RAM minimum 2 GB and onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

### XII. Precautions

1. Do not change type of variable (Global/Local) during program debugging.
2. While declaring a variable, assign initial value.
3. Strictly follow the instruction for writing, compiling and executing the program.
4. Follow ethical practices.

### XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

### XIV Result (Output of the Program)

.....

.....

.....

### XV Conclusion(s)

.....

.....

.....

.....

### XVI Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write expression using operators in C for finding Serial resonance?

2. What expression is required for calculating voltage when current and resistance is given in OHMs law?

Algorithm	Flowchart
1. Consider following algorithm and draw flowchart 1. Start 2. Input 4 numbers 3. Calculate average of these numbers. 4. Display division 5. End	

## XVII Exercise

**Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:**

**(Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write a program to calculate capacitive resistance ( $F_C$ ) with the help of given formula  $F_C = 1/(2 * \pi * f * C)$ .  
 Where  $\pi$ ,  $f$ ,  $C$  are given data.  
**Note.** Develop above programs using local variables, global variables and arithmetic operators.
2. Write a program to convert a temperature of  $65^0$  C to the Fahrenheit scale using the relationship  $F = (9/5) * C + 32$
3. Write a program to display a given distance between two cities in kilometers into miles and meters.
4. Write a program to state whether the entered year is a leap year or not

**[Space for Answer]**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[illegible]

### XVIII References / Suggestions for further Reading

1. [https://en.wikibooks.org/wiki/C\\_Programming/math.h](https://en.wikibooks.org/wiki/C_Programming/math.h) (as on 4/7/2017)
2. <https://www.codingunit.com/c-tutorial-functions-and-global-local-variables> (as on 4/7/2017)
3. Book: C Application programs and projects, Dilip Mali, Pramod Vasambekar, Penram International Publishing (India) Pvt. Ltd. Mumbai, ISBN(10): 81-87972-24-6

### XIX Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### *List of Student Team Members*

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## **Practical No.5: Implement Decision Control Statements in C Using 'IF' Statement.**

### **Problem Statement:**

- (a) Write a program to find whether given number is even or odd.
- (b) Write a program to find whether given number is Positive, negative or zero.

### **I. Practical Significance**

**Control statements** enable us to specify the flow of program **control**; i.e. the order in which the instructions in a program must be executed. Generally Instruction are executed sequentially. They make it possible to make decisions, to perform tasks repeatedly or to jump from one section of code to another. Such cases it becomes necessary to alter the sequence of execution based on certain conditions. For this purpose decision control statement/structure is required.

### **II. Relevant Program Outcomes (POs)**

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based engineering problems in electronic relevant programmes.
3. **Engineering tools:** Apply relevant computer programming / electronic technologies and tools with an understanding of the limitations.
4. **Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.
5. **Communication:** Communicate effectively in oral and written form.

### **III. Competency and Practical skills**

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### **IV. Relevant Course Outcome(s)**

Implement decision making in 'C' programming.

### **V. Practical Outcome (POs)**

Implement decision control statements in C using 'if'

1. Write a program to find whether given number is even or odd.
2. Write a program to find whether given number is Positive, negative or zero.

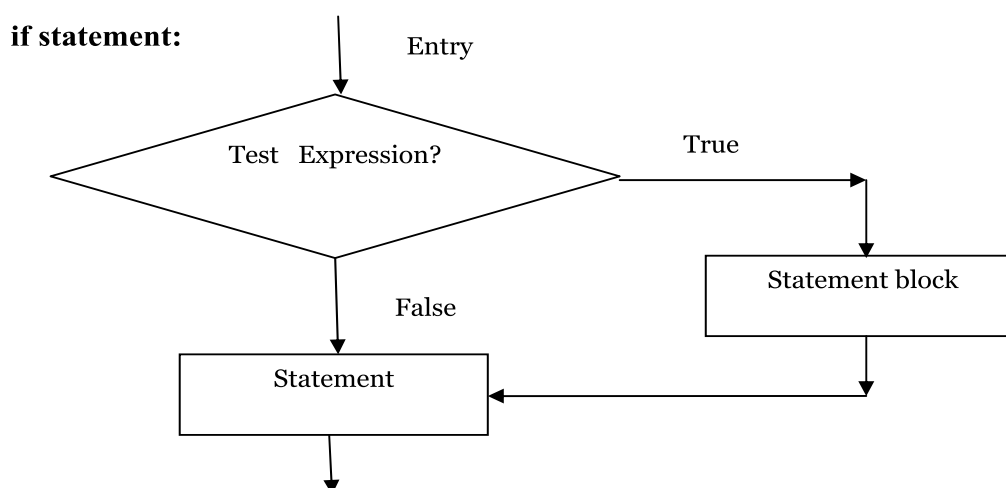
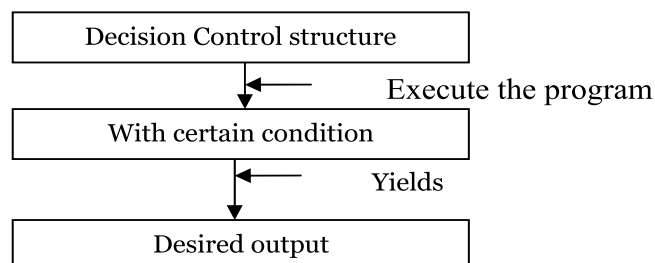
### **VI. Relevant Affective domain related Outcome(s)**

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.



## VII. Minimum Theoretical Background

**Control structure** is a block of programming that analyzes variables and chooses a direction in which to go based on given parameters. The term flow **control** details the direction the program takes (which way program **control** "flows").



If statement is two way decision statement and is used in conjunction with an expression. If the test expression is evaluated to true (nonzero); statements inside the body will be executed. If the test expression is evaluated to false (0), statements inside the body will be skipped from execution.

### Syntax (if-else):

```
if(test expression)
{
    True block statements;
}
```

## VIII. Algorithm

**IX. Flow Chart:**

**X. 'C' Program Code**

## XI. Resources required

Sr. No.	Name of Resource	Specification	Qty.	Remarks
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX onwards.		
3	Software:	Turbo C /C++ Version 3.0 or later		

## XII. Precautions

1. Start and termination of control statements at specific situations.
2. When nesting of control statement is required place the if and/or if-else statements are used at appropriate locations and situations.
3. Start and terminate the if ,if-else statements individually or in nested situations to achieve the necessary situation.
4. Conditions provided in if or if-else with proper format should satisfy the required condition(s).

Check the output as per problem definition

## XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## XIV Result (Output of the Program)

.....  
 .....  
 .....

## XV Conclusion(s)

.....  
 .....  
 .....  
 .....

## XVI Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. Differentiate break and continue.

2. State any two situations where branching is applicable?
3. What will be the output of any expression containing only relational operator?
  - a) 0 or 1
  - b) Always 1
  - c) Any nonzero number
4. To evaluate the value of the expression  $a > b$  the actual operation that takes place is:

Program Code	Output
<b>a)</b> <pre>#include&lt;stdio.h&gt; void main() { int a = 10; printf("%d ",++a); printf("%d",a++); getch(); }</pre>	
Program Code	Output
<b>b)</b> <pre>#include&lt;stdio.h&gt; void main() { int i = 5; if(i==5) printf("\n I am in First year "); else printf("\n I am in Second year "); getch(); }</pre>	
Program Code	Output
<b>c)</b> <pre>#include&lt;stdio.h&gt; void main() { int a = 10; int b = 5; if(a == 5) b++; printf("\n %d %d \n ",a,b++); getch(); }</pre>	

## XVII Exercise

(Note: Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write a program to find whether given number is Positive, negative or zero.
2. Write a program to check whether the student fail, pass, second-class , first-class, first-class with distinction.

**[Space for answers]**

[illegible]

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

### XVIII References / Suggestions for further Reading

1. [www.cprogrammingcodes.blogspot.in/2011/09/algorithms-and-flowchart.html](http://www.cprogrammingcodes.blogspot.in/2011/09/algorithms-and-flowchart.html) (as on 4/7/2017)
2. <https://www.programiz.com/article/flowchart-programming> (As on 4/7/2017)

### XIX Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### *List of Students / Team Members*

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## **Practical No. 6: Implement Decision Control Statements in C Using 'if-else' Statement.**

### **Problem statement:**

- a) Write a program to find the largest among n numbers using 'if- else'.
- b) Write a program to determine leap year using 'if-else'.

### **I. Practical Significance**

Decision Control statements execute a set of instructions depending on certain conditions. These help us to specify the flow of program control; i.e. the order in which the instructions in a program must be executed. Generally Instruction are executed sequentially. They make it possible to make decisions, to perform tasks repeatedly or to jump from one section of code to another. There will be true condition and false condition to in control statement syntax. Such cases it becomes necessary to alter the sequence of execution based on certain conditions. For this purpose decision control statement/structure is required.

### **II. Relevant Program Outcomes (POs)**

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based engineering problems in electronic relevant programmes.
3. **Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.
4. **Communication:** Communicate effectively in oral and written form.

### **III. Competency and Practical skills**

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### **IV. Relevant Course Outcome(s)**

Implement decision making in 'C' programming.

### **V. Practical Outcome (POs)**

1. Write a program to find the largest among n numbers using 'if- else'.
2. Write a program to determine leap year using 'if-else'.

### **VI. Relevant Affective domain related Outcome(s)**

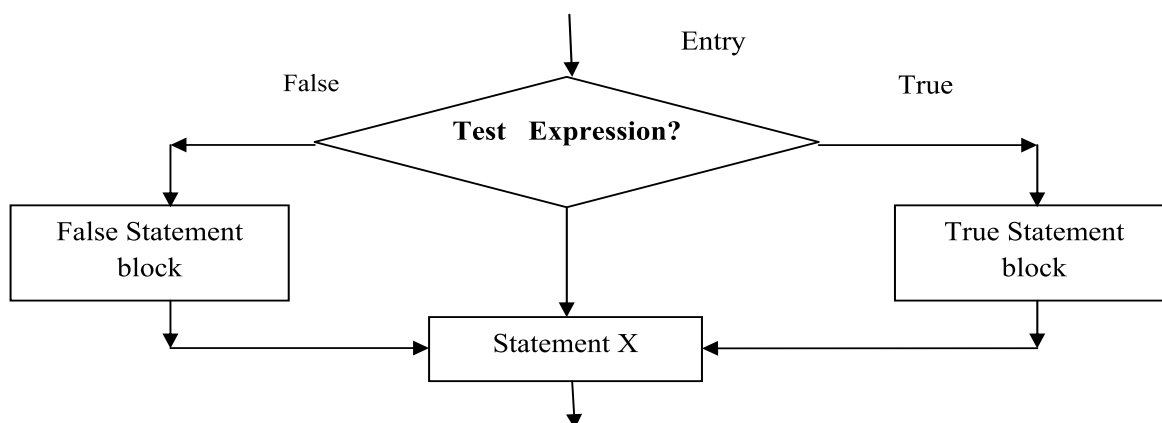
1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.



## VII. Minimum Theoretical Background

**Control structure** is a block of programming that analyzes variables and chooses a direction in which to go based on given parameters. The term flow **control** details the direction the program takes (which way program **control** "flows").

**If-else statement:**



**Syntax (if-else):**

```
if(test expression)
{
    True block statements;
}
else
{
    False block statements;
}
statement x;
```

If statement is two way decision statement and is used in conjunction with an expression. If the test expression is evaluated to true (nonzero), statements inside the 'if block' will get execute. If test expression is evaluated to false (0), statements inside the 'else block' will gets execute.

## VIII. Algorithm

**IX. Flow Chart:**

**X. 'C' Program Code**

## XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## XII Precautions

1. Start and termination of control statements at specific situations.
2. When nesting of control statement is required place the if and/or if-else statements are used at appropriate locations and situations.
3. Conditions provided in if or if-else with proper format should satisfy the required condition(s).
4. Start and terminate the if ,if-else statements individually or in nested situations to achieve the necessary situation.

## XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## XIV Result (Output of the Program)

.....  
 .....  
 .....

## XV Conclusion(s)

.....  
 .....  
 .....  
 .....

## XVI Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. Write an algorithm to input any number between 1-7 and print equivalent day name. For example: 1-Monday, 2-Tuesday etc.
2. Write an algorithm to input three numbers and find maximum between all?
3. What will be the detect error (if any) / write output of the following program segment if input is 16?  
 int i;

```
int a=10,b=5;
if(a>3 && b!=3)
printf("condition satisfies");
else
printf("condition does not sa
```

[illegible]

[illegible]This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.



### XVIII References / Suggestions for further Reading

1. <http://www.cprogramming.com/tutorial/c/lesson2.html> (as on 4/7/2017)
2. [https://www.tutorialspoint.com/cprogramming/c\\_decision\\_making.htm](https://www.tutorialspoint.com/cprogramming/c_decision_making.htm) (As on 4/7/2017)
3. <http://www.codeforwin.in/2015/05/if-else-programming-practice.html> (as on 4/7/2017)

### XIX Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### List of Students / Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## **Practical No.7: Implement Decision Control Statements in C Using 'Nested IF-ELSE' Statement.**

### **Problem Statement:**

- c) Determine whether a string is palindrome.
- d) Find the greatest of the three numbers using conditional operators.

### **I. Practical Significance**

**Control statements** enable us to specify the flow of program **control**, Many times programmer need to make a series of decisions. In these cases need to use more than one if else statements in the nested form.

### **II. Relevant Program Outcomes (POs)**

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based engineering problems in electronic relevant programmes.
3. **Engineering tools:** Apply relevant computer programming / electronic technologies and tools with an understanding of the limitations.
4. **Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.
5. **Communication:** Communicate effectively in oral and written form.

### **III. Competency and Practical skills**

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### **IV. Relevant Course Outcomes(s)**

- Implement decision making in 'C' programming.

### **V. Practical Outcome (POs)**

1. Determine whether a string is palindrome.
2. Find the greatest of the three numbers using conditional operators.

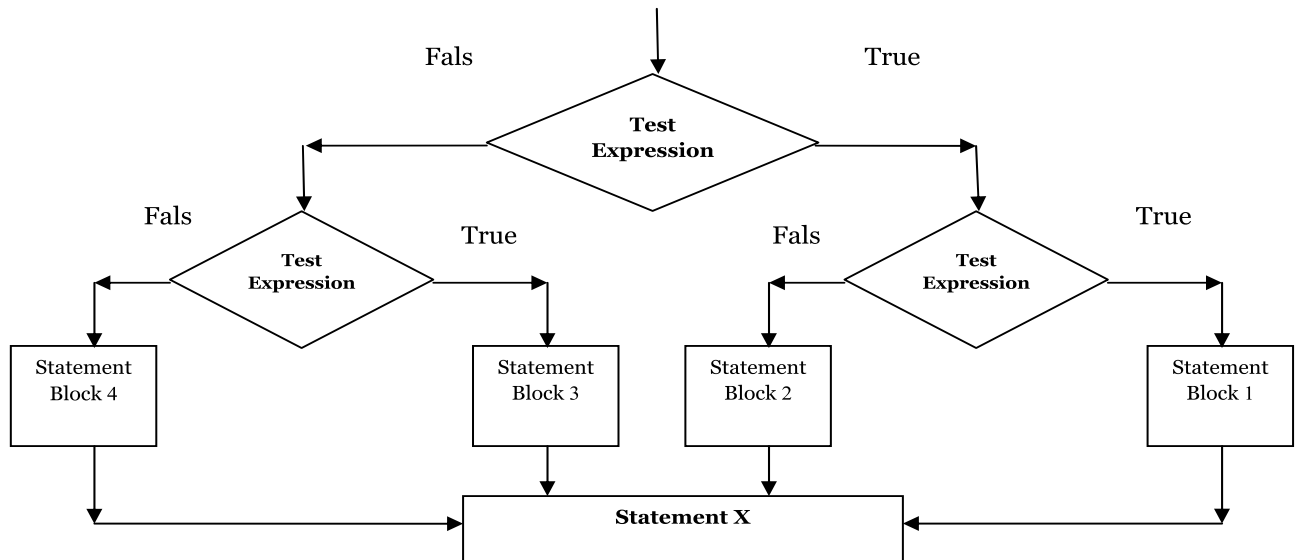
### **VI. Relevant Affective domain related Outcome(s)**

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices



## VII. Minimum Theoretical Background

**Control structure** is a block of programming that analyzes variables and chooses a direction in which to go based on given parameters. The term flow **control** details the direction the program takes (which way program **control** "flows").



### Syntax (nested if-else statement):

```

if (test condition 1)
{
    if(condition 2)
    {
        statement block 1 ;
    }
    else
    {
        statement block 2;
    }
}
else
{
    if(condition 3)
    {
        statement block 3 ;
    }
    else
    {
        statement block 4;
    }
}
statement x;
    
```

If condition 1 is false, then condition 2 is checked and statements are executed if it is true. If condition 2 also gets failure, then else part is executed.

Nested if is the combined version of all if and is the most complex format of all if. In nested if more than one condition are checked one after another to executed a specific statement.

## **VIII. Algorithm**

## **IX. Flow Chart**

## **X. Program Coding:**

## **XI. Resources required**

<b>Sr. No.</b>	<b>Name of Resource</b>	<b>Specification</b>	<b>Qty.</b>	<b>Remarks</b>
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX onwards.		
3	Software:	Turbo C /C++ Version 3.0 or later		

## **XII Precautions**

1. Start and termination of control statements at specific situations.
2. When nesting of control statement is required place if and/or if-else statements are used at appropriate locations and situations.
3. Conditions provided in nested if-else with proper format should satisfy the required condition(s).
4. Start and terminate the nested if-else statements individually or in nested situations to achieve the necessary situation.
5. Check the output as per problem definition.

### XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

### XIV Result (Output of the Program)

.....  
.....  
.....  
.....  
.....

### XV Conclusion(s)

.....  
.....  
.....  
.....

### XVI Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

5. What will be output when you will execute following c code?

```
#include<stdio.h>
void main(){
    int a=100;
    if(a>10)
        printf("M.S. Dhoni");
    else if(a>20)
        printf("M.E.K Hussey");
    else if(a>30)
        printf("A.B. d villiers");
}
```

2. Choose all that apply:

- (A) M.S. Dhoni
- (B) A.B. de villiers
- (C) M.S Dhoni  
M.E.K Hussey  
A.B. de Villiers
- (D) Compilation error: More than one condition is true
- (E) None of the above

**[Space for answers]**

[illegible]



.....

.....

.....

.....

### XVIII References / Suggestions for further Reading

1. <http://www.cprogramming.com/tutorial/c/lesson2.html>
2. [https://www.tutorialspoint.com/cprogramming/c\\_decision\\_making.html](https://www.tutorialspoint.com/cprogramming/c_decision_making.html)
3. <http://www.codeforwin.in/2015/05/if-else-programming-practice.html> (as on 4/7/2017)

### XIX Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### *List of Students / Team Members*

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## **Practical No.8: Write Implement Switch Case Control Statements In C.**

### **Problem Statement:**

Write a program to perform addition, subtraction; multiplication and division according to user's choice using switch case statement for given data.

### **I. Practical Significance**

Switch case statement is control structure where programmer need to select many alternatives, the program using if statement becomes complex. The reliability of program decreases. C has built in multi-way decision statement known as switch. Switch statement tests the value of given variable (or expression against a list of case values and when match is found, block of statements associated with that case is executed.

### **II. Relevant Program Outcomes**

1. **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Electronics related problems.
2. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
3. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
4. **Communication:** Communicate effectively in oral and written form.

### **III. Competency and Practical skills**

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### **IV. Relevant Course Outcome (s)**

Implement decision making in 'C' programming.

### **V. Practical Outcome (POs)**

Write a program to perform addition, subtraction; multiplication and division according to user's choice using switch case statement for given data.

### **VI. Relevant Affective domain related Outcome(s)**

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.



## VII. Minimum Theoretical Background

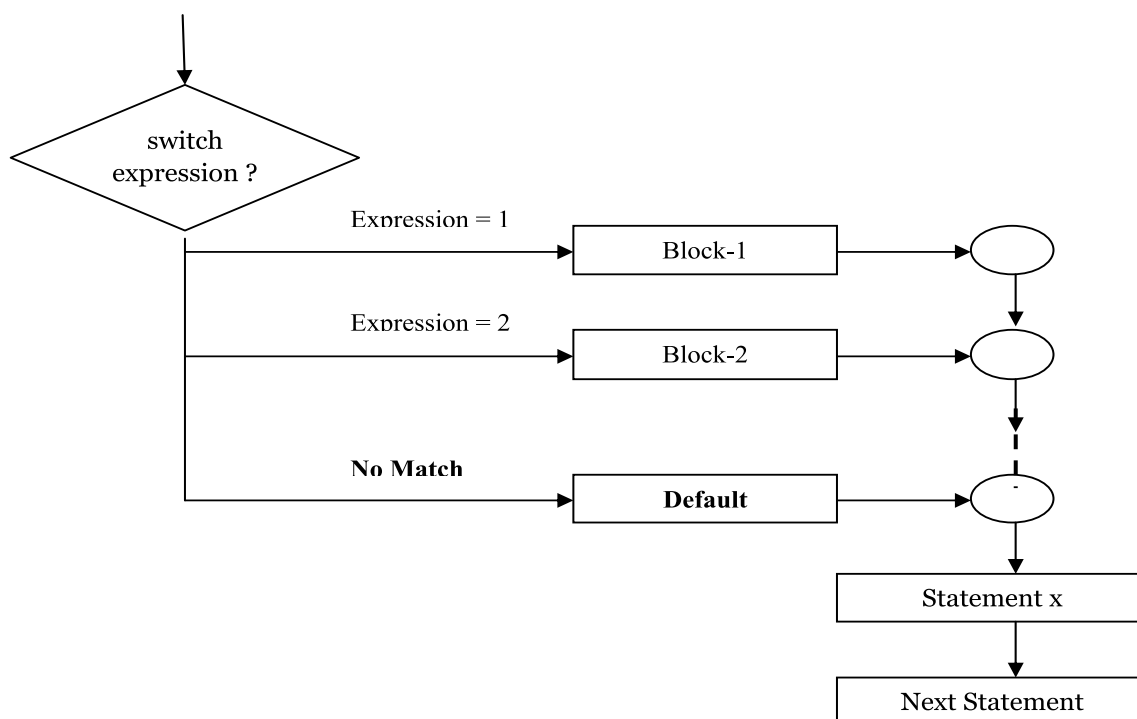
Switch statement has particular group statements to be chosen from several available groups. The selection is based upon the current value of an expression which is included within the switch statement. This determinates switch block when control reaches to switch, it checks value and matches that with the value of case(s) (when not matches default its optional) executed break statement.

### Syntax Switch-case:

```
switch (expression)
{
    case 1:
        block 1
        break;
    case 2:
        block 2
        break;
    .....
    -----

    case n:
        block n;
        break;
    default:
        default block;
        break;
}
statement x;
```

### Flow control of switch statement:



## **VIII. Algorithm**

## **IX. Flow Chart**

## **X. 'C' Program Code**

## **XI. Resources required**

<b>Sr. No.</b>	<b>Name of Resource</b>	<b>Specification</b>	<b>Qty.</b>	<b>Remarks</b>
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB and onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX onwards.		
3	Software:	Turbo C /C++ Version 3.0 or later		

## **XII Precautions**

1. Start and termination of control statements at specific situations.
2. When nesting of control statement is required place if and/or if-else statements are used at appropriate locations and situations.
3. Conditions provided in nested if-else with proper format should satisfy the required condition(s).
4. Start and terminate the nested if-else statements individually or in nested situations to achieve the necessary situation.
5. Write program code
6. Check the output as per problem definition.

### XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

### XIV Result

.....  
.....  
.....  
.....  
.....

### XV Conclusion(s)

.....  
.....  
.....  
.....

### XVI Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. What will be output when you will execute following c code?

```
#include<stdio.h>
void main()
{
    int a=100;
    if(a>10)
        printf("M.S. Dhoni");
    else if(a>20)
        printf("M.E.K Hussey");
    else if(a>30)
        printf("A.B. de villiers");
}
```

2. Choose all that apply:

- (A) M.S. Dhoni
- (B) A.B. de villiers
- (C) M.S Dhoni  
M.E.K Hussey  
A.B. de Villiers
- (D) Compilation error: More than one condition is true
- (E) None of the above

**[Space for answers]**

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

**(Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

**[Space for answers]**

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

### XVIII References / Suggestions for further Reading

1. <http://www.cprogramming.com/tutorial/c/lesson2.html> (as on 4/7/2017)
2. [https://www.tutorialspoint.com/cprogramming/c\\_decision\\_making.htm](https://www.tutorialspoint.com/cprogramming/c_decision_making.htm) (As on 4/7/2017)
3. <http://www.codeforwin.in/2015/05/if-else-programming-practice.html> (as on 4/7/2017)

### XIX Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### *List of Student Team Members*

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## **Practical No. 9: Implement Loop Control Statements in 'C' Using 'for' Loop**

### **Problem Statement:**

- a) Write a program to display the series of add number from 1 to 100.
- b) Write a program to count the number of digit in a given number.

### **I. Practical Significance**

This practical is useful for students to understand the use of loop statements. A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

### **II. Relevant Program Outcome (POs)**

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **Communication:** Communicate effectively in oral and written form.
4. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

### **III. Competency and Practical skills**

This practical will develop the following skills:

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### **IV. Relevant Course Outcomes(s)**

Implement decision making in 'C' programming.

### **V. Practical Outcome (POs)**

Implement loop control statements in 'C' using 'for' loop

1. Write a program to display the series of add number from 1 to 100.
2. Write a program to count the number of digit in a given number.

### **VI. Relevant Affective domain related Outcome(s)**

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

### **VII. Minimum Theoretical Background**

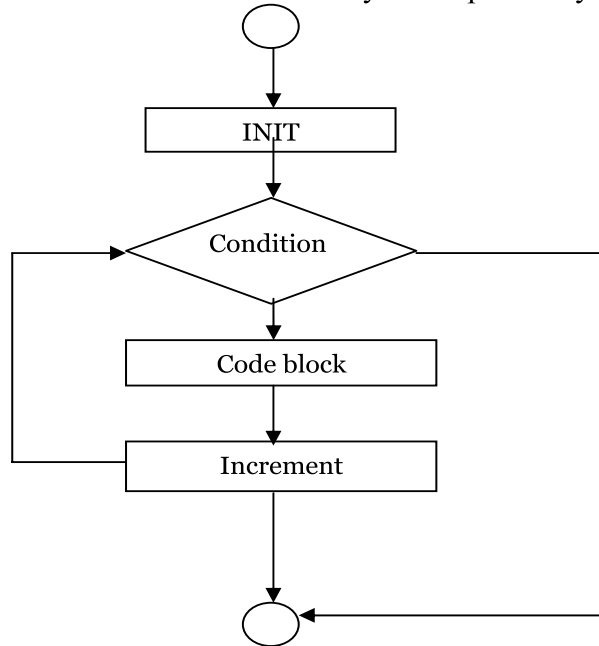
A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

The **init** step is executed first, and only once. This step allows you to declare and initialize any loop control variables.



Next, the condition is evaluated. If it is true, the body of the loop is executed. If it is false, the body of the loop does not execute and the flow of control jumps to the next statement just after the 'for' loop.

After the body of the 'for' loop executes, the flow of control jumps back up to the increment statement. This statement allows you to update any loop control variables.



**Syntax (for statement):**

```
for (initialization_Statement; test_Expression; update_Statement)
{
    // codes
}
Statement;
```

## VIII. Algorithm

## **IX. Flowchart**

## **X. 'C' Program Code:**

## XI. Resources required

Sr. No.	Name of Resource	Specification	Qty.	Remarks
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB and onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## XII. Precautions

1. Use of for loop syntax to be decided first.
2. Loop control variable shall be initialized properly along with step, and termination criteria.
3. Loop should not go in infinite loop execution.

## XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## XIV. Result

.....  
 .....  
 .....

## XV. Conclusion(s)

.....  
 .....  
 .....  
 .....

## XVI. Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. What will be the output given program?

```
#include<stdio.h>
void main()
{
    int i = -10;
    for(;i;printf("%d ", i++));
}
```

- A. -10 to -1
- B. -10 to infinite
- C. -10 to 0
- D. Compiler error

2. What will be the output of given program?

```
#include<stdio.h>
Void main ()
{
    int a=3;
    for (;a;printf("%d ", a--);
}
A. no output
B. 3 2 1 0
C. 3 2 1
D. infinity loop
```

3. The following code 'for(;;)' represents an infinite loop. It can be terminated by.

- a) break
- b) exit(0)
- c) abort()
- d) All of the mentioned

4. What is the output of following program?

```
#import
int main ()
{
    for( ; ; )
    {
        NSLog(@"This loop will run forever.\n");
    }

    return 0;
}
```

4. Which of the following cannot be used as LHS of the expression in for (exp1;exp2; exp3) ?

- a) Variable
- b) Function
- c) typedef
- d) macros

**[Space for answers]**

.....

.....

.....

.....



## XVII. Exercise

**(Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write a program to count the number of digit in a given number
2. Display table of square and Cube of first 10 numbers.

**[Space for answers]**

[illegible]

.....

.....

.....

.....

.....

### XVIII. References / Suggestions for further Reading

1. <http://www.examveda.com/c-program/practice-mcq-question-on-c-fundamentals/>
2. <http://fresh2refresh.com/c-programming/c-loop-control-statements/>
3. <https://www.programiz.com/c-programming/c-for-loop>

### XV Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

### List of Student Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No. 10: Implement Loop Control Statements in 'C'.

### Problem Statement:

- (a) Find Fibonacci series for given number.
- (b) Write a program to produce the following output:

```
      1
    2   3
  4   5   6
7  8   9   10
```

### I. Practical Significance

Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.

### II. Relevant Program Outcomes (POs)

- 1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
- 2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
- 3. **Engineering tools:** Apply relevant Computer programming / electronic technologies and tools with an understanding of the limitations.
- 4. **Communication:** Communicate effectively in oral and written form.

### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

- 1. Write algorithm and Draw Flow Chart.
- 2. Write and Save a simple C program for function using pointers.
- 3. Compile (Turbo C) the C program.
- 4. Debug and execute the program.
- 5. Execute the program

### IV. Relevant Course Outcome(s)

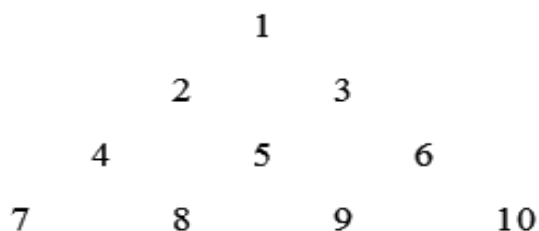
Implement decision making in 'C' programming

### V. Practical Outcome (POs)

Write a program for.

- 1. Find Fibonacci series for given number.
- 2. Write a program to produce the following output:





**VI. Relevant Affective domain related Outcome(s)**

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

**VII. Minimum Theoretical Background (prerequisite to be included)**

The Fibonacci of a given number using recursion. In fibonacci series, each number is the sum of the two preceding numbers. Eg: 0, 1, 1, 2, 3, 5, 8 ...

A loop inside another loop is known as nested loop. We can write any loop inside any loop in c i.e. we can write for loop inside the loop or while loop or do while loop etc.

**Syntax for a nested for loop**

```
for ( init; condition; increment )
{
    for ( init; condition; increment )
    {
        statement(s);
    }
    statement(s);
}
```

A final note on loop nesting is that you can put any type of loop inside of any other type of loop. For example, a for loop can be inside a while loop or vice versa.

**VIII. Algorithm**

## **IX. Flow Chart**

## **X. 'C' Program Code**

## **XI. Resources required**

<b>Sr. No.</b>	<b>Name of Resource</b>	<b>Specification</b>	<b>Qty.</b>	<b>Remarks</b>
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## **XII. Precautions**

1. Handle computer system with care.
2. Strictly follow the instruction for writing, compiling and executing the program.
3. Selection of proper control statement is expected to display proper output.
4. Output format of problem statement should be strictly obtained.

## **XIII. Resources used**

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## **XIV. Results**

.....  
.....  
.....

## **XV. Conclusion(s)**

.....  
.....  
.....

## **XVI. Practical Related Questions**

**Note:** Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.

1. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    int k;
    for (k = -3; k < -5; k++)
        printf("Hello");
}
```

- a) Hello
- b) Infinite hello
- c) Run time error
- d) Nothing

2. What is the output of this C code?

```
#include <stdio.h>
void main()
{
    double k = 0;
    for (k = 0.0; k < 3.0; k++)
        printf ("Hello");
}
```

- a) Run time error

- [Space for answers]**

This image shows a full page of a notebook or worksheet. It features approximately 28 horizontal dotted lines spaced evenly down the page, providing a guide for handwriting practice. The lines are light gray and extend across the entire width of the page. There is no text or other markings on the page.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**XVII Exercise:**

(**Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write a program to produce the following output:

```
*
* *
* * *
* * * *
* * * * *
```

2. Write a program to produce the following output:

```
10      9      8      7
6       5      4
3       2
1
```

**[Space for answers]**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



### XVIII References / Suggestions for further Reading

1. <https://www.programiz.com/c-programming/c-decision-making-loops-examples>
2. [https://www.tutorialspoint.com/objective\\_c/objective\\_c\\_nested\\_loops.html](https://www.tutorialspoint.com/objective_c/objective_c_nested_loops.html)

### XIX Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### List of Student Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No. 11:

**Print the Result sheet:** Conditions given are: marks  $\geq 40\%$  pass, marks  $< 40\%$  fail, marks  $60 \geq$  first class, marks above 75 % distinction, marks  $> 100$  and marks  $< 0$  not valid.

### I. Practical Significance

This practical is useful for students to understand the use of different operators and use computational capability of 'C'.

### II. Relevant Program Outcomes (POs)

1. **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the broad-based Electronics related problems.
2. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
3. **The engineer and society:** Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics.
4. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program

### IV. Relevant Course Outcome(s)

Implement decision making in 'C' programming.

### V. Practical Outcome (POs)

Print the Result sheet: Conditions given are: marks  $\geq 40\%$  pass, marks  $< 40\%$  fail, marks  $60 \geq$  first class, marks above 75 % distinction, marks  $> 100$  and marks  $< 0$  not valid.

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.



**VII. Minimum Theoretical Background** (prerequisite to be included)

Refer following table for grades of students using else if ladder.

Percentage	Grades
$\geq 100$ and $< 0$	Not valid
$\geq 75$	distinction
$\geq 60$	First class
$\geq 40$	pass
$< 40$	fail

**VIII. Algorithm**

## **IX. Flow Chart**

## **X. 'C' Program Code**

## XI. Resources required

Sr. No.	Name of Resource	Specification	Qty.	Remarks
1	Hardware: Computer, System	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## XII. Precautions

1. Handle computer system with care.
2. Strictly follow the instruction for writing, compiling and executing the program.
3. Check the relevant output and formatting of output.

## XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## XIV. Results ( Output of the Program )

.....  
 .....  
 .....

## XV. Conclusion (s)

.....  
 .....  
 .....

## XVI. Practical Related Questions

**Note:** Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.

1. What will happen if you use Increment and decrement operators on constant?
2. What is the output of this C code?

```
#include <stdio.h>
int main()
{
    int i = 2;
    int j = ++i + i;
    printf("%d\n", j);
}
```

3. Solve the following expression.

$$z = ++x + y-- - ++y - x-- - x-- - ++y - x--$$

where  $x = 7$  and  $y = -3$

**[Space for answers]**

[illegible]

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

.....

.....

.....

.....

.....

**XVIII. References / Suggestions for further Reading**

1. <http://cprogramsbasics.blogspot.com/2014/12/c-program-find-grades-if-ladder.html>
2. <http://www.plus2net.com/c-tutorial/grade.php>
3. <http://www.yourowncodes.com/2014/03/simple-if-statement-in-c-programming.html>

**XIX. Assessment Scheme**

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

**List of Students / Team Members**

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No. 12: Processing Elements of Array

### Problem Statement:

1. Write a program to declare, modify and print elements of a given data array.
2. Write a program to find highest marks in a class of n students using array.

### I. Practical Significance

Arrays are useful data types that often show up when it would be convenient to have one name for a group of variables of the same type that can be accessed by a numerical index. For example, a tic-tac-toe board can be held in an array and each element of the tic-tac-toe board can easily be accessed by its position (the upper left might be position 0 and the lower right position 8). Arrays can be defined as one dimensional and two dimensional.

### II. Relevant Program Outcomes (POs)

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **The engineer and society:** Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics.
4. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### IV. Relevant Course Outcomes

Use Arrays and string in 'C' programming.

### V. Practical Outcome (POs)

#### Processing elements of Array

1. Write a program to declare, modify and print elements of a given data array.
2. Write a program to find highest marks in a class of n students using array

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

## VII. Minimum Theoretical Background (prerequisite to be included)

C Array is a collection of variables belonging to the same data type. You can store group of data of same data type in an array.

1. Array might be belonging to any of the data types
2. Array size must be a constant value.
3. Always, Contiguous (adjacent) memory locations are used to store array elements in memory.
4. It is a best practice to initialize an array to zero or null while declaring, if we don't assign any values to array.

For example: if you want to store marks of 100 students, you can create an array for it.

```
int marks [100];
```

### **Array declaration syntax:**

```
data_type arr_name [arr_size];
```

```
i.e. int stud_no[20];
```

### **Array initialization syntax:**

```
data_type arr_name [arr_size]=(value1, value2, value3,...);
```

```
char stud_name[4]={'sunil ','pratik',divyansh','sonali'}
```

### **Array accessing syntax:**

```
arr_name[index];
```

```
roll_no[i];
```

```
roll_no[12];
```

## VIII. Algorithm



## **IX. Flow Chart**

## **X. 'C' Program Code**

## XI. Resources required

Sr. No.	Name of Resource	Specification	Qty.	Remarks
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## XII. Precautions

1. Handle computer system with care.
2. C does not check if an array index declared or not.
3. Array size should be integer only.
4. Initialization of array is must.
5. Reserved words cannot be used as user identifiers and array name.

## XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## XIV. Result

.....  
 .....  
 .....

## XV. Conclusion(s)

.....  
 .....  
 .....

## XVI. Practical Related Questions

**Note:** Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.

1. Which of the following declaration are illegal?
  - a) `int a[ ][ ] = { {1, 2, 3}, {2, 3, 4, 5} };`
  - b) `int *a[ ] = { {1, 2, 3}, {2, 3, 4, 5} };`
  - c) `int a[4][4] = { {1, 2, 3}, {2, 3, 4, 5} };`
  - d) Both (a) and (b).
2. What is right way to Initialize array?
  - a) `int num[6] = { 2, 4, 12, 5, 45, 5 };`
  - b) `int n{ } = { 2, 4, 12, 5, 45, 5 };`
  - c) `int n{6} = { 2, 4, 12 };`
  - d) `int n(6) = { 2, 4, 12, 5, 45, 5 };`

3. An array elements are always stored in \_\_\_\_\_ memory locations.
- a) Sequential
  - b) Random
  - c) Sequential and Random
  - d) None of the above
4. Let x be an array. Which of the following operations are illegal?
- I. ++x
  - II. x+1
  - III. x++
  - IV. x\*2
- a) I and II
  - b) I, II and III
  - c) II and III
  - d) I, III and IV
  - e) III and IV
5. What will be the output of following program code?
- ```
#include <stdio.h>
int main(void)
{
    char p;
    char buf[10] = {1, 2, 3, 4, 5, 6, 9, 8};
    p = (buf + 1)[5];
    printf("%d", p);
    return 0;
}
```
- a) 5
  - b) 6
  - c) 9
  - d) Error
  - e) None of the above

**(Space for answers)**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

(**Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write a C Program to calculate sum of an array.
2. Write a C Program to calculate average of an array

[illegible]

[illegible]

# XVIII. References / Suggestions for further Reading

1. [https://www.tutorialspoint.com/objective\\_c/objective\\_c\\_pointers.htm](https://www.tutorialspoint.com/objective_c/objective_c_pointers.htm)spoken-tutorial.org
2. <http://www.c4learn.com/c-programs/c-program-using-pointers-read-array-of-integers-and-reverse-it.html>
3. [https://www.tutorialspoint.com/objective\\_c/objective\\_c\\_arrays.htm](https://www.tutorialspoint.com/objective_c/objective_c_arrays.htm)
4. [https://www.tutorialspoint.com/learn\\_c\\_by\\_examples/array\\_examples\\_in\\_c.htm](https://www.tutorialspoint.com/learn_c_by_examples/array_examples_in_c.htm)

# XIX. Assessment Scheme

| Performance indicators     |                                                         | Weightage (Marks) |
|----------------------------|---------------------------------------------------------|-------------------|
| Process related(10 Marks)  |                                                         | 40%               |
| 1.                         | Debugging ability                                       | 25%               |
| 2.                         | Follow ethical practices.                               | 15%               |
| Product related (15 Marks) |                                                         | 60%               |
| 3.                         | Correctness of algorithm                                | 10%               |
| 4.                         | Correctness of Flow chart                               | 10%               |
| 5.                         | Correctness of Program codes                            | 20%               |
| 6.                         | Quality of input/output messaging and output formatting | 5%                |
| 7.                         | Timely Submission of report                             | 5%                |
| 8.                         | Answer to sample questions                              | 10%               |
| Total (25 Marks)           |                                                         | 100%              |

## List of Students / Team Members

1. ....
2. ....
3. ....
4. ....

| Marks Obtained      |                     |           | Dated signature of Teacher |
|---------------------|---------------------|-----------|----------------------------|
| Process Related(10) | Product Related(15) | Total(25) |                            |
|                     |                     |           |                            |

## Practical No. 13: Write Programs Using Array

### Problem Statement:

- a) Write a program to copy of one array into second array for given data elements.
- b) Write a program to create an array by reversing the elements of the given array.

### I. Practical Significance

Arrays are important to Objective-C and need lots of more details. Copying an array involves index-by-index copying. For this to work we shall know the length of array in advance, which we shall use in iteration. Another array of same length shall be required, to which the array will be copied.

### II. Relevant Program Outcomes (POs)

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **Engineering tools:** Apply relevant Computer programming / electronic technologies and tools with an understanding of the limitations.
4. **Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.

### III. Competency and Practical skills

. This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### IV. Relevant Course Outcome(s)

Use Arrays and string in 'C' programming.

### V. Practical Outcome (POs)

1. Write a program to copy of one array into second array for given data elements
2. Write a program to create an array by reversing the elements of the given array.

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

### VII. Minimum Theoretical Background (prerequisite to be included)

- Below is the step by step descriptive logic to make copy of an array.
- Input size and elements in array; store it in.
- Declare another to store copy of the first array, say second.
- Now, to copy all elements from first to second array, you just need to run loop through each elements of first array.

- Run a loop from 0 to size of the array. The loop structure should look like for(i=0; i<size; i++).
- Inside the loop assign the current element value of first array to second i.e. second[i] = first[i].

START

Step 1 → Take two arrays A, B

Step 2 → Store values in A

Step 3 → Loop for each value of A

Step 4 → Copy each index value to B array at the same index location

STOP

### **VIII. Algorithm**



## **IX. Flow Chart**

## **X. 'C' Program Code**

## XI. Resources required

| Sr. No. | Name of Resource          | Specification                                   | Qty.              | Remarks             |
|---------|---------------------------|-------------------------------------------------|-------------------|---------------------|
| 1       | Hardware: Computer System | Computer, (i3-i5 preferable), RAM minimum 2 GB. | As per batch size | For all Experiments |
| 2       | Operating system          | Windows XP/Windows 7/LINUX version 5.0 or later |                   |                     |
| 3       | Software                  | Turbo C /C++ Version 3.0 or later               |                   |                     |

## XII. Precautions

- 1 C does not check if an array index declared or not.
- 2 Array size should be integer only.
- 3 Initialization of array is must.
- 4 Reserved words cannot be used as user identifiers and array name.
- 5 Previously declared variables or functions cannot be redeclared.

## XIII. Resources used

| S. No. | Name of Resource                          | Specification |
|--------|-------------------------------------------|---------------|
| 1      | Computer System with broad specifications |               |
| 2      | Software                                  |               |
| 3      | Any other resource used                   |               |

## XIV. Result

.....  
 .....

## XV. Conclusion(s)

.....  
 .....

## XVI. Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. Check whether one of element of one array is zero will it be copied to other array?
2. Can we access array element by initializing first element index as arr[1] instead of arr[0]?
3. Do array subscripts always start with zero?
4. Can the sizeof operator be used to tell the size of an array passed to a function?
5. What will be output if you will execute following c code?  

```
#include<stdio.h>
void main()
{
    char arr[7]="Network";
    printf("%s",arr);
}
```

- (Space for answers)**

This image shows a full page of primary-ruled paper. It features approximately 28 horizontal dotted lines spaced evenly down the page, providing a guide for handwriting practice. The paper is otherwise blank, with no margins, text, or other markings.

[illegible]

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

.....

.....

.....

.....

.....

### XVIII. References / Suggestions for further Reading

1. <http://fresh2refresh.com/c-programming/c-array/>
2. <http://www.techcrashcourse.com/2015/03/c-program-to-reverse-an-array.html>
3. <http://www.tutorialgateway.org/c-program-to-reverse-an-array/>

### XIX. Assessment Scheme

| Performance indicators            |                                                         | Weightage (Marks) |
|-----------------------------------|---------------------------------------------------------|-------------------|
| <b>Process related(10 Marks)</b>  |                                                         | <b>40%</b>        |
| 1.                                | Debugging ability                                       | 25%               |
| 2.                                | Follow ethical practices.                               | 15%               |
| <b>Product related (15 Marks)</b> |                                                         | <b>60%</b>        |
| 3.                                | Correctness of algorithm                                | 10%               |
| 4.                                | Correctness of Flow chart                               | 10%               |
| 5.                                | Correctness of Program codes                            | 20%               |
| 6.                                | Quality of input/output messaging and output formatting | 5%                |
| 7.                                | Timely Submission of report                             | 5%                |
| 8.                                | Answer to sample questions                              | 10%               |
| <b>Total (25 Marks)</b>           |                                                         | <b>100%</b>       |

### List of Students / Team Members

1. ....
2. ....
3. ....
4. ....

| Marks Obtained      |                     |           | Dated signature of Teacher |
|---------------------|---------------------|-----------|----------------------------|
| Process Related(10) | Product Related(15) | Total(25) |                            |
|                     |                     |           |                            |

## Practical No. 14: Write Programs Using Multidimensional Array

### Problem Statement:

- (a) Write a program to sort numbers in ascending and descending in a given array.
- (b) Write a program to add two matrices of size 3\*3 store additions in third matrix for given data elements.

### I. Practical Significance

In C programming, you can create an array of arrays known as multidimensional array. It has row and column. Row represents one dimension and column represents second dimension. Matrix addition is a simple process. Addition of two matrices can be done only and only if both matrices are of same size.

### II. Relevant Program Outcomes (POs)

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **Engineering tools:** Apply relevant Computer programming / electronic technologies and tools with an understanding of the limitations.
4. **Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.

### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### IV. Relevant Course Outcomes

Use Arrays and string in 'C' programming.

### V. Practical Outcome (POs)

1. Write a program to sort numbers in ascending and descending in a given array.
2. Write a program to add two matrices of size 3\*3 store additions in third matrix for given data elements.

### VI Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

## VII Minimum Theoretical Background

|       | Column 1 | Column 2 | Column 3 | Column 4 |
|-------|----------|----------|----------|----------|
| Row 1 | X[0][0]  | X[0][1]  | X[0][2]  | X[0][3]  |
| Row 2 | X[1][0]  | X[1][1]  | X[1][2]  | X[1][3]  |
| Row 3 | X[2][0]  | X[2][1]  | X[2][2]  | X[2][3]  |

In C programming, you can create an array of arrays known as multidimensional array. For example, `int x[3][4]`; Here, x is a two-dimensional (2d) array. The array can hold 12 elements. Imagine an array as table with 3 row and each row has 4 column

**Example:**

| Matrix 1: | Matrix 2: | Sum of both matrix : |
|-----------|-----------|----------------------|
| 1 2 3     | 9 8 7     | 10 10 10             |
| 4 5 6     | 6 5 4     | 10 10 10             |
| 7 8 9     | 3 2 1     | 10 10 10             |

## VIII Algorithm

## **IX Flow Chart**

## **X 'C' Program Code**



## XI Resources required

| Sr. No. | Name of Resource          | Specification                                           | Qty.              | Remarks             |
|---------|---------------------------|---------------------------------------------------------|-------------------|---------------------|
| 1       | Hardware: Computer System | Computer, (i3-i5 preferable), RAM minimum 2 GB onwards. | As per batch size | For all Experiments |
| 2       | Operating system:         | Windows XP/Windows 7/LINUX version 5.0 or later         |                   |                     |
| 3       | Software                  | Turbo C /C++ Version 3.0 or later                       |                   |                     |

## XII Precautions

1. C does not check if an array index declared or not.
2. Array size should be integer only.
3. Initialization of array is must.
4. Reserved words cannot be used as user identifiers and array name.

## XIII Resources used

| S. No. | Name of Resource                          | Specification |
|--------|-------------------------------------------|---------------|
| 1      | Computer System with broad specifications |               |
| 2      | Software                                  |               |
| 3      | Any other resource used                   |               |

## XIV. Result

.....  
 .....  
 .....

## XV. Conclusion(s)

.....  
 .....  
 .....

## XVI. Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. How to pass a 2D array as a parameter in C?
2. How to initialize a multidimensional array?
3. What is the maximum number of dimensions an array in C may have?
  - A. 2
  - B. 8
  - C. 20
  - D. 50
  - E. Theoretically no limit. The only practical limits are memory size and compilers.

4. Size of the array need not be specified, when
- A. Initialization is a part of definition
  - B. It is a declaration
  - C. It is a formal parameter
  - D. All of these

5. Consider the following type definition.

```
typedef char x[10];  
x myArray[5];
```

What will sizeof (myArray) be ? (Assume one character occupies 1 byte)

- A.15
  - B.10
  - C.50
  - D.30
  - E.None of these
6. What will be printed after execution of the following code?

```
void main()  
{  
    int arr[10] = {1,2,3,4,5};  
    printf("%d", arr[5]);  
}
```

- A.Garbage Value
  - B.5
  - C.6
  - D.0
  - E.None of these
7. What will be the output of the following program?

```
void main()  
{  
    char str1[] = "abcd";  
    char str2[] = "abcd";  
    if(str1==str2)  
        printf("Equal");  
    else  
        printf("Unequal");  
}
```

- A.Equal
- B.Unequal
- C.Error
- D.None of these.

**(Space for answers)**

.....

.....

.....

.....

.....



[illegible]

(**Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write a program to add two matrices of size 3\*3 store additions in third matrix for given data elements.
2. Write a program to C program for matrix addition.

[illegible]

.....

.....

.....

### XVIII. References / Suggestions for further Reading

1. <http://www.codeforwin.in/2015/07/c-program-to-sort-array-in-ascending-order.html>
2. <https://www.programiz.com/c-programming/c-multi-dimensional-arrays>
3. <http://www.geeksforgeeks.org/pass-2d-array-parameter-c/>
4. <http://www.studystreet.com/program-sort-array-descending-order/>
5. [https://www.tutorialspoint.com/learn\\_c\\_by\\_examples/average\\_of\\_array\\_in\\_c.html](https://www.tutorialspoint.com/learn_c_by_examples/average_of_array_in_c.html)

### XIX. Assessment Scheme

| Performance indicators            |                                                         | Weightage (Marks) |
|-----------------------------------|---------------------------------------------------------|-------------------|
| <b>Process related(10 Marks)</b>  |                                                         | <b>40%</b>        |
| 1.                                | Debugging ability                                       | 25%               |
| 2.                                | Follow ethical practices.                               | 15%               |
| <b>Product related (15 Marks)</b> |                                                         | <b>60%</b>        |
| 3.                                | Correctness of algorithm                                | 10%               |
| 4.                                | Correctness of Flow chart                               | 10%               |
| 5.                                | Correctness of Program codes                            | 20%               |
| 6.                                | Quality of input/output messaging and output formatting | 5%                |
| 7.                                | Timely Submission of report                             | 5%                |
| 8.                                | Answer to sample questions                              | 10%               |
| <b>Total (25 Marks)</b>           |                                                         | <b>100%</b>       |

### List of Student Team Members

1. ....
2. ....
3. ....
4. ....

| Marks Obtained      |                     |           | Dated signature of Teacher |
|---------------------|---------------------|-----------|----------------------------|
| Process Related(10) | Product Related(15) | Total(25) |                            |
|                     |                     |           |                            |

## **Practical No. 15: PERFORM OPERATIONS ON STRING USING FUNCTION**

### **Problem Statement:**

- (a) Write a program that accept a string from user and print that string.
- (b) Write a program that accept a string and compare it with existing string.

### **I. Practical Significance**

In C programming, Strings are actually one-dimensional array of characters terminated by a **null** character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a **null**.

### **II. Relevant Program Outcomes (POs)**

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **The engineer and society:** Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics.
4. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry

### **III. Competency and Practical skills**

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### **IV. Relevant Course Outcomes**

1. Use Arrays and string in 'C' programming.
2. Use functions in 'C' programs for modular programming approach.

### **V. Practical Outcome (POs)**

1. Write a program that accept a string from user and print that string.
2. Write a program that accept a string and compare it with existing string.

### **VI. Relevant Affective domain related Outcome(s)**

1. Select proper programming environment in C.
2. Follow ethical practices.

### **VII. Minimum Theoretical Background**

The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello."

```
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

If you follow the rule of array initialization then you can write the above statement as follows – `Char greeting [ ] = "Hello";`

Following is the memory presentation of the above defined string in C

| Index    | 0       | 1       | 2       | 3       | 4       | 5       |
|----------|---------|---------|---------|---------|---------|---------|
| Variable | H       | e       | l       | l       | o       | \0      |
| Address  | 0X23451 | 0X23452 | 0X23453 | 0X23454 | 0X23455 | 0X23456 |

Actually, you do not place the *null* character at the end of a string constant. The C compiler automatically places the '\0' at the end of the string when it initializes the array.

We are directly assigning String to variable by writing text in double quotes. In this type of initialization, we don't need to put NULL or Ending / Terminating character at the end of string. It is appended automatically by the compiler.

Ex: `char name [ ] = "PRITESH";`

**strcmp(s1, s2);** Returns 0 if s1 and s2 are the same; less than 0 if s1<s2;  
greater than 0 if s1>s2.

## VIII. Algorithm

## **IX. Flowchart**

## **X. 'C' Program Code**



### XI. Resources required

| Sr. No. | Name of Resource          | Specification                                           | Qty.              | Remarks             |
|---------|---------------------------|---------------------------------------------------------|-------------------|---------------------|
| 1       | Hardware: Computer System | Computer, (i3-i5 preferable), RAM minimum 2 GB onwards. | As per batch size | For all Experiments |
| 2       | Operating system:         | Windows XP/Windows 7/LINUX version 5.0 or later         |                   |                     |
| 3       | Software                  | Turbo C /C++ Version 3.0 or later                       |                   |                     |

### XII. Precautions

1. Previously declared variables or functions cannot be redeclared.
2. Handle computer system with care.
3. Strictly follow the instruction for writing, compiling and executing the program.
4. Start and Shutdown system with proper procedure
5. Follow safety practices.

### XIII. Resources used

| S. No. | Name of Resource                          | Specification |
|--------|-------------------------------------------|---------------|
| 1      | Computer System with broad specifications |               |
| 2      | Software                                  |               |
| 3      | Any other resource used                   |               |

### XIV. Result

.....  
 .....  
 .....

### XV. Conclusion(s)

.....  
 .....  
 .....  
 .....

### XVI. Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. If the two strings are identical, then strcmp() function returns
  - A. 1
  - B. 0
  - C.-1
  - D. true
  - E. None of these

2. Which of the following function is more appropriate for reading in a multi-word string?
- A.scanf()
  - B.gets()
  - C.printf()
  - D.puts()
  - E.None of these
3. What will be the output of the program?
- ```
#include<stdio.h>
#include<string.h>
void main()
{
    char str1[20] = "Hello", str2[20] = " World";
    printf("%sn", strcpy(str2, strcat(str1, str2)));
}
```
- A.Hello World
  - B.World
  - C.WorldHello
  - D.Hello
  - E.None of these

**(Space for answers)**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[illegible]

[illegible]

### XVIII. References / Suggestions for further Reading

1. [https://www.tutorialspoint.com/learn\\_c\\_by\\_examples/string\\_programs\\_in\\_c.htm](https://www.tutorialspoint.com/learn_c_by_examples/string_programs_in_c.htm)
2. <https://www.javatpoint.com/c-programming-language-tutorial>
3. <http://fresh2refresh.com/c-programming/c-strings/c-strncat-function/>

### XIX. Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### List of Student Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No.16: Implement String Functions in C.

### Problem Statement:

- a) Write a program to accept and concatenate two strings.
- b) Write a program to find length of a string.

### I. Practical Significance

The C language provides no explicit support for strings in the language itself. The string-handling functions are implemented in libraries. String I/O operations are implemented in <stdio.h> (puts, gets, etc). A set of simple string manipulation functions are implemented in <string.h>, or on some systems in <strings.h>.

### II. Relevant Program Outcomes

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **The engineer and society:** Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics..
4. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile program (Turbo C) the C.
4. Debug and execute the program.

### IV. Relevant Course Outcomes(s)

1. Use Arrays and string in 'C' programming.
2. Use functions in 'C' programs for modular programming approach.

### V. Practical Outcome (POs)

1. Write a program to accept and concatenate two strings.
2. Write a program to find length of a string.

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

### VII. Minimum Theoretical Background

**strlen(s1);** Returns the length of string s1.

strlen( ) function counts the number of characters in a given string and returns the integer value.

It stops counting the character when null character is found. Because, null character indicates the end of the string in C.

strcat( ) function in C language concatenates two given strings. It concatenates source string at the end of destination string. Syntax for strcat( ) function is given below.

**char \* strcat ( char \* destination, const char \* source );**

**Example:**

strcat ( str2, str1 ); – str1 is concatenated at the end of str2.

strcat ( str1, str2 ); – str2 is concatenated at the end of str1.

## **VIII. Algorithm**

## **IX. Flowchart**

## **X. C' Program Code**

## **XI. Resources required**

<b>Sr. No.</b>	<b>Name of Resource</b>	<b>Specification</b>	<b>Qty.</b>	<b>Remarks</b>
1	Hardware: Personal computer,	Personal computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size ( One computer per two students with necessary software installed)	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		



## **XII. Precautions**

1. You cannot decrement a pointer once incremented.
2. Value Filed of Pointer Can Be Accessed Using '\*' Operator
3. Pointer of one data type should not be assigned to another data type without type cast.
4. Pointers should not use with multiply, divide, modulo operators.
5. Do not change the address of array.
6. whenever you declare a pointer initialize them to the required field or NULL
7. Follow safety practices.

## **XIII. Resources used**

<b>S. No.</b>	<b>Name of Resource</b>	<b>Specification</b>
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## **XIV. Result**

.....  
.....  
.....

## **XV. Conclusion(s)**

.....  
.....  
.....  
.....

## **XVI. Practical Related Questions**

***Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.***

1. How pointers are useful?
2. Differentiate array and pointers.

**(Space for answers)**

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

## XVII. Exercise

**(Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

- Write a Program to search strings in C
- Write a Program to swap strings in C

**(Space for answers)**



.....

.....

.....

.....

.....

### XVIII. References / Suggestions for further Reading

- <http://fresh2refresh.com/c-programming/c-strings/c-strncat-function/>
- <http://fresh2refresh.com/c-programming/c-strings/c-strlen-function/>
- <http://fresh2refresh.com/c-programming/c-strings/c-strrev-function/>

### XIX. Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

### List of Students / Team Members

- .....
- .....
- .....
- .....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No. 17: Write C Program to Perform Library Function

### Problem Statement:

Develop Program to demonstrate:

- a) Use of all string handling functions.
- b) Use of few Mathematical functions

### I. Practical Significance

String.h header file supports all the string functions in C language. All the string functions are given below. All C inbuilt functions which are declared in Math.h header file are given below. The source code for Math.h header file is also given below for your reference.

### II. Relevant Program Outcomes

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **Engineering tools:** Apply relevant Computer programming / electronic technologies and tools with an understanding of the limitations.
4. **The engineer and society:** Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics.

### III. Competency and related skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Draw Flow Chart for given Algorithm.
2. Write and Save a simple C program using editor.
3. Compile (Using Turbo C) a C Program.
4. Debug the program.
5. Execute the program.

### IV. Relevant Course Outcome(s)

1. Use pointers to increase efficiency of programs
2. Use functions in 'C' programs for modular programming approach.

### V. Practical Learning Outcome (PLO)

1. Write a program to add two integer numbers using pointer.
2. Write a program to calculate the sum of elements of given array using pointer.

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

## VII. Minimum Theoretical Background

### List of String Library function:

String functions	Description
<u>strcat ( )</u>	Concatenates str2 at the end of str1
<u>strncat ( )</u>	Appends a portion of string to another
<u>strcpy ( )</u>	Copies str2 into str1
<u>strncpy ( )</u>	Copies given number of characters of one string to another
<u>strlen ( )</u>	Gives the length of str1
<u>strcmp ( )</u>	Returns 0 if str1 is same as str2. Returns <0 if str1 < str2. Returns >0 if str1 > str2
<u>strcmpi ( )</u>	Same as strcmp() function. But, this function negotiates case. "A" and "a" are treated as same.
<u>strchr ( )</u>	Returns pointer to first occurrence of char in str1
<u>strrchr ( )</u>	last occurrence of given character in a string is found
<u>strstr ( )</u>	Returns pointer to first occurrence of str2 in str1
<u>strrstr ( )</u>	Returns pointer to last occurrence of str2 in str1
<u>strdup ( )</u>	Duplicates the string
<u>strlwr ( )</u>	Converts string to lowercase
<u>strupr ( )</u>	Converts string to uppercase
<u>strrev ( )</u>	Reverses the given string
<u>strset ( )</u>	Sets all character in a string to given character
<u>strnset ( )</u>	It sets the portion of characters in a string to given character
<u>strtok ( )</u>	Tokenizing given string using delimiter

**“Math.h” header file** supports all the mathematical related functions in C language. All the arithmetic functions used in C language are given below.

Function	Description
<u>floor ( )</u>	This function returns the nearest integer which is less than or equal to the argument passed to this function.
<u>round ( )</u>	This function returns the nearest integer value of the float/double/long double argument passed to this function. If decimal value is from “.1 to .5”, it returns integer value less than the argument. If decimal value is from “.6 to .9”, it returns the integer value greater than the argument.
<u>ceil ( )</u>	This function returns nearest integer value which is greater than or equal to the argument passed to this function.
<u>sin ( )</u>	This function is used to calculate sine value.
<u>cos ( )</u>	This function is used to calculate cosine.
<u>cosh ( )</u>	This function is used to calculate hyperbolic cosine.
<u>exp ( )</u>	This function is used to calculate the exponential “e” to the x <sup>th</sup> power.
<u>tan ( )</u>	This function is used to calculate tangent.
<u>tanh ( )</u>	This function is used to calculate hyperbolic tangent.
<u>sinh ( )</u>	This function is used to calculate hyperbolic sine.
<u>log ( )</u>	This function is used to calculates natural logarithm.
<u>log10 ( )</u>	This function is used to calculates base 10 logarithm.
<u>sqrt ( )</u>	This function is used to find square root of the argument passed to this function.
<u>pow ( )</u>	This is used to find the power of the given number.
<u>trunc.( )</u>	This function truncates the decimal value from floating point value and returns integer value.

## **VIII. Algorithm**

## **IX. Flowchart**



## **X. 'C' Program Code**

## **XI. Resources required**

<b>Sr. No.</b>	<b>Name of Resource</b>	<b>Specification</b>	<b>Qty.</b>	<b>Remarks</b>
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## **XII. Precautions**

1. You cannot decrement a pointer once incremented.
2. Handle computer system with care.
3. Strictly follow the instruction for writing, compiling and executing the program.
4. Start and Shutdown system with proper procedure
5. Follow safety practices.

**XIII. Resources used**

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

**XIV. Result**

.....  
.....  
.....

**XV. Conclusion(s)**

.....  
.....  
.....  
.....

**XVI. Practical Related Questions**

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1      **Use of functions**

- A. Helps to avoid repeating a set of statements many times.
- B. Enhances the logical clarity of the program.
- C. Helps to avoid repeated programming across programs.
- D. Makes the debugging task easier.
- E. All of the above

2      **Any C program**

- A. Must contain at least one function.
- B. Need not contains any function.
- C. Needs input data.
- D. None of the above

3      Write the difference between string and math function?

**(Space for answers)**

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

1. Use of all String handling functions.
2. Use of Mathematical functions.
3. Implement C program using any two String and Mathematical functions.

**(Space for answers)**

This image shows a full page of a handwriting practice worksheet. It consists of numerous horizontal rows, each defined by two parallel dotted lines. The rows are evenly spaced and extend across the entire width of the page, providing a guide for letter height and placement. There is no text or other markings on the page.

.....

.....

.....

### XVIII. References / Suggestions for further Reading

1. <http://fresh2refresh.com/c-programming/c-function/c-math-h-library-functions/>
2. [https://www.tutorialspoint.com/ansi\\_c/c\\_strncat.htm](https://www.tutorialspoint.com/ansi_c/c_strncat.htm)
3. <http://fresh2refresh.com/c-programming/c-strings/>

### XIX. Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

### List of Students / Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No. 18: Write C Program to Perform User Defined Function.

### Problem Statement:

- (a) Write a program to add two numbers using function.
- (b) Write a program to perform addition, subtraction, multiplication and division using switch case statement and user defined function for given data.

### I. Practical Significance

Each C program can be a collection of function(s). A **function** is a group of statements that together perform a task. Each **C program** has at least one **function**, which is **main ()**, the program execution starts from **main function**. User can create their own **functions**.

### II. Relevant Program Outcomes (POs)

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **The engineer and society:** Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics.
4. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program

### IV. Relevant Course Outcome(s )

Use functions in 'C' programs for modular programming approach.

### V. Practical Outcome (POs)

1. Write a program to add two numbers using function.
2. Write a program to perform addition, subtraction, multiplication and division using switch case statement and user defined function for given data.

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

## VII. Minimum Theoretical Background

The function calling procedure will use in this program to find the sum of two numbers. First user ask the numbers which are use to add. Then it declares a function sum (a, b). The next move it call the function sum with formal arguments int x, int y and int z=x+y in this procedure the value call from the upper class but add in other class and in the last the result print on the screen.

A function is a block of code that performs a specific task. C allows you to define functions according to your need. These functions are known as user-defined functions.

For example:

To create a circle and color it depending upon the radius and color. You can create two functions to solve this problem:

- createCircle() function
- color() function

A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each switch case.

### Switch case Syntax:

```
Switch (expression){  
    Case constant-expression:  
        Statement;  
        break; /* optional */  
    Case constant-expression:  
        Statement;  
        break; /* optional */  
  
    /* you can have any number of case statements */  
    default: /* Optional */  
        Statement;  
}
```

## VIII. Algorithm

## **IX. Flowchart**

## **X. 'C' Program Code**



## XI. Resources required

Sr. No.	Name of Resource	Specification	Qty.	Remarks
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## XII. Precautions

1. Handle computer system with care.
2. Strictly follow the instruction for writing, compiling and executing the program.
3. Start and Shutdown system with proper procedure
4. Follow safety practices.

## XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## XIV. Result

.....  
 .....  
 .....

## XV. Conclusion(s)

.....  
 .....  
 .....  
 .....

## XVI. Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. What will be the following code's output if choice = 'R'?

```
switch (choice)
{
    case 'R' : printf("RED");
    case 'W' : printf("WHITE");
    case 'B' : printf("BLUE");
    default : printf("ERROR");break;
}
```

A. RED

B. RED WHITE BLUE ERROR

2. Why Use a switch Statement?
3. What will be the output of the given program?

```
    }
    printf("%d", num);
```

- (Space for answers)**

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]

[illegible]

### XVIII. References / Suggestions for further Reading

1. <https://www.programiz.com/c-programming/c-user-defined-functions>
2. <http://www.c4learn.com/c-programs/c-program-to-add-two-numbers-using.html>
3. <https://blog.udemy.com/objective-c-switch/>
4. <https://www.programiz.com/c-programming/types-user-defined-functions>

### XIX. Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### List of Student Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No. 19: Write Basic C Programs Using Pointer

### Problem statement:

- a) Write a program to use address operator (&) and pointer operator (\*) for given data.
- b) Write a program to add two integer numbers using pointer

### I. Practical Significance

A pointer in C language is a variable that stores the address of another variable. A pointer in C is used to allocate memory dynamically i.e. at run time.

### II. Relevant Program Outcomes (POs)

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **The engineer and society:** Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics.
4. **Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.
5. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### IV. Relevant Course Outcomes

- Use pointers to increase efficiency of programs

### V. Practical Learning Outcome (PLOs)

1. Write a program to use address operator (&) and pointer operator (\*) for given data.
2. Write a program to add two integer numbers using pointer.

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

## VII. Minimum Theoretical Background

Pointer is a variable which stores the address of another variable. A pointer in C is used to allocate memory dynamically i.e. at run time. Simple Variable that stores a value such as integer ,float, character etc. The pointer variable might be belonging to any of the data type such as int, float, char, double, short etc.

**Pointer Syntax :** data\_type \*var\_name;

Example : int \*p; char \*p;

Where, \* is used to denote that “p” is pointer variable and not a normal variable.

**& operator :** used to find out the address of variable name. **& operator** is also known as “**Address of**” Operator

Example: int \*p;

int c =10;

p = &c

Assigns the address of variable to *c* to the pointer *p*.

## VIII. Algorithm

## **IX. Flowchart**

## **X. 'C' Program Code**



## XI. Resources required

Sr. No.	Name of Resource	Specification	Qty.	Remarks
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## XII. Precautions

1. whenever you declare a pointer initialize them to the required field or NULL
2. Strictly follow the instruction for writing, compiling and executing the program.
3. Start and Shutdown system with proper procedure

## XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## XIV Result

.....

.....

.....

## XV Conclusion(s)

.....

.....

.....

.....

## XVI Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. State the situation in 'C' program, where pointer concept can be used/implemented.
2. Is it essential to initialize NULL pointer?

**(Space for answers)**

.....

.....

.....

[illegible]

(**Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

- (Space for answers)**

This image shows a full page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, providing a template for handwriting practice. There are no margins, text, or other markings on the page.

.....

.....

.....

.....

.....

.....

### XVIII References / Suggestions for further Reading

1. <http://www.c4learn.com/c-programming/c-pointer-address-operator/>
2. <https://www.programiz.com/c-programming/c-pointers>

### XIX Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### List of Student Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No. 20: Perform Operation on Array Using Pointer

### Problem Statement:

- (a) Write a program to calculate the sum of elements of given array using pointer.
- (b) Write a program to access the array elements using pointer.

### I. Practical Significance

A pointer in C language is a variable that stores the address of another variable (\*). A pointer in C is used to allocate memory dynamically i.e. at run time. Array is collection of similar data type of elements. For allocating dynamic memory to the group of variable or array for that it needs group of pointers. In such case array of pointer concept is used.

### II. Relevant Program Outcomes

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **The engineer and society:** Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to practice in field of Electronics.
4. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

### III. Competency and Practicals skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write Algorithm for given program.
2. Draw Flow Chart for given Algorithm.
3. Write / Develop program using pointers in C editor
4. Compile (Using Turbo C) a C Program.
5. Debug and execute the program.

### IV. Relevant Course Outcomes

1. Use Arrays and string in 'C' programming.
2. Use pointers to increase efficiency of programs

### V. Practical Outcome (POs)

1. Write a program to calculate the sum of elements of given array using pointer.
2. Write a program to access the array elements using pointer.

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

## VII. Minimum Theoretical Background

Similar to that of an array of integers, there can be an array of pointers. A pointer is variable that always contains an address of another variable. An array pointer is nothing but a collection of addresses. The addresses present in the array of pointer can be addresses of isolated variables or addresses of array elements or any other addresses. For finding sum of array element, array of pointer concept is required

i.e. `int ARR[ ]`

Element	ARR[0]	ARR[1]	ARR[2]	ARR[3]	ARR[4]
Address	1000	1002	1004	1006	1008

### Example of Pointer to Array

```
void main()
{
    int i;
    int arr[5] = {1, 2, 3, 4, 5};
    int *p = arr; // same as int*p = &arr[0]
    for (i=0; i<5; i++)
    {
        printf("%d", *p);
        p++;
    }
}
```

In the above program, the pointer **\*p** will print all the values stored in the array one by one. Base address can be used (**a** in above case) as pointer and print all the values.

With the help of above program apply the logic for sum of array element using pointer.

## VIII. Algorithm

## **IX. Flowchart**

## **X. 'C' Program Code**

### XI. Resources required

Sr. No.	Name of Resource	Specification	Qty.	Remarks
1	Hardware: computer system,	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system:	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

### XII. Precautions

1. You cannot decrement a pointer once incremented.
2. Value Filed of Pointer Can Be Accessed Using '\*' Operator
3. Pointer of one data type should not be assigned to another data type without type cast.
4. Pointers should not use with multiply, divide, modulo operators.
5. Do not change the address of array.
6. whenever you declare a pointer initialize them to the required field or NULL
7. Strictly follow the instruction for writing, compiling and executing the program

### XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

### XIV. Result

.....  
 .....  
 .....

### XV. Conclusion(s)

.....  
 .....  
 .....  
 .....

### XVI. Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. State situation where pointers are useful?
2. Differentiate array and pointers.

(Space for Answer)

.....  
 .....  
 .....



This image shows a full page of primary-ruled paper. It features approximately 20 horizontal dotted lines spaced evenly down the page, providing a guide for handwriting practice. The paper is otherwise blank, with no margins, text, or other markings.

## XVII. Exercise;

**Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:**

**(Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write a program to access the array elements using pointer.
2. Write a program to demonstrate the use of & and \* operator
3. Write a program to declare a two dimensional array dynamically.

## XVIII. References / Suggestions for further Reading

1. [https://www.tutorialspoint.com/cprogramming/c\\_pointers.htm](https://www.tutorialspoint.com/cprogramming/c_pointers.htm)
2. <http://www.c4learn.com/c-programming/c-pointer-address-operator/>
3. <https://www.programiz.com/c-programming/c-pointers-arrays>

## XIX. Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

### List of Student / Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## **Practical No. 21: Write C Program to Implement Concept of Call by Value and Call by Reference in Function.**

### **Problem Statement:**

- a) Write a program to interchange given values of two variables using call by value mechanism.
- b) Write a program to interchange given values of two variables using call by reference mechanism.

### **I. Practical Significance**

There are two ways to pass value or data to function in C language: call by value and call by reference. Original value is not modified in call by value. In call by reference, original value is changed or modified

### **II. Relevant Program Outcomes (POs)**

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **Individual and team work:** Function effectively as a leader and team member in diverse/ multidisciplinary teams.
4. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

### **III. Competency and Practical skills**

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related problems.**

1. Write algorithm and Draw Flow Chart.
2. Write and Save a simple C program for function using pointers.
3. Compile (Turbo C) the C program.
4. Debug and execute the program.

### **IV. Relevant Course Outcomes**

1. Use functions in 'C' programs for modular programming approach.
2. Use pointers to increase efficiency of programs.

### **V. Practical Learning Outcome (PLOs)**

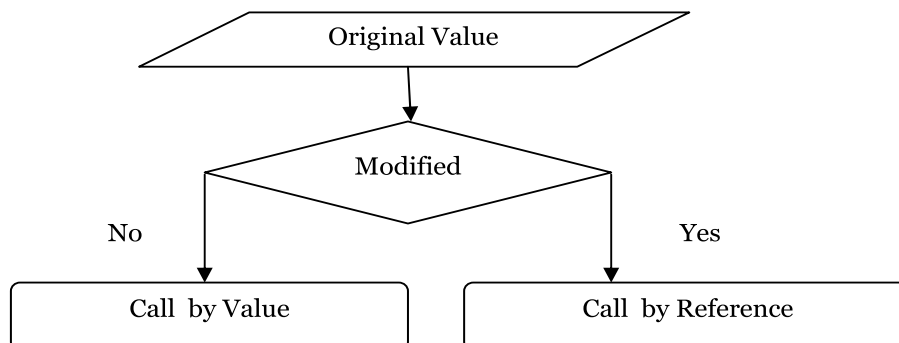
1. Write a program to interchange given values of two variables using call by value mechanism.
2. Write a program to interchange given values of two variables using call by reference mechanism.

### **VI. Relevant Affective domain related Outcome(s)**

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

## VII. Minimum Theoretical Background

Original value is not modified in call by value. Original value is modified by using call by reference.



### Call by value: -

In call by value, **original value is not modified**.

1. While Passing Parameters using call by value , xerox copy of original parameter is created and passed to the called function.
2. Any update made inside method will not affect the original value of variable in calling function.
3. In the above example num1 and num2 are the original values and xerox copy of these values is passed to the function and these values are copied into number1,number2 variable of sum function respectively.
4. As their scope is limited to only function so they cannot alter the values inside main function.

### Call by reference:-

**Original value is modified** by using call by reference. In call by reference, original value is changed or modified because we pass reference (address). Here, address of the value is passed in the function, so actual and a formal argument shares the same address space. Hence, any value changed inside the function, is reflected inside as well as outside the function.

1. While passing parameter using call by reference scheme, we are passing the actual address of the variable to the called function.
2. Any updates made inside the called function will modify the original copy since we are directly modifying the content of the exact memory location.

## VIII. Algorithm

## **IX. Flowchart**

## **X. 'C' Program Code**

## XI. Resources required

Sr. No.	Name of Resource	Specification	Qty.	Remarks
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## XII. Precautions

1. Strictly follow the instruction for writing, compiling and executing the program.
2. Syntax of loops should be followed properly.
3. Avoid infinite loop execution.

## XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

## XIV. Results

.....  
 .....  
 .....

## XV. Conclusion(s)

.....  
 .....  
 .....  
 .....

## XVI. Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. Write output of following program

```
int sum(int a, int b)
{
    int c=a+b;
    return c;
}

int main(
{
    int var1 =10;
```



.....

.....

.....

.....

.....

.....

.....

.....

.....

**XVII. Exercise**

**(Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write a program to interchange given values of two variables using call by reference mechanism.
2. Write c program using function to compute factorial of an integer number.

**(Space for answers)**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....





### XVIII. References / Suggestions for further Reading

1. [https://www.tutorialspoint.com/cprogramming/c\\_function\\_call\\_by\\_value.htm](https://www.tutorialspoint.com/cprogramming/c_function_call_by_value.htm)
2. <http://www.c4learn.com/c-programming/c-function-parameter-passing/>

### XIX. Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

#### List of Student Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No. 22: Implement Concept of Pointer Using C

### Problem statement:

Write a C program to exchange given values of two variables using pointer.

### I. Practical Significance

Pointer provides a flexible, powerful and efficient method to manipulate data in our programs. Pointer variable can be used to access memory locations. The content of memory locations can be accessed and modified easily.

### II. Relevant Program Outcomes

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **Engineering tools:** Apply relevant Computer programming / electronic technologies and tools with an understanding of the limitations.
4. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related Write and Save a simple C program for function using pointers.**

1. Write and Save a simple C program for function using pointers.
2. Develop program using function.
3. Compile (using Turbo C) Program.
4. Debug and execute the program.

### IV. Relevant Course Outcomes

Use pointers to increase efficiency of programs

### V. Practical Outcome (POs)

Write a C program to exchange given values of two variables using pointer.

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

### VII. Minimum Theoretical Background

This Program is for Swap Two Numbers Using Pointer following steps are required

1. Declaring variable pointer type.
2. Using swap condition.
3. Display result on the screen.

In this program two numbers will be swapped with each other using pointer this concept works on the address reference. Variables are declared for storing the value in it.

### **VIII. Algorithm**

### **IX. Flowchart**

## **X. 'C' Program Code**

## **XI. Resources required**

<b>Sr. No.</b>	<b>Name of Resource</b>	<b>Specification</b>	<b>Quantity</b>	<b>Remarks</b>
1	Hardware: Computer System	Computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## **XII. Precautions**

1. Start and Shutdown system with proper procedure.
2. Syntax of loops should be followed properly.
3. Avoid infinite loop execution.
4. Data type of pointer should not be changed during debugging.

### XIII. Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

### XIV. Results

.....  
 .....  
 .....

### XV. Conclusion(s)

.....  
 .....  
 .....  
 .....

### XVI. Practical Related Questions

*Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.*

1. Comment on the following pointer declaration?

```
int *ptr, p;
```

2. What will be the output of following code?

```
void main()
{
    char *p;
    p = "Hello";
    printf("%cn", *&p);
}
```

**(Space for answers)**

.....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....

**Attempt Q1. and teacher shall allot Q. 2/Q.3 from the following:**

1. Write a program to add two integers using pointers.
2. Write a program to find the area of circle using pointers.
3. Write a program using pointer to find the maximum between two inputted numbers.

.....





### XVIII. References / Suggestions for further Reading

1. <http://www.includehelp.com/c-programs/c-pointer-program-to-swap-two-numbers-using-pointers.aspx>
2. <http://www.dailyfreecode.com/code/exchange-numbers-pointers-2212.aspx>

### XIX. Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

### List of Students / Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	

## Practical No.23: Implement Structure in C.

### Problem Statement:

Create structure DATE using 'C' having members' day, month, year and assign initial values to that structure.

#### I. Practical Significance

C Structure is a collection of different data types which are grouped together under a single name and each element in a C structure is called member. If you want to access structure members in C, structure variable should be declared. Structure is for storing the complicated data.

#### II. Relevant Program Outcomes (POs)

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **Engineering tools:** Apply relevant Computer programming / electronic technologies and tools with an understanding of the limitations.
4. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

#### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related Write and Save a simple C program for function using pointers.**

1. Use syntax of structure in program.
2. Write and Save a simple C program using editor.
3. Compile Program using Turbo C.
4. Debug and execute the program.

#### IV. Relevant Course Outcomes

Implement basic concept of structure in 'C'.

#### V. Practical Outcome (POs)

Create structure DATE using 'C' having members' day, month, year and assign initial values to that structure.

#### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

#### VII. Minimum Theoretical Background

When a structure is defined, it creates a user-defined type but no storage or memory is allocated. It is nothing but the "Template / Map / Shape" of the structure. Memory is created, very first time when the variable is created / Instance is created.

### Defining structure

The **struct** keyword is used to define structure

```
struct structure_name
{
    data_type member1;
    data_type member2;
    .
    data_type memberN;
};
```

### Example to define structure for employee in c.

```
struct employee
{
    int id;
    char name[50];
    float salary;
};
```

Here, **struct** is the keyword, **employee** is the tag name of structure.  
id, name and salary are the members or fields of the structure.

### Accessing members of a structure

There are two types of operators used for accessing members of a structure.

1. Member operator (.)
2. Structure pointer operator (->)

Any member of a structure can be accessed as:

structure\_variable\_name.member\_name

## VIII. Algorithm

## **IX. Flowchart**

## **X. 'C' Program Code**

### XI. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

### XII Precautions

1. Don't forget the semicolon at the end of structure.
2. Strictly follow the instruction for writing, compiling and executing the program.
3. Start and Shutdown system with proper procedure

### XIII Resources used

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

### XIV Result

.....  
 .....

### XV Conclusion(s)

.....  
 .....  
 .....  
 .....

### XVI Practical Related Questions

**Note: Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.**

1. Difference between array and structure
2. State the true/false for following statement:
  - i. A structure is a convenient way of grouping several data types of related information together.
  - ii. sizeof( ) operator is not used for finding size of structure.

**(Space for answers)**

.....  
 .....  
 .....

[illegible]

## XVII Exercise

**Attempt Q1 and Q.... from the following (Teacher shall allot the remaining)**

**(Note:** Use Point VIII to X and XIII to XV for all relevant programming exercise use blank pages provided or attach more pages if needed.)

1. Write a program to create a structure for employee having data members like emp\_name, emp\_id, and emp\_salary.
2. Write program for initializing members of structure.

**(Space for answers)**

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

.....

.....

.....

.....

.....

### XVIII References / Suggestions for further Reading

1. <https://www.programiz.com/c-programming/c-structures>
2. <https://www.javatpoint.com/structure-in-c>
3. [https://www.tutorialspoint.com/cprogramming/c\\_structures.htm](https://www.tutorialspoint.com/cprogramming/c_structures.htm)

### XIX Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

### List of Student Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	



## Practical No. 24: Implement Operations Carried on Structure Using C.

### Problem Statement:

Write a program to create a structure for five students having data members like Roll No., Name, Class, and marks in three subjects and calculate the % of marks.

### I. Practical Significance

Structure in C is a collection of different data types which are grouped together and each element in a C structure is called member of structure, if we need to store multiple elements then Array of Structure can be used.

### II. Relevant Program Outcomes

1. **Discipline knowledge:** Apply Computer Programming knowledge to solve broad-based Electronics related problems.
2. **Experiments and practice:** Plan to perform experiments and practices to use the results to solve broad-based Electronics related problems.
3. **Engineering tools:** Apply relevant Computer programming / electronic technologies and tools with an understanding of the limitations.
4. **Communication:** Communicate effectively in oral and written form.
5. **Life-long learning:** Engage in independent and life-long learning activities in the context of technological changes also in the Electronics engineering and allied industry.

### III. Competency and Practical skills

This practical is expect to develop the following skills in you

**Develop 'C' programs to solve broad-based electronic engineering related Write and Save a simple C program for function using pointers.**

1. Use syntax of structure in program.
2. Write and Save a simple C program using editor.
3. Compile (Turbo C) a C Program.
4. Debug and execute the program.

### IV. Relevant Course Outcomes

- Implement concept of structure in 'C'.

### V. Practical Outcome (POs)

Write a program to create a structure for Five students having data members like Roll No., Name, Class, marks in three subjects and calculate the % of marks.

### VI. Relevant Affective domain related Outcome(s)

1. Select proper programming environment in C.
2. Follow safety measures
3. Follow ethical practices.

### VII. Minimum Theoretical Background

Array of structures is nothing but collection of structures. This is also called as structure array in C. To declare an array of structure, you must first define a structure and then declare an array variable of type i.e. to declare 5 elements array of structure of type student can be declared as

**Example:**

```
struct student
{
    int mark;
    char name[10];
    float average;
}s[5];
```

**VIII. Algorithm**

**IX. Flowchart**

## **X. 'C' Program Code**

## **XI. Resources required**

<b>Sr. No.</b>	<b>Name of Resource</b>	<b>Specification</b>	<b>Quantity</b>	<b>Remarks</b>
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For all Experiments
2	Operating system	Windows XP/Windows 7/LINUX version 5.0 or later		
3	Software	Turbo C /C++ Version 3.0 or later		

## **XII. Precautions**

1. Don't forget the semicolon at the end of structure.
2. Strictly follow the instruction for writing, compiling and executing the program.
3. Start and Shutdown system with proper procedure
4. Follow ethical practices.

S. No.	Name of Resource	Specification
1	Computer System with broad specifications	
2	Software	
3	Any other resource used	

.....

.....

.....

.....

.....

.....

.....

**Note:** Below given are few sample questions for reference. Teachers must design more such questions so as to ensure the achievement of identified CO.

1. Difference between array and structure
2. State the true/false for following statement:  
A structure is a convenient way of grouping several pieces of related information together.
3. Create structure for storing specifications of Personal computer as, Microprocessor, RAM, Hard disk size.

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

This image shows a full page of white paper with horizontal dotted lines, typical of primary school handwriting practice paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

**(Space for answers)**

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

.....

.....

.....

### XVIII. References / Suggestions for further Reading

1. <https://www.programiz.com/c-programming/c-structures>
2. <https://www.javatpoint.com/structure-in-c>
3. [https://www.tutorialspoint.com/cprogramming/c\\_structures.htm](https://www.tutorialspoint.com/cprogramming/c_structures.htm)
4. <http://fresh2refresh.com/c-programming/c-structures/>

### XIX. Assessment Scheme

Performance indicators		Weightage (Marks)
<b>Process related(10 Marks)</b>		<b>40%</b>
1.	Debugging ability	25%
2.	Follow ethical practices.	15%
<b>Product related (15 Marks)</b>		<b>60%</b>
3.	Correctness of algorithm	10%
4.	Correctness of Flow chart	10%
5.	Correctness of Program codes	20%
6.	Quality of input/output messaging and output formatting	5%
7.	Timely Submission of report	5%
8.	Answer to sample questions	10%
<b>Total (25 Marks)</b>		<b>100%</b>

### List of Student Team Members

1. ....
2. ....
3. ....
4. ....

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total(25)	





## List Of Laboratory Manuals Developed by MSBTE

### First Semester:

1	Fundamentals of ICT	22001
2	English	22101
3	English Work Book	22101W
4	Basic Science (Chemistry)	22102
5	Basic Science (Physics)	22102

### Second Semester:

1	Bussiness Communication Using Computers	22009
2	Computer Peripherals & Hardware Maintenace	22013
3	Web Page Design with HTML	22014
4	Applied Science (Chemistry)	22202
5	Applied Science (Physics)	22202
6	Applied Machines	22203
7	Basic Surveying	22205
8	Applied Science (Chemistry)	22211
9	Applied Science (Physics)	22211
10	Fundamental of Electrical Engineering	22212
11	Elements of Electronics	22213
12	Elements of Electrical Engineering	22215
13	Basic Electronics	22216
14	'C' programming Language	22218
15	Basic Electronics	22225
16	Programming in "C"	22226
17	Fundamentals of Chemical Engineering	22231

### Third Semester:

1	Applied Multimedia Techniques	22024
2	Advanced Serveying	22301
3	Highway Engineering	22302
4	Mechanics of Structures	22303
5	Building Construction	22304
6	Concrete Technology	22305
7	Strength Of Materials	22306
8	Automobile Engines	22308
9	Automobile Transmission System	22309
10	Mechanical Operations	22313
11	Technology Of Inorganic Chemicals	22314
12	Object Oriented Programming Using C++	22316
13	Data Structure Using 'C'	22317
14	Computer Graphics	22318
15	Database Management System	22319
16	Digital Techniques	22320
17	Principles Of Database	22321
18	Digital Techniques & Microprocessor	22323
19	Electrical Circuits	22324
20	Electrical & Electronic Measurment	22325
21	Fundamental Of Power Electronics	22326
22	Electrical Materials & Wiring Practice	22328
23	Applied Electronics	22329
24	Electrical Circuits & Networks	22330
25	Electronic Measuments & Instrumentation	22333
26	Principles Of Electronics Communication	22334
27	Thermal Engineering	22337
28	Engineering Matrology	22342
29	Mechanical Engineering Materials	22343
30	Theory Of Machines	22344

### Fourth Semester:

1	Hydraulics	22401
2	Geo Technical Engineering	22404
3	Chemical Process Instrumentation & Control	22407
4	Fluid Flow Operation	22409
5	Technology Of Organic Chemicals	22410
6	Java Programming	22412
7	GUI Application Development Using VB.net	22034
8	Microprocessor	22415
9	Database Managment	22416
10	Electric Motors And Transformers	22418
11	Industrial Measurements	22420
12	Digital Electronics And Microcontroller Applications	22421
13	Linear Integrated Circuits	22423
14	Microcontroller & Applications	22426
15	Basic Power Electronics	22427
16	Digital Communication Systems	22428
17	Mechanical Engineering Measuments	22443
18	Fluid Mechanics and Machinery	22445

19	Fundamentals Of Mechatronics	22048
20	Guidelines & Assessment Manual for Micro Projects & Industrial Training	22049

### Fifth Semester:

1	Network Management & Administration	17061
2	Solid Modeling	17063
3	CNC Machines	17064
4	Behavioral Science(Hand Book)	17075
5	Behavioral Science (Assignment Book)	17075
6	Windows Programming using VC++	17076
7	Estimation and Costing	17501
8	Public Health Engineering	17503
9	Concrete Technology	17504
10	Design of Steel Structures	17505
11	Switchgear and Protection	17508
12	Microprocessor & Application	17509
13	A.C. Machines	17511
14	Operating System	17512
15	Java Programming	17515
16	System Programming	17517
17	Communication Technology	17519
18	Hydraulic & Pneumatics	17522
19	Advanced Automobile Engines	17523
20	Basic Electrical & Electronics	17524
21	Measurement and Control	17528
22	Power Engineering	17529
23	Metrology & Quality Control	17530
24	Computer Hardware & Networking	17533
25	Microcontroller	17534
26	Digital Communication	17535
27	Control System & PLC	17536
28	Audio Video Engineering	17537
29	Control System	17538
30	Industrial Electronics and applications	17541
31	Heat Transfer Operations	17560
32	Chemical Process Instrumentation & control	17561

### Sixth Semester:

1	Solid Modeling	17063
2	Highway Engineering	17602
3	Contracts & Accounts	17603
4	Design of R.C.C. Structures	17604
5	Industrial Fluid Power	17608
6	Design of Machine Elements	17610
7	Automotive Electrical and Electronic Systems	17617
8	Vehicle Systems Maintenance	17618
9	Software Testing	17624
10	Advanced Java Programming	17625
11	Mobile Computing	17632
12	System Programming	17634
13	Testing & Maintenance of Electrical Equipments	17637
14	Power Electronics	17638
15	Illumination Engineering	17639
16	Power System Operation & Control	17643
17	Environmental Technology	17646
18	Mass Transfer Operation	17648
19	Advanced Communication System	17656
20	Mobile Communication	17657
21	Embedded System	17658
22	Process Control System	17663
23	Industrial Automation	17664
24	Industrial Drives	17667
25	Video Engineering	17668
26	Optical Fiber & Mobile Communication	17669
27	Therapeutic Equipment	17671
28	Intensive Care Equipment	17672
29	Medical Imaging Equipment	17673

### Pharmacy Lab Manual

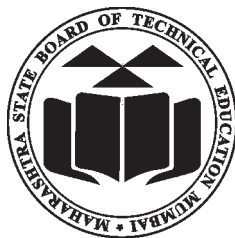
#### First Year:

1	Pharmaceutics - I	0805
2	Pharmaceutical Chemistry - I	0806
3	Pharmacognosy	0807
4	Biochemistry and Clinical Pathology	0808
5	Human Anatomy and Physiology	0809

#### Second Year:

1	Pharmaceutics - II	0811
2	Pharmaceutical Chemistry - II	0812
3	Pharmacology & Toxicology	0813
4	Hospital and Clinical Pharmacy	0816

## HEAD OFFICE



Secretary,

Maharashtra State Board of Technical Education

49, Kherwadi, Bandra (East), Mumbai - 400 051

Maharashtra (INDIA)

Tel: (022)26471255 (5 -lines)

Fax: 022 - 26473980

Email: -secretary@msbte.com

**Web -[www.msbte.org.in](http://www.msbte.org.in)**

## REGIONAL OFFICES:

### MUMBAI

Deputy Secretary (T),

Mumbai Sub-region,

2<sup>nd</sup> Floor, Govt. Polytechnic Building,

49, Kherwadi, Bandra (East)

Mumbai - 400 051

Phone: 022-26473253 / 54

Fax: 022-26478795

Email: rbtemumbai@msbte.com

### PUNE

Deputy Secretary (T),

M.S. Board of Technical Education,

Regional Office,

412-E, Bahirat Patil Chowk,

Shivaji Nagar, Pune

Phone: 020-25656994 / 25660319

Fax: 020-25656994

Email: rbtepn@msbte.com

### NAGPUR

Deputy Secretary (T),

M.S. Board of Technical Education

Regional Office,

Mangalwari Bazar, Sadar, Nagpur - 440 001

Phone: 0712-2564836 / 2562223

Fax: 0712-2560350

Email: rbtepg@msbte.com

### AURANGABAD

Deputy Secretary (T),

M.S. Board of Technical Education,

Regional Office,

Osmanpura, Aurangabad -431 001.

Phone: 0240-2334025 / 2331273

Fax: 0240-2349669

Email: rbteau@msbte.com