SCHEME : K

Name : ______________________________

Roll No. : ______________ Year : 20___ 20___

Exam Seat No. : ______________________

# LABORATORY MANUAL FOR
# PROGRAMMING IN C
# (312303)

## COMPUTER ENGINEERING GROUP

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 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 & 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**

# Programming in C

**(312303)**

**Semester-II**

**( AI/ AN/ BD/ CM/ CO/ CW/ DS/ HA/ IF/ IH )**

**Maharashtra State**

**Board of Technical Education, Mumbai**

**(Autonomous) (ISO 9001:2015) (ISO/IEC 27001:2013)**

## 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 **Programming in C (312303)** 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**

Seal
of
Institutio
n

# 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 'K' 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 'K' scheme laboratory manual development team designed the practical to ***focus*** on the ***outcomes***, rather than the traditional age old practice of conducting practical 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 that 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-cantered 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.

C programming language helps to bring strong foundation for computer programming. This course will help to solve beginner level problem such as mathematical operations, string processing, data structure related processing, with the help of basic concepts, and control flow structures .This Course is basically designed to create a base to develop foundation skills of procedure-oriented programming.

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) are achieved through Practical's of this course.**

Following programme outcomes are expected to be achieved along with programme specific outcomes through the practical's of the course:

**Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve engineering problems.

**Problem analysis:** Identify and analyse well-defined engineering problems using codified standard methods.

**Design/ development of solutions:** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

**Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate techniques to conduct standard tests and measurements.

**Engineering practices for society, sustainability, and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices.

**Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

**Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

**Program Specific Outcomes (PSOs)**

**PSO1Modern Information Technology:** Use latest technologies for operation and application of information.

**PSO2.Information Technology Process**: Maintain the information processes using modern information and communication technologies.

Practical- Course Outcome matrix

Course Outcomes (COs)

a. Develop C program using input-output functions and arithmetic expressions.

b. Develop C program involving branching and looping statements.

c. Implement Arrays and structures using C programs.

d. Develop C program using user defined functions.

e. Write C program using pointer.

<table>
<tr><th>Sr. No .</th><th>Title of the Practical</th><th>CO1</th><th>CO2</th><th>CO3</th><th>CO4</th><th>CO5</th></tr>
<tr><td>1</td><td>*Install and study the C programming Environment.</td><td>√</td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>Implement C programs using Constants and Variables</td><td>√</td><td></td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>* Implement C programs using arithmetic operators to solve given arithmetic operations</td><td>√</td><td></td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>Implement C programs using implicit and Explicit data type conversion</td><td>√</td><td></td><td></td><td></td><td></td></tr>
<tr><td>5</td><td>* Write well commented C programs using formatted Input/Output statements.
<table>
<tr><td>Name</td><td></td><td>FName MName Lname</td></tr>
<tr><td>Roll No</td><td></td><td>XXXX</td></tr>
<tr><td>Percentage</td><td></td><td>(upto 2 decimal places)</td></tr>
<tr><td>Date of Birth</td><td></td><td>DD/MM/YYYY</td></tr>
<tr><td>Branch,College</td><td></td><td>XXXXXXXXXXXXX</td></tr>
</table></td><td>√</td><td></td><td></td><td></td><td></td></tr>
<tr><td>6</td><td>* Implement minimum two C programs using Relational and conditional operator.</td><td>√</td><td></td><td></td><td></td><td></td></tr>
<tr><td>7</td><td>* Implement minimum two C programs using Logical Operators.</td><td>√</td><td></td><td></td><td></td><td></td></tr>
<tr><td>8</td><td>Implement minimum two C programs using Bitwise Operators</td><td>√</td><td></td><td></td><td></td><td></td></tr>
</table>

| | | | | | | |
|---|---|---|---|---|---|---|
| 9 | Implement minimum two C programs using simple If statement and if. Else statement. | | √ | | | |
| 10 | Implement minimum two C programs using nested If .else statement and if.. else if ladder.<br>e.g _Write and execute the C program to print the grade of students based on percentage.<br>Grade: Distinction if per>=75<br>Grade:A if per>=60 and per<75<br>Grade:B if per>=55 and per<60<br>Grade:Pass if per>=40 and per<55<br>Grade:Fail if per<40 | | √ | | | |
| 11 | * Develop C program using Switch statements. | | √ | | | |
| 12 | * Write a C program to print English Calendar months as per given number (eg: If input is 4 then print “April”) using Switch statement. | | √ | | | |
| 13 | * Implement minimum two C programs using 'while' loop and 'do...while' loop statements. | | √ | | | |
| 14 | Implement C programs using for loop statement (e.g.- Write a C Program to print numbers from 1 to 100). | | √ | | | |
| 15 | * Print various patterns using loops. e.g. - Write C Program to print following or similar pattern.<br>*<br>* *<br>* * *<br>* * * * | | √ | | | |
| 16 | * Implement C programs using One Dimensional Array. (e.g.-Write C | | | √ | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | program to input 5 numbers using array and display sum of it). | | | | | |
| 17 | * Implement C programs using Two-Dimensional Array. (e.g.- Write C program to calculate addition of two 3X3 matrices.) | | | √ | | |
| 18 | * Write C program to perform following operations without using standard string functions. i) Calculate Length of given string ii) Print reverse of given string. | | | √ | | |
| 19 | Implement 'Structure' in C (e.g. - Add and Subtract complex numbers using structure). | | | √ | | |
| 20 | * Implement 'Array of Structure' in C (e.g.-Accept and Display 10 Employee information using structure). | | | √ | | |
| 21 | * Develop C program using in-built mathematical and string functions. | | | √ | | |
| 22 | * Write C program to demonstrate User defined Functions. | | | | √ | |
| 23 | Implement recursive functions in C program. | | | | √ | |
| 24 | * Write C Program to print addresses and values of variables using Pointer. (e.g.- Write C program to access and display address of variables.) | | | | | √ |
| 25 | * Implement C Programs to perform arithmetic operations using Pointer. | | | | | √ |

## Industry / Employer Expected Outcome

The aim of this course is to help the student to attain the student to attain the following industry identified outcomes through various teaching learning experiences:

‘Develop ‘C’ programs that address issues with processing strings, mathematic operations ’

Practical No. 1 Install and study the C programming environment.

## I. Practical Significance

This practical is useful for students to understand procedure of Installation of C programming Environment.

II. Industry /Employer Expected Outcome

This practical is expected to develop the following skills as:

Installation of C programming Environment.

III. Course Level Learning Outcome

Installation of C programming language compilers.

IV. Laboratory Learning Outcome

Write logical steps for given program flow.
Write the standard English like statements for programming flow of given problem statement.

V. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

VI. Minimum Theoretical Background

Concept of software download procedure.

VII. Exercise

**1.** Install the C compiler on the PC and write the steps.
**2.** Draw the notations of flowchart.
**3.** Write an algorithm and draw flowchart for the C program to read two numbers and print those two numbers.
**4.** Write a program in C for Q.3.

## VIII. Algortithm

## IX. Flow chart

## X. 'C' Program Code

Write 'C' Program Code for above exercise on the blank pages attached at the end of practical.

XI. Required Resources:

| Sr. No. | Name of Resource | Major Specifications | Qty. | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | | |
| 2. | 'C' Compiler | Turbo C/gcc | | |

## XII. Precautions

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and Shutdown system with proper procedure.
4. Don't forget to include a header file related to functions.

## XIII. Conclusion (Action/decision to be taken based on the interpretation of the results)

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

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

## XIV. Practical Related Questions

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

1. What is compiler and debugger.
2. State any 6 features of C programming language.
3. Define algorithm.
4. Define Flowchart.
5. Write an algorithm and draw the flowchart for printing “Hello World “.

[**Space for Answer**]

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

## XV. References for further Reading

1. https://www.w3schools.in/c-programming/install
2. https://www.guru99.com/c-gcc-install.html

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved(LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| Marks obtained | | | Dated Sign of Teacher |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No 2: Implement C programs using Constants and Variables.

### I. Practical Significance

This practical is useful for students to understand procedure for writing, compiling, and executing simple C programs. After the completion of this practical students will be

### II. Industry Outcomes

The aim of this course is to help students to attain the following industry identified outcomes through various teaching and learning experiences:
Develop C programs that address issues with processing strings, mathematic operations, and data structures.

### III. Course Level Learning Outcomes

Develop C program using input - output functions and arithmetic expressions.

### IV. Laboratory Learning Outcomes

Write Simple C program using constant and
variables. Use the arithmetic operators for developing
C Program.

### V. Relevant Affective domain related Outcomes(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

### VI. Relevant Theoretical Background

Concept of flowchart and algorithm, Symbols of flowchart guidelines for preparing algorithm and flowchart, Concept of variable and constant, knowledge of arithmetic operator

### VII. Exercise

1. Ramesh's basic salary is input through the keyboard. His dearness allowance is 40% of basic salary, and house rent allowance is 20% of basic salary. Write a program to calculate his gross salary.
2. If the marks obtained by a student in five different subjects are input through the keyboard, find out the aggregate marks and percentage marks obtained by the student. Assume that the maximum marks that can be obtained by a student in each subject is 100.

## VIII. Algorithm

## IX. Flowchart

## X. ‘C’ Program Code

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

XI. Required Resources

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |

| | | | | |
|---|---|---|---|---|
| 2. | 'c' Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

**XII. Precautions to be followed.**

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.
4. Don't forget to include a header file related to particular functions.

XIII. Conclusion

………………………………………………………………………………
………………………………………………………………………………

XIV. Practical Related Questions

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

1. The length & breadth of a rectangle and radius of a circle are input through the keyboard. Write a program to calculate the area & perimeter of the rectangle, and the area & circumference of the circle.
2. Two numbers are input through the keyboard into two locations C and D. Write a program to interchange the contents of C and D.

[**<u>Space for Answer</u>**]

………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………

## XV. References / Suggestions for further Reading Software/Learning Websites

a. https://www.geeksforgeeks.org/constants-vs-variables-in-c-language/
b. https://www.javatpoint.com/variables-vs-constants-in-c

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No. 3: Implement C programs using arithmetic operators to solve given arithmetic operations.

### I. Practical Significance

This practical is useful for students to understand arithmetic expressions. Students will be able to use arithmetic expressions in the C program.

### II. Industry /Employer Expected Outcome

This practical is expected to develop the following skills as:

1. Develop C program using input-output functions.
2. Develop C program using arithmetic expressions.

### III. Course Level Learning Outcome:

Develop C program using input - output functions and arithmetic expressions.

### IV. Laboratory Learning Outcome:

Use Arithmetic operators in C Program

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

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

### VI. Minimum Theoretical Background

Concept of algorithms, concept of variables and constants, knowledge of arithmetic operators.

**Arithmetic Operators** are the type of operators in C that are used to perform mathematical operations in a C program.

They can be used in programs to define expressions and mathematical formulas.

What are C Arithmetic Operators?

The C arithmetic operators are the symbols that are used to perform mathematical operations on operands. There are a total of 9 arithmetic operators in C to provide the basic arithmetic operations such as addition, subtraction, multiplication, etc.

Types of Arithmetic Operators in C

The C Arithmetic Operators are of two types based on the number of operands they Work.

These are as follows:

- **Binary Arithmetic Operators**
- **Unary Arithmetic Operators**

### VII. Exercise

**Write a program in C for:**

1.Celsius to Fahrenheit conversion.
2.Area calculation of rectangle
3.Area calculation of circle with PI value constant and #define function.

## VIII. Algorithm

## IX. Flow Chart

## X. ‘C’ Program Code

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

## XI. Required Resources:

| Sr. No. | Name of Resource | Major Specifications | Qty . | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | | |
| 2. | ‘C’ Compiler | Turbo C/gcc | | |

## XII. Precautions

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and Shutdown system with proper procedure.
4. Don't forget to include header file related to functions.

## XIII. Conclusion (Action/ decision to be taken based on the interpretation of the results)

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

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

XIV. Practical Related Questions

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

I. Write an error message given by C compiler during program compilation if you use %d to read float variable.

II. Write a program to print square of given number.

III. Evaluate the following expressions and show their

hierarchy.

G=big/2+big*4/big -big + abc / 3;

( abc =2.5, big=2, assume G to be an float)

IV. Convert the following mathematical formula into appropriate C statements.

X=-b (b*b)24ac/2a

SPACE FOR ANSWERS

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## XV. References for further Reading

a. https://www.w3schools.in/c-tutorial/operators/

b. https://spoken-tu torial.org/ tutorial-search/?search_for=C++ and +Cpp search language =English

c. https://www.geeksforgeeks.org/arithmetic-operators-in-c/

XVI. Assessment Scheme

| **Performance indicators** | | **Weightage** |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (10)** | **Product Related (35)** | **Total (50)** | |
| | | | |

Practical No 4: Implement C programs using implicit and Explicit data type conversion.

## I. Practical Significance

This practical is useful for students to understand the concept of typecasting. After completion of this practical students can convert data into one type to another.

II. Industry /Employer Expected Outcome

This practical is expected to develop C program using data type conversion.

III. Course Level Learning Outcomes

Develop C program using input - output functions and arithmetic expressions.

## IV. Laboratory Learning Outcomes

Write code for type casting in C.

V. Relevant affective domain relative outcomes

1. Follow Safety practices.
2. Practice optimal way to solve problem.
3. Follow ethical practices.

VI. Relevant Theoretical

**Implicit Conversion**

The process of assigning a smaller type to a larger one is known as implicit conversion.

**Byte** → **short** → **int** → **long** → **float** → **double**

Explicit Conversion

- The process of assigning a larger type into a smaller one is called Explicit Conversion.
- Casting into a smaller type may result in loss of data.

  **double** → **long** → **int** → **short** → **byte**

VII. Exercise

1.Develop program to show the use of implicit typecasting.

2. Develop program to show the use of explicit typecasting.

## VIII. Algorithm

## IX. Flowchart

## X. ‘C’ Program Code

Write „C" Program Code for above exercise on the blank pages attached at the end of practical.

## XI. Required Resources

| **Sr. No .** | **Name of Resource** | **Major Specification** | **Qty** | **Remarks** |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | “c" Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions to be followed.

1. Handle computer system with care.
2. Be cautious while performing files related operations in computer system. Closely observe and remember the file name and its folder.

## XIII. Conclusion

……………………………………………………………………………
……………………………………………………………………………

## XIV. Practical Related Questions

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

**1. Write the output of the following program.**

```
# include
  <stdio.h>
  Void main()
{ int m=4;
   float
  n=m+4.7;
  printf(“%f
  \n”,n);
  printf(“%d \n”,n);
 }
```

2. Write the output of the following program.

```
# include <stdio.h>
     Void main()
 { float
    m=4.5;
    Int
    n=m+4;
   printf("%f \n",m);
   printf("%d \n",n);

  }
```

[**Space for Answer**]

XV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.programiz.com/c-programming/type-conversion
2. https://www.geeksforgeeks.org/type-conversion-c/

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No.5 Write well commented C programs using formatted Input/ Output statements.

### I. Practical Significance

This practical is useful for students to use different format specifiers for different data types. After the completion of this practical student will be able to use formatted input output statements in program.

### II. Industry /Employer Expected Outcome

This practical is expected to develop the following skills for the industry identified Competency skills such as:

1. Develop ‘C’ programs using formatted input output statements.
2. Develop ‘C programs and introduce comments into them.

### III. Course Level Learning Outcomes

Develop a C program using input-output functions and arithmetic expressions.

### IV. Laboratory Learning Outcome

Write C code for displaying formatted output with comments wherever applicable.

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

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

### VI. Relevant Theoretical Background

Concept of different data types and their format specifiers.

### VII. Exercise

Write a program in ‘c’ to develop a code for following formatted structure:

| | | |
|---|---|---|
| Name | | FName MName Lname |
| Roll No | | XXXX |
| Percentage | | (upto 2 decimal places) |
| Date of Birth | | DD/MM/YYYY |
| Branch,College | | XXXXXXXXXXXXX |

## VIII. Algorithm

## IX. Flowchart

## X. ‘C’ Program Code

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

**XI.** Resources required.

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | ‘c’ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.

## XIII. Conclusion (Action/decision to be based on the interpretation of the result)

…………………………………………………………………………………
…………………………………………………………………………………

## XIV. Practical related questions

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

1. State significance of format specifier.
2. Write different data types and their format specifiers.
3. What is the purpose of "%d" & "% ld"? How are they used?
4. Write a C program to read integer, character, float, double values from the user and display it. Mention different format specifiers used in program in comment.
5. Write a C program to print the address of variable. mention format specifier used in comment.

[**Space for Answer**]

…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………
…………………………………………………………………………………

## XV. References for further reading and suggestions

1. https://www.w3schools.in/c-tutorial/operators/

2. https://spoken-tutorial.org/tutorial-search/?search_foss=C+and+Cpp&search language—English

## XVI. Assessment Scheme

| **Performance indicators** | | **Weightage** |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No 6: Implement minimum two C program using relational and conditional. Operator

### I. Practical Significance

Students will be able to use relational and conditional operators in the C program.

### II. Industry Outcomes

The aim of this course is to help students to attain the following industry identified outcomes through various teaching and learning experiences:
Develop C programs that address issues with mathematic operations.

### III. Course Level Learning Outcomes

Develop C program using input - output functions and arithmetic expressions.

### IV. Laboratory Learning Outcomes

1. Use Relational and logical operators in C to solve the given problem.
2. Write C program using Relational and logical operators for solving given problem.

### V. Relevant Affective domain related Outcomes(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

### VI. Relevant Theoretical Background

Concept of variable and constant, knowledge of relational operator and conditional operator. **Relational Operator:** When comparison of two quantities is performed depending on their relation, certain decisions are made. C supports six relational operators as shown in table.

| Operator | Meaning |
|---|---|
| < | Less than |
| > | Greater than |
| <= | Less than or equal to |
| >= | Greater than or equal to |
| == | Equal to |
| != | Not equal to |

The **Conditional Operator** (?:) is an operator that takes three arguments. The first argument is a comparison argument, the second is the result upon a true comparison, and the third is the result upon a false comparison. If it helps you can think of the operator as a shortened way of writing an if-else statement. It is often used to assign variables based on the result of a comparison. When used correctly it can help increase the readability and reduce the number of lines in your code.

Syntax: expression1? expression2: expression3

- Expression1 can be any expression that evaluates to a Boolean value.
- If expression1 is true, then expression2 is evaluated; otherwise, expression3 is evaluated.
- The result of the? Operation is that of the expression evaluated.
- Both expression2 and expression3 are required to return the same type, which can't be void.

## VII. Exercise

1.Develop a program to print two numbers are equal or not.
2. Develop a program to print the largest of two numbers using conditional operator.

## VIII. Algorithm

## IX. Flowchart

## X. ‘C’ Program Code

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

## XI. Required Resources

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | ‘c’ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions to be followed.

1. Handle computer system with care.
2. Be cautious while performing files related operations in computer system.
3. Closely observe and remember the file name and its folder.

## XIII. Conclusion

………………………………………………………………………………
………………………………………………………………………………

## XIV. Practical Related Questions

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

1. Write a program for the largest of three numbers.
2. Write a program to demonstrate the working of relational operator.
3. Write a program using conditional operator to check whether a user can vote or not based on their age.

[**Space for Answer**]

………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………
………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

## XV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.javatpoint.com/relational-operator-in-c
2. https://www.geeksforgeeks.org/conditional-or-ternary-operator-in-c/

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No. 7 Implement minimum two C programs using Logical Operators.

### I. Practical Significance

Students will be able to understand logical operators.
After completion of practical student will be able to use logical operators in C program.

### II. Industry /Employer Expected Outcome

This practical is expected to develop the following skills as:
Develop ‘C’ programs that address issues with processing of Logical operators in C Language.

### III. Course Level Learning Outcome:

Develop C program using input - output functions and arithmetic expressions.

### IV. Laboratory Learning Outcome:

Use logical operators in given expressions.

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

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

### VI. Minimum Theoretical

Background Concepts of
Logical Operators

Logical operators in C are used to combining multiple conditions/constraints. Logical Operators returns either 0 or 1, it depends on whether the expression result is true or false. In C programming for decision-making, we use logical operators.
We have 3 logical operators in the C language:

Logical AND : &&
Logical OR : ||
Logical NOT : !

### VII. Exercise

**Write a C program to:**

1.Identify whether given number is Positive even, Positive odd, Negative even or Negative odd.

2. Find the greatest/smallest number between the given 3 numbers.

## VIII. Algorithm

## IX. Flow Chart

## X. ‘C' program code

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

XI. Required Resources:

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |

| | | | | |
|---|---|---|---|---|
| 2. | ‘c’ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.
4. Don’t forget to include a header file related to particular functions.

## XIII. Conclusion

………………………………………………………………………………

………………………………………………………………………………

## XIV. Practical Related Questions

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

1. Write a program to find the greatest number between three by using logical operators.
2. What are different logical operators?
3. Write a program that accepts marks of 5 subjects calculate the percentage and display grade of student from percentage. (Grade Distinction, first class, second class, pass class, fail).
4. Write a program to find whether the given number is even and divisible by 5 or not.

**[<u>Space for Answer</u>]**

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

………………………………………………………………………………

## XV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.w3schools.in/c-tutorial/operators/
2. https://spoken-tu torial.org/ tutorial-search/?search_for=C++ and +Cpp search_language =English
3. https://www.geeksforgeeks.org/logical-operators-in-c/

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No 8: Implement minimum two C programs using Bitwise Operators.

### I. Practical Significance

Students will be able to understand Bitwise operators. After completion of practical students will be able to use Bitwise operator in C program.

### II. Industry Outcomes

The aim of this course is to help students to attain the following industry identified outcomes through various teaching and learning experiences:
Develop C programs that address issues with mathematic operations.

### III. Course Level Learning Outcomes

1.Develop C program using input - output functions and arithmetic expressions.

### IV. Laboratory Learning Outcomes

1. Write expressions using bitwise operators in the given problem statement.

### V. Relevant Affective domain related Outcomes(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.
4. Practice optimal way to solve problem.

### VI. Relevant Theoretical Background

**Bitwise operators** for manipulation of data at values of bit level. These operators are used for testing the bits or shifting bits right or left in the operations.

**1) Bitwise NOT (~):** called bitwise complement, the unary NOT operator, inverts all the bits of its operand.

```
Example: ~ 0111 (decimal 7)
         = 1000 (decimal 8)
```

**2) Bitwise AND (&):** the AND operator & produce a 1 bit if both operands are also 1, A zero is produced in all the cases.

```
e.g   0101 (decimal 5)
  &   0011 (decimal 3)
  =   0001 (decimal 1)
```

**3) Bitwise OR( | ) :** the OR operator, | , combines bits such that if either of the bits in the operand is a 1, then the resultant bit is a 1.

```
e.g   0101 (decimal 5)
 |    0011 (decimal 3)
  =  0111 (decimal 7)
```

**4) Bitwise XOR( ^ ) :** the XOR operator , ^ , combines bits such that if exactly one operand is 1,then the result is 1. Otherwise, the result is zero.

```
e.g  0101 (decimal 5)
^      0011 (decimal 3)
=      0110 (decimal 6)
```

**5) The Left Shift (<<):** the left shift operator, <<, shifts all of the bits in a value ‘to the left a  specified number of times specified by num.

General form: value <<num

```
  e.g. x << 2 (x=12) 0000
   1100 << 2
= 0011 0000 (decimal 48)
```

**6) The Right Shift (>>):** the right shift operator, >>, shifts all of the bits in a value ‘to the right a specified number of times specified by num.

General form: value >>num

e.g. x>> 2 (x=32) 0010
0000 >> 2
= 0000 1000 (decimal 8)

## VII. Exercise

1.Develop a program to demonstrate the use of Bitwise operator.

## VIII. Algorithm

## IX. Flowchart

## X. ‘C’ Program Code

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

## XI. Required Resources

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | ‘c’ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions to be followed.

1. Handle computer system with care.
2. Be cautious while performing files related operations in computer system.
3. Closely observe and remember the file name and its folder.

## XIII. Conclusion

……………………………………………………………………………………
……………………………………………………………………………………

## XIV. Practical Related Questions

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

1. Write a output of the following program main( )

```
 {
 int i = 32, j = 65, k, l, m, n, o,
p ; k = i | 35 ; l = ~k ; m = i &
j ;
 n = j ^ 32 ; o = j << 2 ; p = i >> 5 ;
printf ( "\nk = %d l = %d m = %d", k, l, m ) ;
printf ( "\nn = %d o = %d p = %d", n, o, p ) ;
```

}

2. Explain left shift and right shift operator with example.

[**Space for Answer**]

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

XV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.programiz.com/c-programming/bitwise-operators
2. https://www.javatpoint.com/bitwise-operator-in-c

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No.9 Implement minimum two C programs using simple If statement and if..else statement.

### I. Practical Significance

The ability to control the normal flow of a program to make decisions on what will be executed is valuable to the programmer. One of the important functions of a statement is that it allows you to control if a program entering a section of code based on whether a given condition is true or false. After the completion of this practical, students will be able to use decision making, conditional and branching statements to solve the given problem.

**II.** Industry /Employer Expected Outcome

This practical is expected to develop the following skills as:
Develop 'C' programs that address control and decision-making statements.

### III. Course Level Learning Outcome:

Develop C program involving branching and looping statements.

IV. Laboratory Learning Outcome:

1. Write the syntax for various statements.
2. Write C program for any problem using If statements.
3. Write syntax of if. else statements.

V. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

VI. Minimum Theoretical Background

Concept of variable, constant, keywords and data types.
Concept of relational and logical operators, if statement, if-else statement, nested if-else, if- else ladder.

VII. Exercise

Write a program in C to:

1.Find Whether given number is even or odd.
2.Find Whether given number is prime number or not.

## VIII. Algorithm

## IX. Flow Chart

## X. 'C' Program code

Write 'C' Program Code for above exercise on the blank pages attached at the end of practical.

XI. Required Resources :

| Sr. No. | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | 'c' Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling and executing the program.
3. Start and shutdown system with proper procedure.
4. Don’t forget to include header file related to particular functions

XIII. Conclusion

……………………………………………………………………………………
……………………………………………………………………………………

XIV. Practical Related Questions

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

1. Point out the errors, if any, in the following program. #include<stdio.h>

```
int main()
Int        x=30,
y=40;
printf("x      is
equal   to   y");
elseif(x>y)
printf("x    is    greater
 than y"); elseif(x<y)
printf("y is greater than x");

}

Return (0);

}
```

2. Write a C program to determine whether a given year is a leap year or not. .

3.Write a C program to determine whether a string is palindrome.

[**Space for Answer**]

……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………
……………………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

XV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.w3schools.com/c/c_conditions.php
2. https://www.geeksforgeeks.org/c-if-else-statement/

XVI. Assessment Scheme

| **Performance indicators** | | **Weightage** |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved(LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related 10** | **Product Related 35** | **Total 50** | |
| | | | |

## Practical No 10: Implement minimum two C programs using nested If ..else statement and if ..else if ladder.

**I.** Practical Significance

The ability to control the normal flow of a program to make decisions on what code to be executed is valuable to the programmer. One of the important functions of the nested if - else statement and if-else-if ladder, is that it allows you to control if a program enters a section of code or not based on whether a given condition is true or false.
After completion of this practical students will be able to use decision-making conditions and branching statements to solve the given problem.

**II.** Industry Outcomes

The aim of this course is to help students to attain the following industry identified outcomes through various teaching and learning experiences:
Develop C programs that will use decision making and branching statements to solve the given problem.

**III.** Course Level Learning Outcomes

1.Develop C program using input - output functions and arithmetic expressions.
2.Develop C program involving branching and looping statements.

**IV.** Laboratory Learning Outcomes

1. Write syntax of if.. else statements.

**V.** Relevant Affective domain related Outcomes(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.
4. Practice optimal way to solve problem.

**VI.** Relevant Theoretical Background

**If else statement** is used for decision making. It is used to control flow of execution of statement.

Syntax: -

```
            if(test expression)

            {

            True block statement
            }

            else

            {
```

If the test expression is true then the true block statement, immediately following the if, statements are executed. If the test expression is false then the false block statement, immediately following the else, statements are executed.

Nested if else statement:

When a series of decisions are involved, we may have to use more than one if…else statement in nested form as shown below:

Syntax:

```
if(test condition1)

        {

        if(test condition2)

          {

        statement-1;
          }

        else

          {

        statement-2;
            }

        }

else
```

**VII. Exercise**

1.Write and Execute the C program to print the grades of students based on percentage. Grade: Distinction If per>=75

Grade: A   If per>=60 and Per<75  
Grade: B   If per>=55 and Per<60  
Grade: Pass     If per>=4  
0 and Per<55  
Grade: Fail      if per<40.

2. A store gives discounts based on membership status and the amount spent. Members spending over Rs.10000 get a 10% discount, non-members spending over Rs.15000 get a 5% discount, and all others get no discount.

## VIII. Algorithm

## IX. Flowchart

**X.** **‘C’ Program Code**

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

**XI.** Required Resources

| **Sr. No .** | **Name of Resource** | **Major Specification** | **Qty** | **Remarks** |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | ‘c’ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

**XII.** **Precautions to be followed.**

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.

**XIII.** Conclusion

…………………………………………………………………………………
…………………………………………………………………………………

**XIV.** Practical Related Questions

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

1. A bank introduced an intensive policy of giving bonus to all account holders.

The policy is as follows: A bonus of 2% of the balance is given to everyone. irrespective of their balance and 5% to women if their balance is more than 5000. Enter the balance, gender, and calculate the bonus.

2. An Electric power distribution company charges are as follows.
1. for 0-200 units -- rs. 0.5/unit
2. for 201-400 units-- rs 100+0.65/unit excess of 200units
3. for 401-600 units-- rs 230+0.80/unit excess of 400units
4. for 601 and above units-- rs 390+1.00/unit excess of 600units

Input customer number and power units consumed. Print amount to be paid.

[**<u>Space for Answer</u>**]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**XV.** References / Suggestions for further Reading Software/Learning Websites

**1.** https://www.w3schools.com/c/c_conditions.php
2. https://www.geeksforgeeks.org/c-if-else-if-ladder/

## XVI . Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| Marks obtained | | | Dated Sign of Teacher |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No. 11 Develop C Program using Switch Statements.

### I. Practical Significance

Student will be able to understand use of switch case over if-else statement and will solve given problem using switch statement.

II. Industry /Employer Expected Outcome

This practical is expected to develop the following skills as:
Develop 'C' programs that address issues of branching statements and looping statements.

III. Course Level Learning Outcome:
Develop C program involving branching and looping statements.

IV. Laboratory Learning Outcome:

Write syntax of Switch statement to solving given problem

V. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

VI. Minimum Theoretical Background

Switch case statement evaluates a given expression and based on the evaluated value (matching a certain condition), it executes the statements associated with it. Basically, it is used to perform different actions based on different conditions(cases).

- Switch case statements follow a selection-control mechanism and allow a value to change control of execution.
- They are a substitute for long if statements that compare a variable to several integral values.
- The switch statement is a multiway branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression.

In C, the switch case statement is used for executing one condition from multiple conditions. It is like an if-else-if ladder.
The switch statement consists of conditional-based cases and a default case.

**Syntax of switch Statement** in C

```
switch(expression
)
{
case value1: statement_1;
         break;
case value2: statement_2;
```

```
   break;
case value_n: statement_n;
           break;

default: default_statement;
}
```

VII. Exercise

1. Write a C program that reads the value in the range of 1 -12 and print the name of that month and next month. Print error for any other value.

**VIII. Algorithm**

## IX. Flow chart.

## X. ‘C' program code

(student shall either write a C program code or attach a printout of the code )

XI. Required Resources:

| Sr. No. | Name of Resource | Major Specifications | Qty . | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | | |
| 2. | ‘C’ Compiler | Turbo C/gcc | | |

## XII. Precautions

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.
4. Don’t forget to include header file related to functions.
5. Syntax for switch cases should be given properly.
6. The case condition must be constant, or some value that may be evaluated at time.

**XIII. Conclusion** (Action/decision to be taken based on the interpretation of the results)

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

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

XIV. Practical Related Questions

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

1. Write output of following
program int main()

```
{
Int ch='a'+'b';
switch(ch)
{
case ‘a’ :
 printf("You have entered a");
case 'b':
printf("\n You entered  b");
case 'A':
printf("You have entered A");
case 'b'+'a':
 printf("You have entered a & b");
}
return 0;
}
```

Write output of following program

int main()

```
{
int check=2; Switch (check)
{
case1:
printf("D.W.Steyen") ;
case 2 :
printf("M.G.Johnson");
case 3:
printf("Mohammad Asif");
default:
printf("M.Muralidaran");
}
return 0;
}
```

2.Point out the error in the following program.

```
int main()
{
 float a=3.5; switch(a)
{
 case 0.5:
printf("The art of C")
case 1.5:
printf("The spirit of C");
 case 2.5:
printf("See through C");
 case 3.5:
printf("Simply C");
}
return 0;
 }
```

4. Write a C program to print day of the week by taking number from 1 to 7 using switch statement.

5. Print student's grade by accepting percent marks using switch statement.

SPACE FOR ANSWERS

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

……………………………………………………………………………………………………………

## XV. References for further Reading

1. https://www.geeksforgeeks.org/c-switch-statement/
2. https://spoken-tu torial.org/ tutorial-search/?search_for=C++ and +Cpp search_language

   =English
3. https://www.prohramiz.com/c-programming/c-switch-case-statement

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| Process related: 30 Marks | | 60% |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| Product related: 20 Marks | | 40% |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | Total 50 Marks | 100% |

| Marks obtained | | | Dated Sign of Teacher |
|---|---|---|---|
| Process Related 10 | Product Related 35 | Total 50 | |
| | | | |

## Practical No 12: Write a C program to print English Calendar months as per given number using switch statement.

.

**I.** Practical Significance

Students will be able to solve logical problems using switch statements and will develop an ability to understand how multiple if -else statements can be replaced by simple switch case.

**II.** Industry Outcomes

The aim of this course is to help students to attain the following industry identified outcomes through various teaching and learning experiences:
Develop C programs that will use switch statements to solve the given problem.

**III.** Course Level Learning Outcomes

1. Develop C program using input - output functions and arithmetic expressions.
2. Develop C program involving branching and looping statements.

**IV.** Laboratory Learning Outcomes

1. Write C program using Switch statement.

**V.** Relevant Affective domain related Outcomes(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.
4. Practice optimal way to solve problem.

**VI.** Relevant Theoretical

Background Switch Case

The switch statement is a multi-way decision that tests whether an expression matches one of a few constant integer values and branches accordingly.
This is another form of the multiway decision.
Instead of writing many if. Else statements, you can use the switch statement.

Syntax: -

```
switch(expression)

{

case value-1:

block-1;

break;

case value-2:

block-2;

break;

........

........ default:

default-block;

break;

}

statement-x;
```

The switch statement selects one of many code blocks to be

executed: This is how it works:

- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.
- The break statement breaks out of the switch block and stops the execution.
- The default statement is optional and specifies some code to run if there is no case match.

**VII.** Exercise

1. Write a C program to print English Calendar months as per given number (eg : If input is 4 then print “April”) using Switch statement.
2. Write a C program to Print days of week by taking number from 1 to 7.

## VIII. Algorithm

## IX. Flowchart

## X. ‘C’ Program Code

Write „C‟ Program Code for above exercise on the blank pages attached at the end of practical.

## XI. Required Resources

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | „c‟ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions to be followed.

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.

## XIII. Conclusion

……………………………………………………………………………………
……………………………………………………………………………………

## XIV. Practical Related Questions

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

1. write a program to check whether the triangle is isosceles, equilateral, scalene or Right-angled triangle.
2. Display the season -summer, winter, rain for given month of year.

Write output of following program:

```
#include<stdio.h
> void main()
{
Int choice=0 switch (choice)
{
Case0;
printf("Mumbai");
Case1;
printf("Pune");
}
getch();
}
```

[**Space for Answer**]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## **XV.** References / Suggestions for further Reading Software/Learning Websites

1. https://www.programiz.com/c-programming/c-switch-case-statement
2. https://www.javatpoint.com/c-switch

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No. 13: Implement C minimum 2 C programs using 'while' loop and 'do—while' loop statements.

### I. Practical Significance

Loops are used in programming to repeat a specific block of code.
Students will be able to write programs using different loop statements.

II. Industry /Employer Expected Outcome

This practical is expected to develop the following skills as:
Develop 'C' programs that address issues with processing of repetitions based on conditions.

III. Course Level Learning Outcome

Develop C program involving branching and looping statements.

IV. Laboratory Learning Outcome

Implement iterative solution to problem using while and do - - while loop

V. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

VI. Relevant Theoretical Background

Concept of decision control statements and iterative statements.
Selection of appropriate control or looping statements for given problem.

VII. Exercise

**Write a C program to:**

1. Find sum of digits of a given number.
2.Generate multiplication table up to 10 for numbers 1 to 5

### VIII. Algorithm

## IX. Flow Chart

## X. ‘C' program code

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

XI. Required Resources :

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | ‘c’ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling and executing the program.
3. Start and shutdown system with proper procedure.
4. Don’t forget to include header file related to particular functions
5. Syntax of 'for loop' should be given properly.
6. Avoid infinite loop execution.
7.

XIII. Conclusion

……………………………………………………………………………………

……………………………………………………………………………………

**XIV.** Practical Related Questions

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

1. Find sum of digits of a given number using do-while loop.
2. Write a program to reverse the number of given number print using while loop.
3. Write a program to check given number is palindrome or not using while loop.

[**Space for Answer**]

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

XIV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.programiz.com/c-programming/c-do-while-loops
2. https://spoken-tutorial.org/ tutorial-search/?search_for=C++ and +Cpp search_language =English
3. https://www.geeksforgeeks.org/c-do-while-loop/

XV. Assessment Scheme

| **Performance indicators** | | **Weightage** |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No 14: Implement C programs using for loop statement.

I. Practical Significance

Loops are used in programming to repeat a specific block of code. students will be able to write programs using different loop statements.

II. Industry Outcomes

The aim of this course is to help students to attain the following industry identified outcomes through various teaching and learning experiences:
Develop C programs that will be used for loop statement to solve the given problem.

III. Course Level Learning Outcomes

1.Develop C program using input - output functions and arithmetic expressions.
2.Develop C program involving branching and looping statements.

IV. Laboratory Learning Outcomes

1. Write the syntax for statement.
2. Write C code for solving given problem using for loop.

V. Relevant Affective domain related Outcomes(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.
4. Practice optimal way to solve problem.

VI. Relevant Theoretical Background

**For loop**

The for loop is the most used loop statement. It is pre -test loop and it is used when the number of iterations of the loop is known before the loop is entered.

Syntax:

```
for (expression 1; expression 2;
expression 3)
{
  // code block to be executed
```

**Expression 1** is executed (one time) before the execution of the code block.

**Expression 2** defines the condition for executing the code block.

**Expression 3** is executed (every time) after the code block has been executed.

VII. Exercise

1. Write a C Program to print odd numbers from 1 to 100.
2. Find Fibonacci series for given number.

## VIII. Algorithm

## IX. Flowchart

**X. ‘C’ Program Code**

Write „C‟ Program Code for above exercise on the blank pages attached at the end of practical.

XI. Required Resources

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | „c‟ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

**XII. Precautions to be followed.**

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.

XIII. Conclusion

**XIV. Practical Related Questions**

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

**1. Write output of the following C program.**

```
Main()
{
    int i=0;
    For ( ;i ; )
                                        {
Printf(“ \n  hi”);
}
```

2. Write output of the following C program.

```
Int main()
```

```
{
Int x=100,I;
For(i=0;i<x;i+=3)
{
Printf(“start”); Continue; Printf(“end”);
}
Return 0;
}
```

[**Space for Answer**]

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

## XIV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.w3schools.com/c/c_for_loop.php

2. https://www.programiz.com/c-programming/c-for-loop

## XV. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No.15 : Print various patterns using loops

### I. Practical Significance

Student will be able to understand the nesting of different loops statements. Students will be able to use different nested loop.

II. Industry /Employer Expected Outcome

This practical is expected to develop the following skills as:
Develop program using different loops and the necessity of nesting the loops.

III. Course Level Learning Outcome:

Develop C program involving branching and looping statements.

IV. Laboratory Learning Outcome:

Write syntax for while and do ... while loop Write syntax for 'for' loop

V. Relevant Affective domain related Outcome(s)
Follow safety practices Maintain tools and equipment. Follow ethical practices

VI. Relevant Theoretical Background

Concept of nesting, and different iterative statements, such as for loop, while loop, do-while loop.

VII. Exercise

**Write a C program to :**

1.Write C Program to print following or similar pattern

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

2.Write C Program to print following or similar pattern

```
1
12
123
1234
```

## VIII. Algorithm

## IX. Flow Chart

## X. ‘C' program code

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

XI. Required Resources :

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | ‘c’ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

**XII. Precautions to be followed.**

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling and executing the program.
3. Start and shutdown system with proper procedure.
4. Don’t forget to include a header file related to particular functions.

XIII. Conclusion

……………………………………………………………………………

……………………………………………………………………………

XIV. Practical Related Questions

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

1. Write a program to sort given array in ascending order.
2. Write comments in the above C Program.
3. Write the output of following code also write the comments in the code:

```
void main()

{

int x, y;

for(x=l;x<=5;x++)

{

for(y=1;y<=10;y++)


printf(“/n%”,x*y); getch();

}
```

4. Write a C program to generate the output as follows:

```
4

4 3

4 3 2

4 3 2 1
```

[Space for Answer]

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

## XV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.w3schools.in/c-tutorial/operators/
2. https://study.com/academy/lesson/do-while-loop-definition-example-results.

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |

|  |  |  |  |
|---|---|---|---|
|  |  |  |  |

## Practical No 16: Implement C programs using One Dimensional Array.

### I. Practical Significance

Students will be able to write programs using arrays and store multiple variable values under one variable name with subscript and understand contiguous storage location used in memory for storing these variables.

### II. Industry Outcomes

The aim of this course is to help students to attain the following industry identified outcomes through various teaching and learning experiences:
Develop C programs that will use array to solve the given problem.

### III. Course Level Learning Outcomes

1. Develop C program using input - output functions and arithmetic expressions
2. Develop C program involving branching and looping statement.
3. Implement Arrays and structures using C programs.

### IV. Laboratory Learning Outcomes

Declare and initialize the Array.

Write C program for implementation of one-dimensional array.

### V. Relevant Affective domain related Outcomes(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.
4. Practice optimal way to solve problem.

### VI. Relevant Theoretical Background

Arrays are used to store multiple values in a single variable, instead of declaring Separate variable for each value.

To create an array, define the data type (like int) and specify the name of the array followed by **square brackets []**.

**Syntax: -**
**Declaration of array:**

**Data type array _name [size];**

**Array initialization done by two ways**

1) compile time.
2) Run time.

**<u>At Compile Time :</u>**

Syntax:

Data type arrayname={values of integer};
Example
int num[5]={1,3,4,10,9};
integer array num stores 5 elemnts

**<u>At Run time :</u>**

By using for loop array elements get initialized

Example

```
-      int n, i,arr[i];
       printf("Enter total number of elements(1 to 100): ");
       scanf("%d",&n);
       for(i=0;i<n;++i)  /* Stores number entered by user. */
       {
         printf("Enter Number %d:
          ",i+1);
          scanf("%d”,&arr[i]);
         }
```

**VII.** Exercise

1. Write C program to input 5 numbers using array and display sum of it.
2. Write C program to print array in ascending order.

**VIII. Algorithm**

## IX. Flowchart

## X. ‘C’ Program Code

Write „C‟ Program Code for above exercise on the blank pages attached at the end of practical.

## XI. Required Resources

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | „c‟ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions to be followed.

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.

## XIII. Conclusion

……………………………………………………………………………
……………………………………………………………………………

## XIV. Practical Related Questions

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

1. Check if the following array declaration is correct.

```
int a (25);
int size =10,
b[size];
int c= {0,1,2}
```

2.Point out the error in the following C program.

```
main()
{
int t[3][3]={2,4,3,6,4,5,1,2,3};
Printf("\n %d",t[1][1]);
}
```

[**Space for Answer**]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## XV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.geeksforgeeks.org/c-arrays/
2. https://www.simplilearn.com/tutorials/c-tutorial/array-in-c

## XVI. Assessment Scheme

| **Performance indicators** | | **Weightage** |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No. 17 Implement C programs using Two-Dimensional Array. (e.g.-Write C program to calculate addition of two 3X3 matrices.)

### I. Practical Significance

Students will be able to write programs using two dimensional arrays and store multiple variable values under one variable name with subscript and understand contiguous storage locations used in memory for storing these variables.

### II. Industry /Employer Expected Outcome

This practical is expected to develop the following skills as :
Develop C program to prepare a matrix.
Develop a C program to perform operations on matrix.

### III. Course Level Learning Outcome

Implement Arrays and structures using C programs.

### IV. Laboratory Learning Outcome:

Declare and initialize the two-dimensional Array.
Write C program for implementation of two-dimensional array.

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

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

### VI. Minimum Theoretical Background

Concept of array variables and their declaration. Concept and single and multi-dimensional array.

### VII. Exercise

1. Write C program to calculate addition of two 3X3 matrices.)
2. Write C program to calculate multiplication of two 3X3 matrices.)

### VIII. Algorithm

## IX. Flow Chart

## X. ‘C’ Program Code

Write C Program Code for above exercise on the blank pages attached at the end of practical.

**XI.** Required Resources:

| Sr. No. | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | “c" Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions

1. Handle computer system with care. '
2. Strictly follow the instructions for writing, compiling and executing the program.
3. Start and Shutdown system with proper procedure.
4. Don't forget to include header file related to functions.
5. Array should be declared and used properly.
6. Array size should be given properly.

## XIII. Conclusion

.......... ………………………………………………………………………………

……………………………………………………………………………………..

## XIV. Practical Related Questions

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

1. Check if the following array declarations are correct.

```
  int a(2)(2);

  int row=3, column =3;

b[row][column];

int c[][]={{0,1,2},{4,5,6}};
```

2. Point out the error in the following C program.

```
int   main()

{

int three[3][]= { 2,4,3,

6,8,2,

2,3,1 };

Printf("\n%d", three[1][1]); } return 0; )
```

3. Write a program in C to calculate the addition of 3x3 matrix.
4. Write a program in C to display data in 4x4 table format.

[**<u>Space for Answer</u>**]

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

………………………………………………………………………………

**XV.** References / Suggestions for further Reading Software/Learning Website

1. https://www.w3schools.in/c-tutorial/operators/
2. https://www.w3schools.com/c/c_arrays.php
3. https://www.programiz.com/c-programming/c-arrays#google_vignette

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (10)** | **Product Related (35)** | **Total (50)** | |
| | | | |

## Practical No 18: Write C program to perform following operations without using standard string functions.

### I. Practical Significance

The C string functions are built-in functions that can be used for various operations and manipulations on strings. These string functions can be used to perform tasks such as string copy, concatenation, comparison, length, etc. Students will be able to write programs using string handling functions.

### II. Industry Outcomes

The aim of this course is to help students to attain the following industry identified outcomes through various teaching and learning experiences:
Develop C programs that address issues with processing strings to solve given problem.

### III. Course Level Learning Outcomes

1. Develop C program using input - output functions and arithmetic expressions.
2. Develop C program involving branching and looping statements.
3. Implement Arrays and structures using C programs.
4. Develop C program using user-defined functions.

### IV. Laboratory Learning Outcomes

1. Declare character array as Strings in C.
2. Write C programs for print string operations without using string handling functions.

### V. Relevant Affective domain related Outcomes(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.
4. Practice optimal way to solve problem.

### VI. Relevant Theoretical Background

**Declaration of string:**

**Char string _name[size].**

e.g

char str[15];
string name is str and we can store maximum 15 characters.

## Initialization of string:

### Initialization of strings at compile time:

In C, string can be initialized in different number of ways.

```
 char c[]="abcd";     OR,
char c[5]="abcd";    OR,
char c[]={'a','b','c','d','\0'};  OR;
char  c[5]={'a','b','c','d','\0'};
```

**String can also be initialized using pointers.**

```
char *c="abcd";
```

**Intialization of string at Run time :**

```
#include <stdio.h>

   int main(){
      char name[20];
      printf("Enter name:
      ");
      scanf("%s",name);

      printf("Your name is
      %s.",name); return 0;

   }
```

**VII. Exercise**

Calculate Length of given string.

Print reverse of given string.

**VIII.Algorithm**

## IX. Flowchart

## X. ‘C’ Program Code

Write „C" Program Code for above exercise on the blank pages attached at the end of practical.

## XI. Required Resources

| Sr. No. | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | „c" Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions to be followed.

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.

## XIII. Conclusion

………………………………………………………………………………………

………………………………………………………………………………………

## XIV. Practical Related Questions

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

1. Write a program to check string is palindrome or not.
2. Write a program to demonstrate all string library functions.

3. Find the error in the following C program:

```
Main()
{
Char str[ ]=”Good
Morning”; If (strstr
(str,”hi”)==0) Printf(“\n
string Found”):
}
```

[**Space for Answer**]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**XV.** References / Suggestions for further Reading Software/Learning Websites

4. https://www.programiz.com/c-programming/string-handling-functions
5. https://www.javatpoint.com/c-string-functions

## XVI. Assessment Scheme

| | **Performance indicators** | **Weightage** |
|---|---|---|
| | **Process related: 30 Marks** | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| | **Product related: 20 Marks** | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No. 19 Implement Structures.

**I. Practical Significance**

Students will be able to write programs using structures and use size of operator typedef, Enumerated Data Type in C.

**II.** Industry /Employer Expected Outcome

This practical is expected to develop the following skills as:

To develop a program in C to implement structures.

**III.** Course Level Learning Outcome

Implement Arrays and structures using C programs.

**IV.** Laboratory Learning Outcome

Declare, define, and access structure variables.

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

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

**VI.** Minimum Theoretical Background

Concept of structures, declaration, and use. Concept of typedef and enumerated Data types

**VII.** Exercise

**Write a program in C for:**

1. Create a structure called a library to hold details of the book viz. accession number, title of the book, author name, price of the book and flag indicating whether book is issued or not. Fetch some sample data and display the same.
2. Develop and execute C program to Add Two distances given in kilometer-meter using structures.

**VIII. Algorithm**

## IX. Flow Chart

## X. ‘C’ Program Code

Write „C" Program Code for above exercise on the blank pages attached at the end of practical.

**XI.** Required Resources

| **Sr. No .** | **Name of Resource** | **Major Specification** | **Qty** | **Remarks** |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | „c" Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions

1. Handle computer system with care. '
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and Shutdown system with proper procedure.
4. Don't forget to include header file related to functions.
5. Don't forget the semicolon; in the ending line after declaration of structure.

**XIII.** Conclusion

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

## XIV. Practical Related Questions

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

1. Write output of the following C Program int main ()

```
 { struct sample
{
int num;
 char ml [50];
 char m2[50];
 }m;
m.num
scanf("%d",&size);
 int arr[size];
 for(i=l;i<=size;i++)
{
scanf("%d",arr[i]); printf("%d",arr[i]);
}
 return 0;
}
```

2. Write a output of the following program. #include<stdio.h>

```
#include<conio.h>
enum year {Jan, Feb, Mar, Apr, May, Jun, Jul,
Aug, Sep, Oct, Nov, Dec };
int main ()
{
inti;
 for( i=Jan; i<= Dec; i++)
printf("%d", i);
return 0;
```

[**Space for Answer**]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## XV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.w3schools.in/c-tutorial/operators/
2. https://www.geeksforgeeks.org/structures-c/

## XVI. Assessment Scheme

| **Performance indicators** | | **Weightage** |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No 20: Implement 'Array of Structure' in C.

### I. Practical Significance

An array of structures in C can be defined as the collection of multiple structures variables where each variable contains information about different entities. The array of structures in C are used to store information about multiple entities of different data types. The array of structures is also known as the collection of structures. Students will be able to write programs using structures.

### II. Industry Outcomes

The aim of this course is to help students to attain the following industry identified outcomes through various teaching and learning experiences:
Develop C programs with array of structures to solve given problem.

### III. Course Level Learning Outcomes

1. Develop C program using input - output functions and arithmetic expressions.
2. Develop C program involving branching and looping statements.
3. Implement Arrays and structures using C programs.

### IV. Laboratory Learning Outcomes

1. Write C programs using Array of Structure.

### V. Relevant Affective domain related Outcomes(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.
4. Practice optimal way to solve problem.

### VI. Relevant Theoretical

Background Declaration of

Array of Structures

**struct** *structure_name array_name* [number_of_elements];

Initialization of Array of Structures

We can initialize the array of structures in the following way:

```
struct structure_name array_name [number_of_elements] = {
            {element1_value1, element1_value2, },
            {element2_value1, element2_value2, },
                                    ......
                                    ......
                        };
```

**VII.** Exercise

1. Accept and Display 10 employees 's information using structure.
2. Accept and Display 10 books information using structure.

## VIII. Algorithm

## IX. Flowchart

**X. ‘C’ Program Code**

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

**XI.** Required Resources

| **Sr. No .** | **Name of Resource** | **Major Specification** | **Qty** | **Remarks** |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | ‘c’ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

**XII. Precautions to be followed.**

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.

**XIII.** Conclusion

………………………………………………………………………………………
………………………………………………………………………………………

**XIV.** Practical Related Questions

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

1.Point out the errors, if any, in the following program struct virus

```
{

char  signature[25] ; char  status[20] ;
int  size ;

} v[2] = {  "Yankee Doodle", "Deadly", 1813,

"Dark Avenger",  "Killer", 1795

} ;

main( )

{

int  i ;

for ( i = 0 ; i <=1 ; i++ )

printf ( "\n%s %s", v.signature, v.status ) ;

}
```

2. A record contains the name of cricketer, his age, number of test matches that he has played and the average runs that he has scored in each test match. Create an array of structure to hold records of 20 such cricketer and then write a program to read these records.

[**Space for Answer**]

## XV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.javatpoint.com/array-of-structures-in-c
2. https://www.geeksforgeeks.org/c-array-of-structure/

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No.21 : Develop C program using in-built mathematical and string functions.

-

I Practical Significance

This practical will be helpful for the students to define and declare user defined functions and library functions and use different string handling functions, mathematical functions as well as miscellaneous functions. Students will be able to develop the ability to compare strings, copy strings, generate random members, sine, cosine and log values.

II Industry /Employer Expected Outcome

This practical is expected to develop the following skills as:

1. Develop C program using input-output functions and arithmetic expressions.
2. Develop C program using user-defined functions.

III Course Level Learning Outcome

Develop C program using user-defined functions.

IV Laboratory Learning Outcome:

Use built-in library functions in C programs.

V Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

VI Minimum Theoretical Background

Concept and need of function.
Concept of library functions like Math and string handling functions and other miscellaneous functions.
Declaration and use of User Defined
functions. Some string related functions are:

- String Copy: strncpy( )
- String Concate: strncat( )
- String Compare: strncmp( )
- First Character Occurrence: strchr ( )
- Last Character Occurrence: strrchr( )
- String Search: strstr( )
- String Token Break: strtok( )
- Lowercase String: strlwr( )

Basic Math Functions in C are those that perform elementary mathematical operations, such as addition, subtraction, multiplication, division, and modulus.
Some of these Basic Math Functions include:

- abs() - returns the absolute value of an integer.
- ceil() - rounds up a floating-point number to the nearest integer.
- floor() - rounds down a floating-point number to the nearest integer

- pow() - raises a number to the power of another number
- sqrt() - calculates the square root of a number

**VII** Exercise

1. Checking palindrome using string function
2. Calculating average using mathematical function

## VIII Algorithm

## IX Flowchart

## X ‘C’ Program code

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

## XI Required Resources:

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | ‘c’ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII Precautions

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.

## XIII. Conclusion

………………………………………………………………………………
………………………………………………………………………………

## XIV. Practical Related Questions

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

1. Write output of the following C Program.

```
#include<stdio.h>
#include<math.h>
int main()
{
int i=l0; printf("%f\n",log10(i));
}
return 0;}
```

2. Point out the errors of the following program.

```
Void main( )
{
```

```
char str[]="Good Morning";
if(strstr(str,"Welcome")==0)
printf("\n  substring  found");}
```

[**Space for Answer**]

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

XV. References for further Reading

1. https://www.w3schools.in/c-tutorial/operators/
2. https://spoken-tu torial.org/ tutorial-search/?search_for=C++ and +Cpp

   search language =English -

XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved(LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related 10** | **Product Related 35** | **Total 50** | |
| | | | |

## Practical No 22: Write C program to demonstrate User defined Functions.

I. Practical Significance

Students will be able to define and declare user define functions and can understand the scope of variables. Students will use parameters using call by value and call by reference in C program.

II. Industry Outcomes

The aim of this course is to help students to attain the following industry identified outcomes through various teaching and learning experiences:
Develop C programs with user define function to solve given problem.

III. Course Level Learning Outcomes

1. Develop C program using input - output functions and arithmetic expressions.
2. Develop C program using user-defined functions.

IV. Laboratory Learning Outcomes

1. Write C programs using user defined functions.

V. Relevant Affective domain related Outcomes(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.
4. Practice optimal way to solve problem.

VI. Relevant Theoretical Background

A **user-defined function** is a type of function in C language that is defined by the user himself to perform some specific tasks. It provides code reusability and modularity to our program. User-defined functions are different from built-in functions as their working is specified by the user and no header file is required for their usage.

C Function Definition

```
    return_type function_name (type1 arg1, type2 arg2..typeN argN)
  {


     // actual statements to be executed
     // return value if any
  }
```

C Function Prototype

A function prototype is also known as a function declaration which specifies the **function's name, function parameters,** and **return type.**

Syntax

return_type **function_name** (type1 *arg1*, type2 *arg2*, ... typeN *argN*);

C Function Call

To transfer control to a user-defined function, we need to call it. Functions are called using their names followed by round brackets. Their arguments are passed inside the brackets.

Syntax

function_name(arg1, arg2, ... argN);

VII. Exercise

1. Find factorial of a given number using function
2. Create a function to find GCD of given number. Call this function in a program.

**VIII. Algorithm**

## IX. Flowchart

## X. ‘C’ Program Code

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

XI. Required Resources

| **Sr. No .** | **Name of Resource** | **Major Specification** | **Qty** | **Remarks** |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | ‘c’ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

## XII. Precautions to be followed.

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.

XIII. Conclusion

…………………………………………………………………………………
…………………………………………………………………………………

XIV. Practical Related Questions

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

1. Write program to reverse the number 1234 using function.
2. What would be the output of the following programs:

**a.**
```
main()

{

printf (“\n Only stupid’s use C?" ) ;
display() ;
}

display()

{

printf ( "\nFools too use C!" ) ; main() ;
}
```

**b**.
```
main ()

{

int i = 45, c ;
c = check(i ) ;
printf (“\n%d", c ) ; check (int ch )
{

if ( ch >= 45 ) return (100 ) ;
else return (10 * 10 ) ;
}
```

[**Space for Answer**]

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

XV. References / Suggestions for further Reading Software/Learning Websites

1. https://www.shiksha.com/online-courses/articles/user-defined-functions-in-c/
2. https://www.geeksforgeeks.org/user-defined-function-in-c/

## XVI. Assessment Scheme

| **Performance indicators** | **Weightage** |
|---|---|
| **Process related: 30 Marks** | **60%** |

| | | |
|---|---|---|
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No. 23: Implement Recursive functions in C program.

### I Practical Significance

In this practical student will be able to use recursive function in C program.

### II Industry /Employer Expected Outcome

This practical is expected to develop the following skills as:

1. Develop C program using Recursive functions.

### III Course Level Learning Outcome:

Develop C program involving branching and looping statements.

### IV Laboratory Learning Outcome:

Write Recursive functions in C.

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

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

### VI. Relevant Theoretical Background

Concept of function, function declaration, function definition, function call. Understanding *of* user defined functions and recursion.

Recursion is the process of a function calling itself repeatedly till the given condition is satisfied. A function that calls itself directly or indirectly is called a recursive function and such kind of function calls are called recursive calls.

In C, recursion is used to solve complex problems by breaking them down into simpler sub-problems. We can solve large numbers of problems using recursion in C. For example, factorial of a number, generating Fibonacci series, generating subsets, etc.

### VII Exercise

1. Write a C Program to generate Fibonacci series using recursion.
2. Write a C Program to find Prime factors of number using recursion.

## VIIl. Algorithm

## IX. Flow Chart

## X. 'C' program code

Write ‘C’ Program Code for above exercise on the blank pages attached at the end of practical.

## XI. Required Resources

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | ‘c’ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

X11. Precautions

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.
4. Function name should start with alphabet & may consist of alpha numeric characters.
5. If there are two words in function name then it should be attached with_ or -
   eg.emp salary() or emp-data()
6. User defined function name should not be same as any built-in function.

## XIII. Conclusion

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

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

XIV. Practical Related Questions

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

1.Write output of the following Program.

```
#include<stdio.h>

int fun(int count)

{

printf("\n %d",count);
if (count<3)
{
fun(fun(fun(++count)));
 return count;
}
 fun(l);
return 0;
}
```

2. Write a program in C to find and print the factorial of 10 numbers using recursive function.

[**Space for Answer**]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

……………………………………………………………………………
……………………………………………………………………………
……………………………………………………………………………
…………………………………………………………………………

XV. References/Suggestions for further Reading Software /Learning Websites

1. https://www.w3schools.com/c/c_functions_recursion.php
2. https://www.programiz.com/c-programming/c-recursion

### XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| Process related: 30 Marks | | 60% |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved(LLO mapped) | 10% |
| Product related: 20 Marks | | 40% |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| Marks obtained | | | Dated Sign of Teacher |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No 24: Write C Program to print addresses and values of variables using Pointer.

### I. Practical Significance

Students will be able to understand the concept of pointer variables and how to access their addresses. Students will be able to declare, initialize and access pointers. After the completion of this practical student will be able to use pointers to get the memory address of variable.

### II. Industry Outcomes

The aim of this course is to help students to attain the following industry identified outcomes through various teaching and learning experiences:
Develop C programs with pointers to solve the given problem.

### III. Course Level Learning Outcomes

1. Develop C program using input - output functions and arithmetic expressions.
2. Write C program using pointer.

### IV. Laboratory Learning Outcomes

1. Declare and initialize pointer variables.

2 Write C program to access variables using pointers.

### V. Relevant Affective domain related Outcomes(s)

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.
4. Practice optimal way to solve problem.

### VI. Relevant Theoretical Background

**Pointers** are variables used to store the memory addresses of the variables. Variables store the values and pointers store their addresses at which these variables are located. The values stored in the pointers are integer values.

Pointer declaration & initialization

Likewise other variables pointer also needs to be declared.

Syntax:-

```
data_type * name_of_variable;
```

Example:-

```
int *ptr ;
```

Here ptr variable of data type integer pointer is declared as name of variable ‟ptr‟ is preceded by * (indirection operator) means that variable ptr is a pointer variable.

Pointers are initialized by address of another variable and for accessing address of a variable “Address of “operator is used.
This operator will fetch address of variable to which it prefix (Precedes).

ptr=&name_of_variable;

## VII. Exercise

1. Write C program to access and display address of variables.

## VIII. Algorithm

## IX. Flowchart

**X. ‘C’ Program Code**

Write „C‟ Program Code for above exercise on the blank pages attached at the end of practical.

XI. Required Resources

| **Sr. No .** | **Name of Resource** | **Major Specification** | **Qty** | **Remarks** |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | „c‟ Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

**XII. Precautions to be followed.**

1. Handle computer system with care.
2. Strictly follow the instructions for writing, compiling, and executing the program.
3. Start and shutdown system with proper procedure.

XIII. Conclusion

…………………………………………………………………………………
…………………………………………………………………………………

XIV. Practical Related Questions

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

1. Write the output of the following

program: void main()

{

int***r,**q,*p,i=5; p=&i;

```
q=&p; r=&q;
```

printf(“%d%d%d”,*p,**q,***r);

}

2. Write the value of d in the following component of

program: int a=20;

int*b=

&a;

int*c=

b;

int d=*b+*c;

[**Space for Answer**]

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

**XV.** References / Suggestions for further Reading Software/Learning Websites

1. https://www.w3schools.com/c/c_pointers.php
2. https://www.programiz.com/c-programming/c-pointers

## XVI. Assessment Scheme

| **Performance indicators** | | **Weightage** |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |

## Practical No. 25: Implement C programs to perform arithmetic operations using pointer.

### I. Practical Significance

Student will be able to understand arithmetic operations that can be performed on pointers. Students will be able to write a c program to perform arithmetic operations on pointers.

### II. Industry /Employer Expected Outcome

This practical is expected to develop the following skills as:

1. Develop C program using input-output functions and arithmetic expressions.
2. Write a program using pointers.

### III. Course Level Learning Outcome:

Write C program using pointer.

### IV. Laboratory Learning Outcome

Perform arithmetic operations using pointers.

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

1. Follow safety practices.
2. Maintain tools and equipment.
3. Follow ethical practices.

### VI. Relevant Theoretical Background

Concept and need of arithmetic operations
Concept of pointers in C.

### VI. Minimum Theoretical Background

Concept and need for functions. Concept of library functions like Math and String functions. Arithmetic Operators are the type of operators in C that are used to perform mathematical operations in a C program. They can be used in programs to define expressions and mathematical formulas.

What are C Arithmetic Operators?

The C arithmetic operators are the symbols that are used to perform mathematical operations on operands. There are a total of 9 arithmetic operators in C to provide the basic arithmetic operations such as addition, subtraction, multiplication, etc.

Types of Arithmetic Operators in C

The C Arithmetic Operators are of two types based on the number of operands they work on.

**These are as follows:**
Binary Arithmetic
Operators Unary
Arithmetic Operators

Pointers

A pointer is defined as a derived data type that can store the address of other C variables or a memory location. We can access and manipulate the data stored in that memory location using pointers.

As the pointers in C store the memory addresses, their size is independent of the type of data they are pointing to. This size of pointers in C only depends on the system architecture.

**Syntax of C Pointers**

The syntax of pointers is similar to the variable declaration in C, but we use the ( * ) dereferencing operator in the pointer declaration.

datatype *
ptr; where

ptr is the name of the pointer.
datatype is the type of data it is pointing to.
The above syntax is used to define a pointer to a variable. We can also define pointers to functions, structures, etc.

VII. Exercise

1.Addition using pointer

2.Subtraction using pointer

3.Multiplication using pointer

4. Division using pointer

## VIII. Algorithm

## IX. Flow chart

## X. 'C' program code

Write „C" Program Code for above exercise on the blank pages attached at the end of practical.

## XI. Required Resources

| Sr. No . | Name of Resource | Major Specification | Qty | Remarks |
|---|---|---|---|---|
| 1. | Computer System | Any desktop or laptop computer with basic configuration | One computer system for each student | |
| 2. | „c" Compiler | Turbo C/gcc/Online compiler | One for each computer system | |

XII. Precautions

1. Handle computer system with care. '-
2. Strictly follow the instructions for writing, compiling and executing the program.
3. Start and Shutdown system with proper procedure.
4. You Should take care to define and declare pointers.

## XIII. Conclusion

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

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

## XIV. Practical Related Questions

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

1. Assume that float takes 4 bytes. Predict the output of the following program. #include<stdio.h>

```
int main ()
{
float arr[S]={12.5,10.0,13.5,90.5,0.5};
float *ptrl =&arr[0];
float *ptr2= ptr+1;
printf("%f\n", *ptr2);
 printf(" %d", ptrl-ptr2);
 return 0;
}
```

[**<u>Space for Answer</u>**]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

XV. References/ suggestions for further Reading Software /Learning Websites

1. https://www.javatpoint.com/c-pointers
2. https://www.geeksforgeeks.org/c-pointers/

-

## XVI. Assessment Scheme

| Performance indicators | | Weightage |
|---|---|---|
| **Process related: 30 Marks** | | **60%** |
| 1. | Debugging ability | 20% |
| 2. | Correctness of Program codes | 30% |
| 3. | Quality of output achieved (LLO mapped) | 10% |
| **Product related: 20 Marks** | | **40%** |
| 1. | Completion and submission of practical in time | 20% |
| 2. | Answer to sample questions | 20% |
| | **Total 50 Marks** | **100%** |

| **Marks obtained** | | | **Dated Sign of Teacher** |
|---|---|---|---|
| **Process Related (30)** | **Product Related (20)** | **Total (50)** | |
| | | | |