**MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE**
Belawadi, Srirangapatna Taluk, Mandya-571477
**CSE(Data Science)**

# "C Programming Basics-Workshop" Report

**Date of Event:** 11/09/2026
**Venue:** SCADE Lab
**Attendees:** 3rd Semester students

**Title of the Workshop:** Pointers and Dynamic Memory Allocation, Looping Concepts, and Function Concepts

**Resource Persons / Presenters:**
1. Dr. Pushpa D
2. Prof. Vijay Kumar, Department of Information Science and Engineering, MITM
3. Prof. Deepthi N

## Introduction

A technical workshop on C programming was organized to strengthen the programming knowledge and practical skills of students. The workshop focused on three important areas of C programming: pointers and dynamic memory allocation, looping concepts, and functions.
These topics are essential for developing a strong foundation in programming, understanding memory management, writing reusable code, and solving computational problems efficiently. The sessions combined conceptual explanations with programming examples and hands-on exercises to help participants understand the practical applications of the concepts.
The sessions were presented by Dr. Pushpa D, Prof. Vijay Kumar from the Department of Information Science and Engineering, MITM, and Prof. Deepthi N.

## Objectives of the Workshop

The major objectives of the workshop were:
- To introduce the fundamental concepts of pointers and their applications in C programming.
- To explain dynamic memory allocation and the efficient utilization of memory.
- To familiarize participants with different looping statements and their applications.
- To develop an understanding of functions and modular programming.
- To improve logical thinking and problem-solving abilities through practical programming exercises.
- To encourage students to write structured, reusable, and efficient C programs.

## Topics Covered During the Workshop

### Pointers in C Programming

The session introduced pointers as an important feature of C programming that allows programmers to work with memory addresses. The presenters explained pointer declaration and initialization, along with the address-of (&) and dereference (*) operators.

The following concepts were discussed:

- Introduction to pointers
- Pointer declaration and initialization
- Address-of and dereference operators
- Pointers and variables
- Pointers and arrays
- Pointers and functions
- Pointer arithmetic
- Pointer to pointer

Suitable examples were used to demonstrate how pointers access and manipulate data stored in memory. The relationship between arrays and pointers, along with the use of pointers in functions, was also explained.

**Dynamic Memory Allocation**

The session on dynamic memory allocation explained how memory can be allocated and managed during program execution. The difference between stack memory and heap memory was discussed to help participants understand memory organization.
The following memory management functions were covered:

- malloc() – allocates a block of memory.
- calloc() – allocates memory and initializes it to zero.
- realloc() – changes the size of previously allocated memory.
- free() – releases dynamically allocated memory.

The presenters explained the importance of allocating sufficient memory, resizing memory when required, and releasing allocated memory after use.
Practical programming exercises included creating dynamic arrays using malloc() and calloc(), resizing arrays using realloc(), freeing allocated memory, and finding the sum and largest element in a dynamically allocated array.
These examples helped participants understand the practical importance of dynamic memory allocation in developing flexible programs.

**Looping Concepts in C**

The session on looping concepts focused on the repeated execution of statements using different looping structures in C. The presenters explained the syntax, working principles, and applications of the following loops:

- for loop
- while loop
- do-while loop

The use of loops in solving repetitive computational problems was demonstrated through suitable examples.
The programming exercises included:

1. Printing numbers from 1 to 10.
2. Displaying even and odd numbers.
3. Finding the sum of the first N natural numbers.
4. Calculating the factorial of a number.
5. Generating multiplication tables.
6. Checking whether a number is prime.
7. Generating the Fibonacci series.
8. Reversing a number.

9. Checking whether a number is a palindrome.
10. Finding the sum of digits.
11. Printing star patterns.
12. Developing a simple menu-driven program.

The exercises provided participants with opportunities to understand loop execution, develop algorithms, and improve their ability to solve programming problems.

**Functions in C Programming**

The session on functions highlighted the importance of dividing a program into smaller, manageable modules. The presenters explained how functions improve code reusability, readability, maintainability, and program organization.

The following topics were covered:

- Introduction, need, and advantages of functions
- Library functions and user-defined functions
- Function declaration or prototype
- Function definition and function calling
- Function arguments and parameters
- Return values
- Types of user-defined functions
- Call by value and call by reference
- Passing arrays to functions
- Nested function calls
- Scope of variables
- Local and global variables
- Static variables in functions
- Functions with pointers
- Practical applications of functions
- Problem-solving using functions

Examples and programming exercises were used to illustrate how functions can be designed and called to perform specific tasks. The importance of passing arguments, returning values, and organizing code into reusable functions was emphasized.

**Teaching and Learning Methodology**

The workshop followed a concept-oriented and practice-based approach. The presenters explained the theoretical concepts and demonstrated their applications through C programs.

The teaching methodology included:

- Explanation of fundamental programming concepts.
- Demonstration of syntax and program structure.
- Discussion of practical examples.
- Hands-on programming exercises.
- Problem-solving activities based on loops, pointers, dynamic memory allocation, and functions.

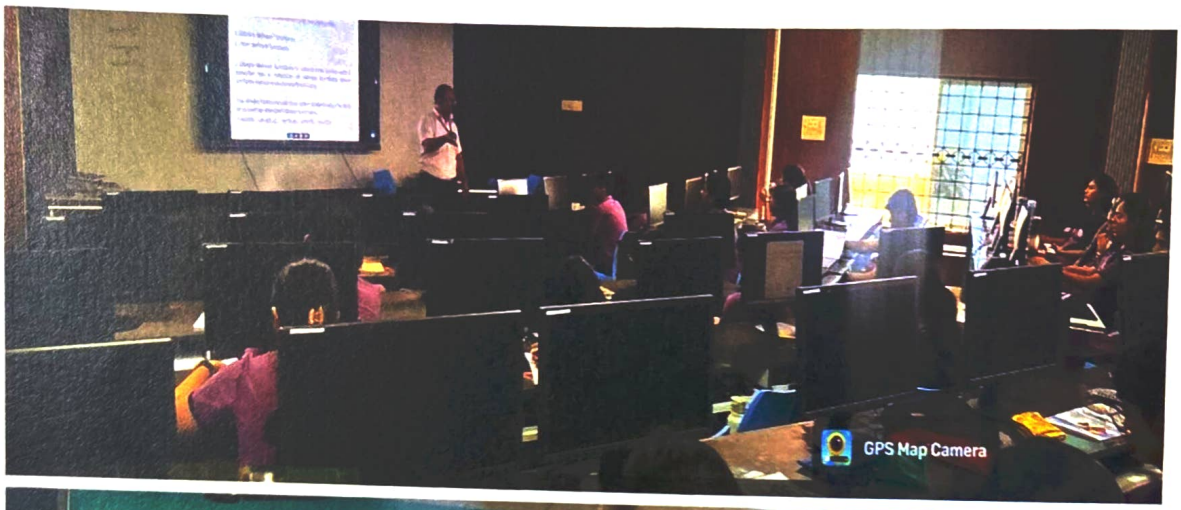
This approach helped participants connect theoretical knowledge with practical implementation.

**Outcomes of the Workshop**

At the end of the workshop, participants were expected to:

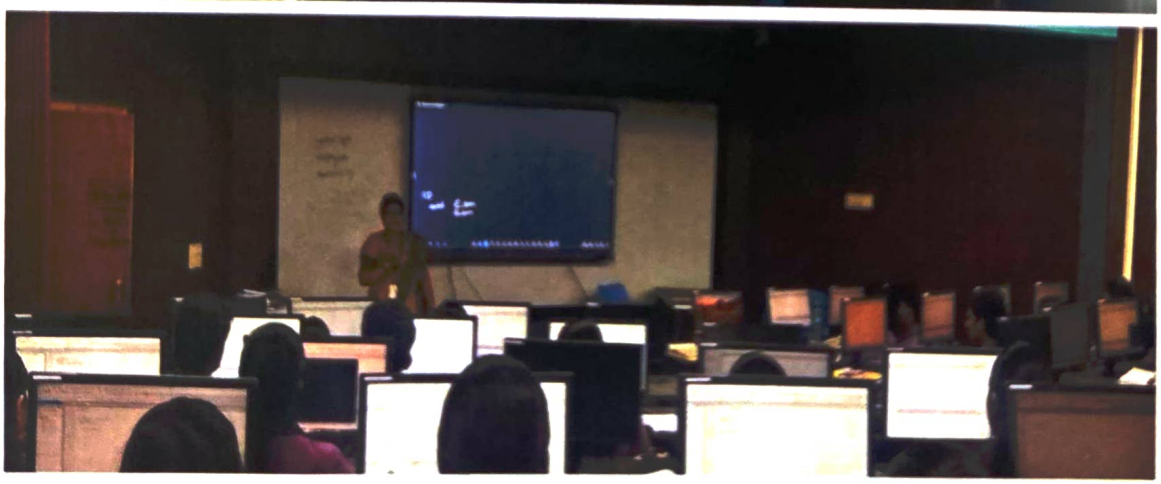
- Understand the declaration, initialization, and use of pointers.

- Apply pointer concepts to arrays and functions.
- Use dynamic memory allocation functions appropriately.
- Understand the differences between malloc(), calloc(), realloc(), and free().
- Implement repetitive tasks using for, while, and do-while loops.
- Develop programs to solve numerical and logical problems.
- Design and implement user-defined functions.
- Improve program modularity and code reusability.
- Strengthen logical reasoning and practical C programming skills.









**Conclusion**

The technical workshop on C programming provided a structured learning opportunity to understand pointers, dynamic memory allocation, looping constructs, and functions. The sessions presented by Dr. Pushpa D, Prof. Vijay Kumar from ISE, MITM, and Prof. Deepthi N covered essential programming concepts through explanations, examples, and practical exercises.

The workshop aimed to strengthen the participants' programming fundamentals and enhance their ability to develop efficient, modular, and logically structured C programs. The knowledge gained through these sessions can support further learning in data structures, algorithms, and advanced programming concepts.

The workshop concluded with an emphasis on continuous practice and the application of programming concepts to real-world problem-solving.

Coordinator's Signature
Prof. Deepthi N

HOD
Dr Pushpa D