

**INSTRUCTIONS TO PAPER SETTERS:**

1. Question No. 1 should be compulsory and cover the entire syllabus. There should be 10 questions of short answer type of 2.5 marks each, having at least 2 questions from each unit.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions to evaluate analytical/technical skills of candidate. However, student may be asked to attempt only 1 question from each unit. Each question should be 12.5 marks including subparts, if any.

**LEARNING OBJECTIVE:**

After covering the core C in about 25 lectures the course shall aim to acquaint the learners about advanced features of the language. The following features are listed as suggested guideline for the teacher.

- Passing by value and pass by reference
- Difference between array names and pointers
- Allocating memory over the heap to two dimensional array ( Matrices application could be taken as a case study)
- Pointer and pointer operations( Linked lists , doubly linked lists circular linked lists can be taken as a case study)
- Pointers to functions and call back functions
- Bitwise operations and a case based upon these operations
- MACROs and their pitfalls
- Final case study could be an application making extensive handling of binary files.

**PRE-REQUISITES:**

• Basic Programming
• <a href="http://bvicam.in/spec-subject-files/ProgramminginC/C">http://bvicam.in/spec-subject-files/ProgramminginC/C</a> Pre Requisite based Study Material.pptx
• MOOC Course Title: Introduction to Programming in C by IIT Kanpur URL: <a href="https://nptel.ac.in/courses/106104128/">https://nptel.ac.in/courses/106104128/</a>

**COURSE OUTCOME (CO):**

After completion of this course, the learners will be able to:-

CO1	Understand the basics of C programming like data type primitives, control structures, arrays, functions & storage classes and apply them for program development. (BTL2)
CO2	Create, apply and extend advanced data types, pointers and dynamic memory allocation functions for the solution of complex problems. (BTL3&6)
CO3	Implement the various C pre-processor directives, standard library functions (system calls using Ubuntu) and file handling techniques.(BTL3)
CO4	Explore and illustrate C programming skills to control and manipulate files,

directories and processes in Ubuntu for building real-time software utilities.(BTL4)
--

### UNIT- I

**Introduction:** GCC, Using MAKE Utility, GDB, **C Basics:** History of C, Characteristics of C, C Program Structure, Variables, Defining Global Variables, Printing Out and Inputting Variables, Constants, Arithmetic Operations, Comparison Operators, Logical Operators, Order of Precedence, Conditionals (The if statement , The ? operator, The switch statement) Looping and Iteration (The for statement, The while statement, The do-while statement, break and continue) Arrays and Strings (Single and Multi-dimensional Arrays, Strings) Functions (Function Prototyping, passing parameters, returning values, recursion) Storage classes (auto, extern, static, register)

[No. of Hrs. : 10 Hrs]

### UNIT-II

**Further Data Types:** Defining New Data Types, Structures, Unions, Type-Casting, Enumerated Types, Low Level Operators and Bit Fields (Bitwise Operators, Bit Fields)

**Pointers:** Pointers arithmetic and Arrays, const pointers, void pointers, near, far and huge pointers

**Dynamic Memory Allocation and Dynamic Structures:** (malloc, calloc and realloc; sizeof, free, introduction to Linked Lists and dynamic 2- dimensional arrays)

**Advanced Pointer Topics:** (Pointers to Pointers, Pointer to array, Array of pointers, Command line input, Pointers to a Function, Implementing Callbacks)

[No. of Hrs. : 12 Hrs]

### UNIT -III

**The C Preprocessor:** (#define, #undef, #include, #if -- Conditional inclusion, Other Preprocessor Commands) **C, Linux and Standard Libraries:** (Advantages of using Linux with C, Using Linux System Calls and Library Functions) Integer Functions, Random Number, String Conversion, Searching and Sorting: <stdlib.h> Mathematics: <math.h> (Math Functions, Math Constants), Input and Output (I/O):stdio.h Reporting Errors (perror(), errno, exit() ) Streams (Predefined Streams, Redirection) Basic I/O (Formatted I/O, printf, scanf ), String Handling: <string.h> (Basic String Handling Functions and safety issues, String Searching), Character conversions and testing: ctype.h, **Files** Character and Line Based I/O, Formatted I/O, Block I/O, File Positioning, Status Functions, Deletion and Renaming, Temporary Files

[No. of Hrs. : 11 Hrs]

### UNIT -IV

**File Accessibility and Directories** (access, stat, chmod, chown ..., chdir, chroot...), **Process Control:** (Running Linux Commands from C, fork(), the exec family, wait(), exit() ), Thread creation-a simple implementation.

[No. of Hrs.: 09 Hrs]

**TEXT BOOKS:**

1. Yashwant Kanetkar, "Let us C", BPB Publications, 2002.
2. Mark Mitchell, Jeffrey Oldham, and Alex Samuel, "Advanced Linux Programming", New Riders Publishing, 2001.
3. B. Kernighan and D. Ritchie, "The ANSI C Programming Language", PHI., 2000

**REFERENECES:**

1. Yashwant Kanetkar, "Pointers in C", BPB Publications, 2002.
2. Paul Deitel and Harvey Dietel, "How to Program", PHI, 6<sup>th</sup> Ed., 2010.
3. Behrouz A. Forouzan and Richard F. Gilberg, "Computer Science A Structured Programming Approach Using C", PHI, 3<sup>rd</sup> Ed., 2007.
4. Jeri R. Hanly and Elliot B. Koffman, "Problem Solving and Programming in C", Pearson, 5<sup>th</sup> Ed. 2007.
5. Rama N. Reddy and Carol A. Ziegler, "C Programming for Scientist and Engineers with Applications", Jones and Bartlet, 2010.

**PRACTICAL:****Course Code: MCA-153****Paper: Programming in C Lab****L P C****0 4 2****LEARNING OBJECTIVE:**

In this course, which is based on the above theory course, the working skill of learners in Data and File Structure will be developed.

**COURSE OUTCOME (CO):**

After completion of the Practical Course, the learners will be able to:-

CO1	Develop programs in C for different algorithms using basic building blocks of the C language.(BTL6)
CO2	Develop efficient programs using advanced data types, pointers and dynamic memory allocation functions.(BTL6)
CO3	Implement real-world computing solutions through appropriate usage of the pre-processor as well as file handling on Ubuntu environment, using library functions and system calls.(BTL3)
CO4	Apply C constructs to programming problems to control, manipulate files, directories and processes on Ubuntu.(BTL3)
CO5	Work in teams to develop project for real-life cases.(BTL6)