Subject : Computer Programming Lab
Semester : 1st
Department : Information Technology
Course No. : ITP100
Credits : 1
L T P : 0 0 2

Course Outcomes:

CO1	To provide exposure to problem solving through programming.
CO2	To understand the concept of various tools available in C and to learn how to solve the problems using the code.
CO3	To understand the basics of programming including the most common library functions.
CO4	To learn how to write a program in C using loops, functions, pointers etc.

Lab Details:

1. Programs to understand how integers, characters, and strings are stored and represented in C.
2. Programs to understand the ASCII character encoding.
3. Programs to understand how to use different operators available in C.
4. Programs to understand differences between a logical and arithmetic operators.
5. Programs to understand differences between a logical and bitwise operators.
6. Programs to obtain a full understanding of signed, unsigned, long and short numbers in C.
7. Programs to understand exactly how numbers are represented in computers(octal,hexadecimal and binary numbers systems) .
8. Programs to evaluate algebraic expressions in C.
9. Programs to understand printing of various data types using different output functions.
10. Programs to exercise all flags in printf() functions.
11. Programs to understand printing of display patterns of numbers and asterisks.
12. Programs to understand taking input from user using different input functions.
13. Programs to exercise all flags in scanf() functions.
14. Programs to understand how arrays work in C, how to use them, and how they are stored in memory.
15. Programs to understand searching in an array.
16. Programs to understand sorting techniques using arrays.

17. Programs to understand pointers in C.
18. Programs to understand the relationship between array indexing and pointer arithmetic.
19. Programs to understand dynamic memory allocation especially with respect to1D and 2D arrays.
20. Programs to understand modularize of code using functions.
21. Programs to implement function with/without arguments and with/without return types.
22. Programs to understand direct and indirect recursions using functions.
23. Programs to use pointer to pass the address of data and arrays to functions.
24. Programs to understand static data types and static functions.
25. Programs to understand creating, accessing and using structures.
26. Programs to understand use of arrays of structures.
27. Programs to understand pointers to structures and pointers as structures members.
28. Programs to understand creating, accessing and using unions.
29. Programs to understand creating, reading, writing a file.
30. Programs to understand taking input through arguments to main() function.