

KLiC C Programming

Detailed Syllabus:

KLiC C Programming

1) Getting Started

- Brief Introduction
- Programming Language
- About C Programming
- C Character Set
- Constants, Variables & Keywords
- Constants in C
- Variables in C
- Writing a C Program
- Instructions and Assignments
- Basic Operators in C Programming

2) The Decision Control Structure

- Decisions Control Structure & the If Statement
- The if-else Statement
- Use of Logical Operators
- Different types of Operators
- Points to remember

3) Loop Control Structure

- Loops and the While loop
- While Loop
- For Loop
- Operators in Loop
- The Odd Loop
- Break Statement
- Continue Statement
- do-while loop
- Tips to remember

4) Case Control Structure

- Decisions using switch
- The Tips and Traps
- Switch versus if-else Ladder
- The goto keyword

5) Functions and Pointers

- About Functions
- Passing Values between Functions
- Scope Rule of Functions

- Calling Convention
- One Dickey Issue
- Advanced Features of Functions
- Function Declaration and Prototypes
- Call by Value or Call by Reference
- An Introduction to Pointers
- Pointer Notation
- Function Calls
- Basics of Call by value and call by reference
- Conclusions
- Recursion
- Recursion and Stack
- Adding Functions to the Library

6) Data Types Revisited

- Data type
- Integer number variables
- Integers, signed and unsigned
- Chars, signed and unsigned
- Floats and Doubles
- Issues related to Data types
- Storage Classes in C
- Automatic Storage Class
- Register Storage Class
- Static Storage Class
- External Storage Class
- To study the Ground rules for the Storage Class

7) The C Preprocessor

- Features of C Preprocessor
- Preprocessor and Macro Directives
- Macros with Arguments and Macros versus Functions
- Various Directives

8) Arrays

- About Array
- Usage of Arrays
- Pointers and Arrays
- Passing an Entire Array to a Function
- Two Dimensional Arrays
- Initializing a 2-Dimensional Array
- Memory Map of a 2-Dimensional Array
- Pointers and 2-Dimensional Arrays
- Pointer to an Array 295
- Passing 2-D array to a Function

- Array of Pointers
- Three Dimensional Array

9) Pupating on Strings

- What are Strings?
- Pointers and Strings
- Standard Library String Functions
- Two-Dimensional Array of Characters
- Array of Pointers to Strings
- Limitations of Array of Pointers to Strings

10) Structures

- Why Use Structures?
- Declaring a Structure
- Accessing Structure Elements
- Array of Structures
- Additional Features of Structures
- Uses of Structures

11) Console Input/output

- Types of I/O
- Formatted Console I/O Functions
- printf() and scanf() Functions
- Unformatted Console I/O Functions

12) File Input/Output

- Data Organization
- File Operations
- Opening a File
- Reading from a File
- Trouble in Opening a File
- Closing the File
- Counting Characters, Tabs, Spaces
- A File-copy Program
- Writing to a File
- File Opening Modes
- String (line) I/O in Files
- The Awkward Newline
- Record I/O inFiles
- Text Files and Binary Files
- Record I/O Revisited
- Database Management
- Low Level Disk I/O
- A Low Level File-copy Program
- I/O Under Windows

13) More Issues in Input/output

- Using argc and argv
- Detecting Errors in Reading/Writing
- Explanation
- Standard I/O Devices
- I/O Redirection
- Redirecting the Output
- Redirecting the Input & Both Ways at Once

14) Operations on Bits

- Binary System & Bitwise Operators
- Bitwise AND Operator
- Bitwise OR Operator
- Bitwise XOR Operator
- One's Complement Operator
- Shift Operator
- The showbits() Function

15) Miscellaneous Features

- Enumerated Data Type and its uses
- Understanding with a Program
- Renaming Data Types with typedef
- Typecasting
- Bit Fields
- Pointers to Functions
- Functions Returning Pointers
- Functions with Variable Number of Arguments
- Unions & Union of Structure

16) Under Windows 535

- Uses of Windows
- Integers
- The Use of typedef
- Pointers in the 32-bit World
- Memory Management & Device Access
- DOS Programming Model
- Windows Programming Model
- Event Driven Model & Windows programming
- The First Windows Program
- Hungarian Notation

17) Windows Programming

- The Role of a Message Box
- Here Comes the windows

- More Windows
- A Real-World Window
- Creation and Displaying of Window
- Interaction with Window
- Reacting to Messages
- Program Instances

18) Graphics under Windows

- Graphics fundamentals
- Device Independent Drawing
- Hello Windows program
- Drawing Shapes
- Types of Pens
- Types of Brushes
- Code and Resources
- Freehand Drawing, the Paintbrush Style
- Capturing the Mouse
- Device Context, a Closer Look
- Displaying a Bitmap
- Animation at Work
- WM_CREATE and On Create()
- WM_TIMER and On Timer()
- Points to remember

19) interaction with Hardware

- Hardware Interaction
- Hardware Interaction, DOS Perspective
- Hardware Interaction, Windows Perspective
- Communication with Storage Devices
- The Read Sector() Function
- Accessing Other Storage Devices
- Communication with Keyboard
- Dynamic Linking
- Windows Hooks
- Caps Locked, Permanently
- Mangling Keys
- Key Logger

20) Under Linux

- What is Linux
- C Programming Under Linux
- The 'Hello Linux' Program
- Processes
- Parent and Child Processes
- More Processes

- Zombies and Orphans

21) More Linux Programming

- Communication using Signals
- Handling Multiple Signals
- Registering a Common Handler
- Blocking Signals
- Event Driven Programming

22) memory Mapping

- Introduction to Memory Map
- Memory Organization
- Segmentation
- Loading OS & Booting Process
- The resident and transient memory area
- Program memory area at run time
- Memory representation of data & function objects

23) C Traps & Pitfall

- Introduction
- Lexical pitfalls
- Exceptions, String & characters
- Understanding Declaration
- Exceptions in Operators' precedence
- Use of Semicolons
- The Switch statement
- Calling functions
- The Dangling else problem
- Linkages
- External Types
- Expression evaluation sequence
- Issues related to actual parameters
- Eshew Synecdoche
- Library Function
- Preprocessor
- Portability pitfalls
- Signed & Unsigned characters
- Random numbers
- Portability problems

```

File Edit Search Run Compile Debug Project Options Window Help
PRIME.C 1
PRIMENUM.C 2=[↑]
int isPrime(intd num);
void main()
{
    int num,ans;
    char res;
    h;
    clrscr();
    printf("Enter any ddrNumber:-->");
    scanf("%d",&num);
    if(isPrime(num))
        printf("Number is Prime");
    else
        printf("Number is not prime");
}
1:17
Message 3
Compiling PRIMENUM.C:
•Error PRIMENUM.C 1: 'indt' cannot start a parameter declaration
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```

Prime Numbers

```

Turbo C++ IDE
File Edit Search Run Compile Debug Project Options
REVERSE_.C
#include<stdio.h>
void main()
{
    void reverse_rec(char*);
    int S[50];
    printf("\nENTER STRING TO REVERSE : ");
    scanf("%s",&S);
    printf("\nREVERSED STRING IS :- ");
    reverse_rec(S);
    getch();
}
void reverse_rec(char *STR)
{
    if(*STR)
    {
        reverse_rec(STR+1);
        printf("%c",*STR);
    }
}

```

Reversing an integer