






Guide2Code - C Programming Roadmap

◆ Phase I: Beginner Level

📌 Topics to Learn:

1. Basic Syntax & Structure (main function, header files, comments)
2. Variables & Data Types (int, float, char, double, etc.)
3. Operators (arithmetic, relational, logical, bitwise)
4. Control Statements (if-else, switch-case)
5. Loops (for, while, do-while)
6. Functions (function declaration, definition, recursion)
7. Arrays & Strings (1D, 2D arrays, string handling functions)

🚀 Beginner Project Ideas:






-  **Calculator** – Basic arithmetic operations
-  **Student Grade System** – Input marks and display grades
-  **Number Guessing Game** – User guesses a number, program provides hints
-  **Simple ATM System** – Withdraw, deposit, check balance
-  **Temperature Converter** – Celsius ↔ Fahrenheit conversion

◆ Phase 2: Intermediate Level

📌 Topics to Learn:

1. Pointers & Memory Management (malloc, calloc, free)
2. Structures & Unions (data organization)
3. File Handling (read, write, append)
4. Dynamic Memory Allocation
5. Command-Line Arguments
6. Sorting and Searching Algorithms
7. Preprocessor Directives (macros, #define, #include)

Intermediate Project Ideas:






-  **Library Management System** – Add, remove, search books
-  **Contact Management System** – Store, retrieve, and edit contacts
-  **Bank Management System** – Manage accounts, transactions
-  **Quiz Game** – Display questions and evaluate user input
-  **Employee Record System** – Store and display employee details

Phase 3: Advanced Level

Topics to Learn:

1. Data Structures (linked lists, stacks, queues, trees, graphs)
2. Multi-threading and Concurrency
3. Networking in C (sockets, client-server model)
4. System Programming (working with OS-level operations)
5. Interprocess Communication (pipes, message queues, shared memory)
6. Compiler Design Basics (tokenization, parsing)
7. Embedded C Programming

Advanced Project Ideas:

-  **Operating System Shell** – Basic command-line shell
-  **Chat Application** – Real-time messaging using sockets
-  **Text Editor** – Implement features like find, replace, and save
-  **Compiler Simulator** – Convert expressions into assembly-like code
-  **IoT-Based Smart Home System** – Control devices using C