1. **Working with Larger Programs**

* Compiling multiple source files from the command line
* Makefiles
* Communication between files
* Using Header files effectively
* Heap and Stack Memory Allocation

1. **Storage Classes**

* Automatic Variables
* External
* Static
* Register

1. **Advanced Data**

* The #define statement (constants)
* Using typedef
* Variable Length Arrays
* Flexible Array Members
* Complex number types
* Designated Initializers

1. **Type Qualifiers**

* const
* volatile
* restrict

1. **Bit Manipulation**

* Binary numbers and bits
* Bitwise Operators (Logical)
* Bitwise Operators (Shifting)
* Bitmasks
* Using Bit Operators to pack data
* Using Bit Fields to pack data

1. **Advanced Control Flow**

* The goto statement
* The null statement
* The comma operator
* setjmp and longjmp functions

1. **Input and Output**

* char functions (input)
* char functions (output)
* string functions
* Formatting functions

1. **Advanced Function Concepts**

* Variadic Functions
* va\_copy
* Recursion
* Inline Functions
* \_Noreturn Functions

1. **Unions**

* Defining a Union
* Accessing Union Members

1. **The Preprocessor**

* Conditional Compilation
* Include guards and #undef
* #pragma and #error

1. **Macros**

* Macros vs. Functions
* Creating your own Macros
* Preprocessor Operators
* Predefined Macros

1. **Advanced Debugging, Analysis, and Compiler Options**

* GCC Compiler Options
* Debugging with the preprocessor
* Debugging with gdb
* core files
* Profiling
* Static Analysis

1. **Advanced Pointers**

* Double pointers
* Double pointers (pointer to a pointer)
* Function pointers
* void pointers

1. **Static Libraries and Shared Objects**

* Creating a Static Library (archive)
* Creating a Dynamic Library
* Dynamically loading a shared object

1. **Useful C Libraries**

* Assert
* General Utilities (stdlib.h)

1. **Data Structures**

* Abstract Data Types
* Linked Lists
* Stacks
* Binary Trees

1. **Interprocess Communication and Signals**

* Interprocess Communication
* Signals
* Raising a Signal
* Handling a Signal using the signal function
* Handling a Signal using sigaction
* The fork() system call

1. **Threads**

* Creating a thread
* Passing arguments and returning values
* Common Thread functions
* Thread Synchronization Concepts
* Mutexes
* Condition Variables