Advanced C Programming

Linux Commands & Shell scripting

* Introduction to Linux & History of Linux
* Features of Linux
* Basic Linux commands, usage of vi editor.
* usage of gcc, g++ and their common option parameters.
* The architecture of Linux OS
* Linux Commands
* vim editor
* File operation
* Redirections
* Filters
* Bash Shell scripting

Basic C Programming

• Introduction to C

• Constants

• Data types and Variables

• scanf

• printf

• Comments

• Operators

o Assignment operators

o Arithmetic operators

o Increment & decrement operators

o Compound assignment operations

o Relational operators

o Logical operators

o Conditional operators

o Type casting

o sizeof operator

Control Statements

• Conditional Control Statements

o If, if-else

o nested if-else, if-else-if ladder

• Multiple Branching Control Structure

o switch-case

• Loop Control statements

o while

o do-while

o for

• Nested Loops

• Jump Control structures

o break

o continue

Functions

• Modular Programming

• void Functions

• Calling a void Function

• Parameters

• Calling a function with arguments

• Function returning a value

• return statement

• Calling a function that returns a value

• Function Prototype

• Local variables and global variables

• Parameter Passing Methods

o Call by value

o Call by reference

• Recursion

• Library Functions

Data Structures in C

• Arrays

o Creating and using arrays

o Character arrays

o 2 dimensional arrays

o Passing arrays as arguments to functions

• Structures

o Creating and using structures

o Array as the member if a structure

o Array of structures

o Structure as the member of a structure

o Passing structures as parameters to functions

o Returning a structure from a function

• Unions

o Creating a using unions

o Difference between structures and unions

Character and String Functions

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

Pointers in C

• Pointers

o Declaring and using a pointer

o \* operator

o & operator

o Array of pointers

o Passing pointers as arguments to functions

o Pointer to an array

▪ Pointer arithmetic

o Pointer to a structure

▪ The -> operator

o Pointer to a pointer

o Returning a pointer

ADVANCED POINTERS

* Double pointers (pointer to a pointer)
* Double pointers (pointer to a pointer) Part 2
* (Challenge) Double pointers
* (Demonstration) Double pointers
* Function pointers
* (Challenge) Function pointers
* (Demonstration) Function pointers
* void pointers

Files

* Input and Output File Handling
* Opening & Closing a File
* Modes of File Opening
* Detection End of File
* File Pointers & Their Manipulation
* Text and Binary Files
* Sequential Files
* Random Access Files

Dynamic Memory Allocation

▪ malloc function

▪ calloc function

▪ ralloc function

▪ free function

▪ Using malloc/calloc and free with arrays

ADVANCED DEBUGGING, ANALYSIS, AND COMPILER OPTIONS

* GCC Compiler Options (part 1)
* GCC Compiler Options (part 2)
* Debugging with the preprocessor
* Debugging with gdb (part 1)
* Debugging with gdb (part 2)
* core files
* Profiling
* Static Analysis
* valgrind tool

WORKING WITH LARGER PROGRAMS

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

STATIC LIBRARIES AND SHARED OBJECTS

* Overview
* Creating a Static Library (archive)
* (Challenge) Static Library
* (Demonstration) Static Library
* Creating a Dynamic Library (Shared object)
* (Challenge) Dynamic Library
* (Demonstration) Dynamic Library
* Dynamically loading a shared object
* (Challenge) Dynamic Loading
* (Demonstration) Dynamic Loading

USEFUL C LIBRARIES

* Assert
* Lecture General Utilities (stdlib.h)
* Lecture Date and Time functions

THE PREPROCESSOR

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

Macros

* Overview
* Macros vs. Functions
* Creating your own Macros
* Preprocessor Operators
* Predefined Macros
* (Challenge) Macros
* (Demonstration) Macros

THREADS

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

UML & Design patterns

* UML – Class diagram
* Use case diagram
* Sequence diagram
* Introduction to Design Patterns
  + Singleton

Git and GitHub

* Introduction to Git
* Version control
* Repositories and Branches
* Working Locally with GIT
* Working Remotely with GIT

Case Study

1. Case study topics approved from the project

2. Team size of 5

3. Team needs to follow software engineering process

4. Tools used -Linux OS

5. Assessment parameters-

a. Understanding the problem

b. Completion of Functionalities

c. Coding standards & clean code

d. Usage of DevOps tools

e. Output

f. SRS

g. Design document

h. Test plan

i. Technical viva