**Library Module – Static library**

1. Create 3 files as below. Let cal\_utility.c, .h files be part of the library

· libapplication.c – will contain main() and will invoke functions in cal\_utility.c

· cal\_utility.c – will contain atleast 2 or more functions [ You may add definitions of the functions in this file ]

· cal\_utility.h – will contain the extern declarations/prototypes of the functions in cal\_utility.c

* cal\_utility.h - This header file will declare the prototype of the utility functions.
* cal\_utility.c - This source file will implement the functions declared in the header file.
* libapplication.c – This is the main application that will utilize the function from the utility library.

a. libapplication.c (contains main() and invokes functions from cal\_utility.c)

Create this file with the following content:

#include "cal\_utility.h"

int main() {

int sum = add(5, 3); // Calls add function

int diff = subtract(10, 4); // Calls subtract function

printf("Sum: %d\n", sum);

printf("Difference: %d\n", diff);

return 0;

}

b. cal\_utility.c (contains definitions of functions)

Create this file with the following content:

#include "cal\_utility.h"

// Function to add two numbers

int add(int a, int b) {

return a + b;

}

// Function to subtract two numbers

int subtract(int a, int b) {

return a - b;

}

c. cal\_utility.h (contains extern declarations for functions in cal\_utility.c)

Create this header file with the following content:

#ifndef CAL\_UTILITY\_H

#define CAL\_UTILITY\_H

extern int add(int a, int b);

extern int subtract(int a, int b);

#endif

2. Create a Static Library:

In order to use cal\_utility.c as a static library, we need to compile the source file (cal\_utility.c) into an object file (cal\_utility.o), and then create the static library (libcal\_utility.a).

a. Compile the cal\_utility.c file into an object file:

**gcc -c cal\_utility.c -o cal\_utility.o**

This generates cal\_utility.o, which is the object file for cal\_utility.c.

b. Create the static library libcal\_utility.a:

**ar rcs libcal\_utility.a cal\_utility.o**

This command creates the static library libcal\_utility.a from the object file cal\_utility.o.

3. Compile the Application with the Static Library:

Now, we need to compile libapplication.c and link it with the static library libcal\_utility.a.

**gcc -o libapplication libapplication.c -L. -lcal\_utility**

4. Execute the Application:

After successfully compiling the application, run it using:

**./libapplication**

This should print the sum and difference of the two numbers, as calculated by the functions from cal\_utility.c.