**ITT-310 Design Report Template**

|  |  |
| --- | --- |
| **Topic:** | Milestone 5: Final Application Release |
| **Date:** | *10 NOV 2021* |
| **Revision:** | *Final* |
| **Milestone Summary:** | |  |  |  | | --- | --- | --- | | Milestone 5 | **Hours Worked** | **Hours Remaining** | | In this milestone, students will demonstrate the ability to write complex programs using appropriate and secure use of library functions.  Students will design and develop the final release of their C programming application. From previous assignments and activities, students should be proficient with the following skills:   1. Writing a C program that performs file I/O 2. Writing secure code   **Execution**  Execute this assignment according to the following guidelines:   1. Update your C application to demonstrate the following features:    1. Use of file I/O    2. Use of secure programming techniques 2. All code should be fully commented with single-line comments for all logic in code statements and multi-line comments for the all functions as well as code modules. 3. Complete test cases using the “Test Case Template,” located in the Course Materials that validates all functionality of the program. Positive, negative, and abuse test cases MUST be included in the assignment. 4. Update the technical design report with the required design documentation. |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |
| **GIT URL:** | *The GIT URL that can be used to clone your code.* |

**Design Documentation**

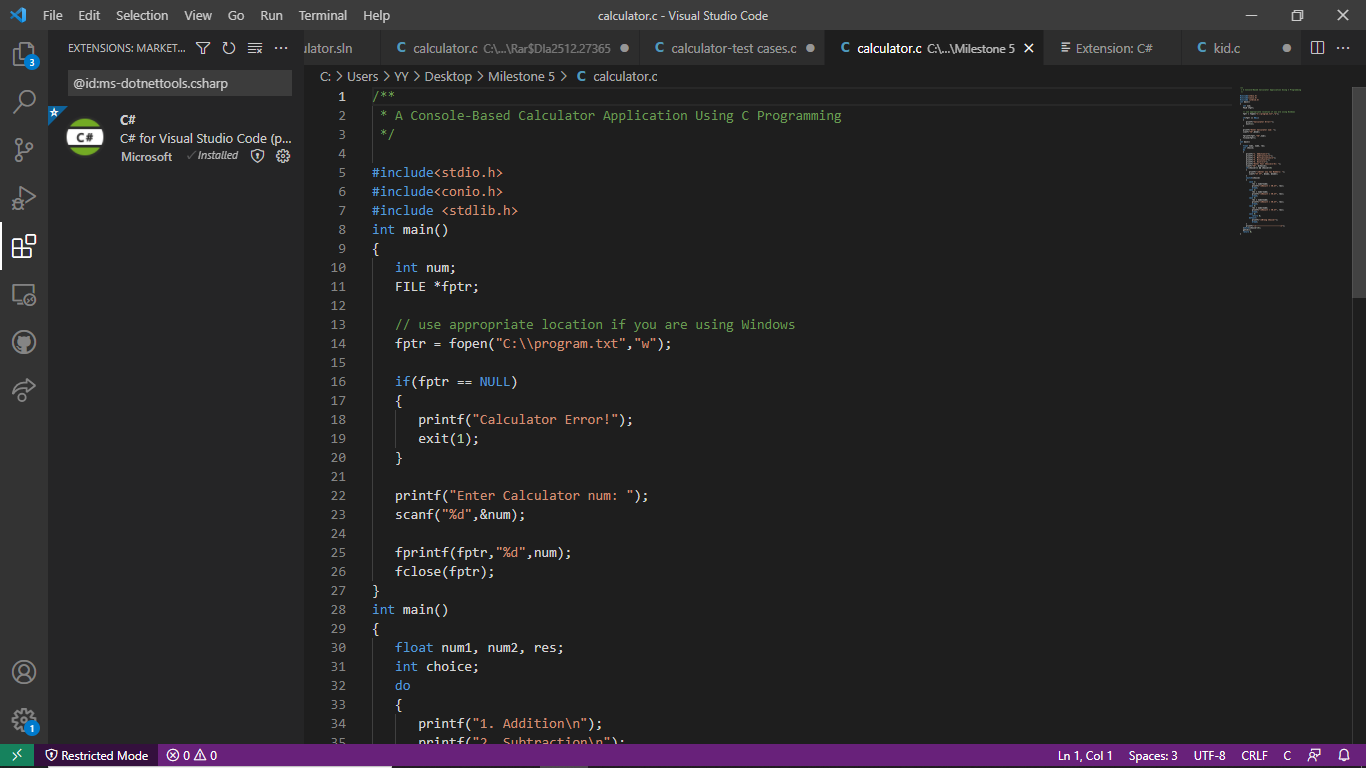
**Execution**

Execute this assignment according to the following guidelines:

Update your C application to demonstrate the following features:

* 1. Use of file I/O
* *This program takes a number from the calculator user and stores in the file program.txt.*
* *After you compile and run the program, you can see a text file program.txt created in drive of your computer. When you open the file, you can see the integer you entered.*
* *The fopen( ) function to create a new file or to open an existing file. This call will initialize an object of the type FILE, which contains all the information necessary to control the stream.*
* *The fclose(fptr) function returns zero on success, or fptr if there is an error in closing the file. This function actually flushes any data still pending in the buffer to the file, closes the file, and releases any memory used for the file. The fptr is a constant defined in the header file stdio.h.*
* *This program takes a number from the calculator user and stores in the file program.txt.*
* *After you compile and run the program, you can see a text file program.txt created in drive of your computer. When you open the file, you can see the integer you entered.*
* /\*\*
* \* A Console-Based Calculator Application Using C Programming
* \*/
* #include<stdio.h>
* #include<conio.h>
* #include <stdlib.h>
* int main()
* {
* int num;
* FILE \*fptr;
* // use appropriate location if you are using Windows
* fptr = fopen("C:\\program.txt","w");
* if(fptr == NULL)
* {
* printf("Calculator Error!");
* exit(1);
* }
* printf("Enter Calculator num: ");
* scanf("%d",&num);
* fprintf(fptr,"%d",num);
* fclose(fptr);
* }
* int main()
* {
* float num1, num2, res;
* int choice;
* do
* {
* printf("1. Addition\n");
* printf("2. Subtraction\n");
* printf("3. Multiplication\n");
* printf("4. Division\n");
* printf("5. Exit\n\n");
* printf("Enter Your Choice(1-5): ");
* scanf("%d", &choice);
* if(choice>=1 && choice<=4)
* {
* printf("\nEnter any two Numbers: ");
* scanf("%f %f", &num1, &num2);
* }
* switch(choice)
* {
* case 1:
* res = num1+num2;
* printf("\nResult = %0.2f", res);
* break;
* case 2:
* res = num1-num2;
* printf("\nResult = %0.2f", res);
* break;
* case 3:
* res = num1\*num2;
* printf("\nResult = %0.2f", res);
* break;
* case 4:
* res = num1/num2;
* printf("\nResult = %0.2f", res);
* break;
* case 5:
* return 0;
* default:
* printf("\nWrong Choice!");
* break;
* }
* printf("\n------------------------\n");
* }while(choice!=5);
* getch();
* return 0;
* }

1. All code should be fully commented with single-line comments for all logic in code statements and multi-line comments for the all functions as well as code modules.

**

1. Update the technical design report with the required design documentation.

## **Opening a File**

*You can use the fopen( ) function to create a new file or to open an existing file. This call will initialize an object of the type FILE, which contains all the information necessary to control the stream. The prototype of this function call is as follows −*

FILE \*fopen( const char \* filename, const char \* mode );

## **Closing a File**

*To close a file, use the fclose( ) function. The prototype of this function is −*

int fclose( FILE \*fp );

*The fclose(-) function returns zero on success, or fptr if there is an error in closing the file. This function actually flushes any data still pending in the buffer to the file, closes the file, and releases any memory used for the file. The fptr is a constant defined in the header file stdio.h.*

*There are various functions provided by C standard library to read and write a file, character by character, or in the form of a fixed length string.*

1. Complete a screencast, using Loom or a similar free application, that provides the audience with on overview of the project, what functionality was implemented, how your embedded application was designed, and brief functional demonstration of your application running on the embedded board. Make sure to use industry terminology.

