LAB 2

Libraries + Assignment 1

Objectives

- Create an example project in CLion that uses libraries
- Familiarize with an example program that reads a file line by line.
- Address Point 1 of the assignment by creating and using custom libraries.

Create and Use Libraries

Creating and Using Libraries

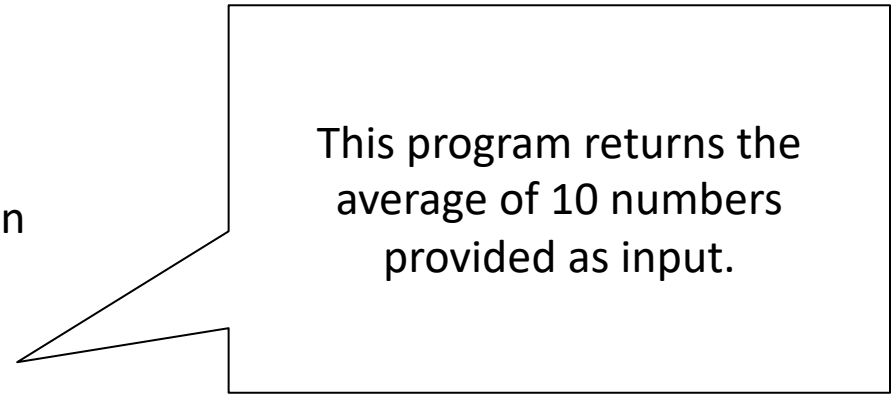
- Libraries allow creating re-usable software modules.
- To create libraries you need to define 2 files:
  - (*.h) ➔ It is a header file containing the method prototypes
  - (*.c) ➔ It is a source file containing the implementation of the methods listed in the header file.

Let’s Do It!

Create the main source file

- Create a new project in CLion
- Cut and paste the following in your main file:

```
#include <stdio.h>
#include <stdlib.h>
#include "library.h"
#define MAX_SIZE  10
int main(){
        //result of the average function
        double average = 0;
        //numbers to be averaged
        int numbers[10];
        insertNumbers(numbers, MAX_SIZE);
        average = avg(numbers,MAX_SIZE);
         printf ("\nThe average is: %.2f", average);
}
```



Creating and Using Libraries

- Create a new project in CLion
- Cut and paste the following in your main file:

```
#include <stdio.h>
#include <stdlib.h>
#include "library.h"
#define MAX_SIZE  10
int main(){
        //result of the average function
        double average = 0;
        //numbers to be averaged
        int numbers[10];
        insertNumbers(numbers, MAX_SIZE);
        average = avg(numbers,MAX_SIZE);
         printf ("\nThe average is: %.2f", average);
}
```

- It uses 2 functions that are not implemented inside the main (insertNumbers and avg).

Creating and Using Libraries

- Create a new project in CLion
- Cut and paste the following in your main file:

```
#include <stdio.h>
#include <stdlib.h>
#include "library.h"
#define MAX_SIZE  10
int main(){
        //result of the average function
        double average = 0;
        //numbers to be averaged
        int numbers[10];
        insertNumbers(numbers, MAX_SIZE);
        average = avg(numbers,MAX_SIZE);
         printf ("\nThe average is: %.2f", average);
}
```

These functions are implemented in another module called library which is imported in the main using *include*.

Creating and Using Libraries

- In your CLion Project create a new C/C++ Header file and call it "*library.h*"
- Cut and paste the following in library.h as follows:

```
#ifndef LAB2_LIBRARY_H
#define LAB2_LIBRARY_H
/* * Calculates the average of a set of integers. It takes as input:
   * - numbers: the numbers to be averaged
   * - size: home many numbers should be averaged
   * It returns the result of the average function */
double avg(int numbers[], int size);

/* * Allows to insert a set of integers form the standard input.
   *   It takes as input
   * - numbers: the numbers to be inserted
   * - maxSize: maximum amoount of numbers to be inserted */
void insertNumbers(int numbers[], int maxSize);
#endif //LAB2_LIBRARY_H
```

Creating and Using Libraries

- In your CLion project create a new C/C++ Source File and call it *"library.c"*
- Cut and paste the following in library.c:

```
#include <stdio.h>

#include "library.h"
double avg(int numbers[], int size){
    int i =0;
    double result =0;
     for(i=0; i< size; i++)
         result += numbers[i];
     return result/size;
}

void insertNumbers(int numbers[], int maxSize){
    int i =0;
     for(i =0; i<maxSize;i++){
       printf("insert an integer number: ");
       scanf ("%d", &numbers[i]);    }
}
```

Creating and Using Libraries

- In your CLion project create a new C/C++ Source File and call it *"library.c"*
- Cut and paste the following in library.c:

```
#include <stdio.h>
#include "library.h"
double avg(int numbers[], int size){
    int i =0;
    double result =0;
     for(i=0; i< size; i++)
         result += numbers[i];
     return result/size;
}

void insertNumbers(int numbers[], int maxSize){
    int i =0;
     for(i =0; i<maxSize;i++){
       printf("insert an integer number: ");
       scanf ("%d", &numbers[i]);    }
}
```

It includes the library that contains the method prototypes it implements.

Creating and Using Libraries

- In your CLion project create a new C/C++ Source File and call it *"library.c"*
- Cut and paste the following in library.c:

```
#include <stdio.h>
#include "library.h"
double avg(int numbers[], int size){
    int i =0;
    double result =0;
     for(i=0; i< size; i++)
         result += numbers[i];
     return result/size;
}

void insertNumbers(int numbers[], int maxSize){
    int i =0;
     for(i =0; i<maxSize;i++){
       printf("insert an integer number: ");
       scanf ("%d", &numbers[i]);    }
}
```

Implemented methods

How is a library imported?

```
#include <stdio.h>
#include <string.h>
#include “library.h”
…
```

If you open *main.c* or *library.c* you will see the lines above

- Remember that your project should only contain 1 main method.
- The source file implementing the methods declared in the library should not contain a main method.

Now try to run the following example in a new CLion Project.

Example

- Create a new directory inside your project called *inputFiles*
- Then, create inside the inputFiles directory, a new text file and call it *"weather.txt"*
- The file should look like the following

```
Athlone 10C
Galway 9C
Dublin 11C
Cork 9C
```

Example

- Create a main source file and cut and paste the program in the following slide.
- Modify the location of weather file depending on the location of this file in YOUR COMPUTER.

Exercise 2

```
Include <stdio.h>
// Remember to MODIFY the location of weather file in YOUR COMPUTER
const char *WEATHER_FILE_PATH = ”./inputFiles/weather.txt";
char weatherArr[5][70]; // stores weather data (limited to five rows)
int lineNum;
int i;

Int main ()
{
FILE *fp = fopen(WEATHER_FILE_PATH,  "r+" );      /* open for reading */

// This will take each row in the file and store it in weatherArr.
        if (fp == NULL ){     /* check does weather file exist etc */
                perror ("Error opening weather file");
                lineNum = -1; /* use this as a file not found code */
        } else {
                // fgets returns NULL when it gets to the end of the file
                while ( fgets( weatherArr[lineNum], sizeof(weatherArr[lineNum]), fp ) != NULL ) {
                        lineNum++;
                }
                fclose (fp);
        }
// Print out  the lines that were read from the file
for(i =0; i < lineNum; i ++)
  {
        printf(“\n%s\n”, weatherArr[i]);
  }
}
```

Source: Material by Prof. P Cunningham 2015

Now put it together…

- For the rest of the lab, start familiarising with the libraries and the example provided.

- A target objective for this week is to address point 1 in the assignment: Read file *input.txt* line by line and place it in a 2D array