Government College University, Lahore

Department of Computer Science

Programming Fundamentals (Spring 2019)

Lab – 08

Week (06-05-2019 – 10-05-2019)

Submission Deadline 17-05-2019 (Friday) 11:59 PM

Startup (01) (Understand Functions)

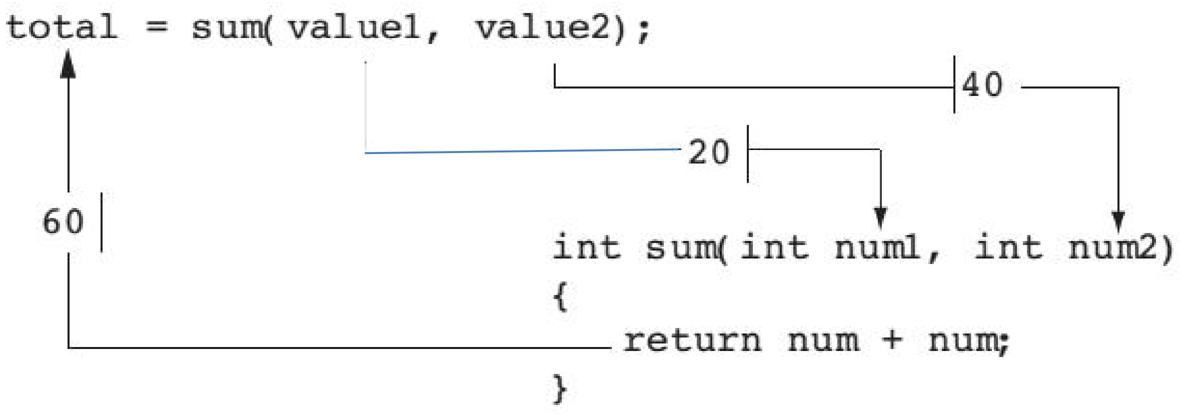
Type the following program in your editor and observe the results

```
#include <iostream>

using namespace std;

void displayMessage(){
     cout<<"Hello from the diplayMessage() Function.\n";
}

main(){

     cout<<"Hello from the Main.\n";

     diplayMessage();

     cout<<"Good Bye from main.";
}
```

Understanding how function call works

This will allow student to understand how a function call works to get results from functions.

```
Total = sum (value1, value2);          Function Call
```

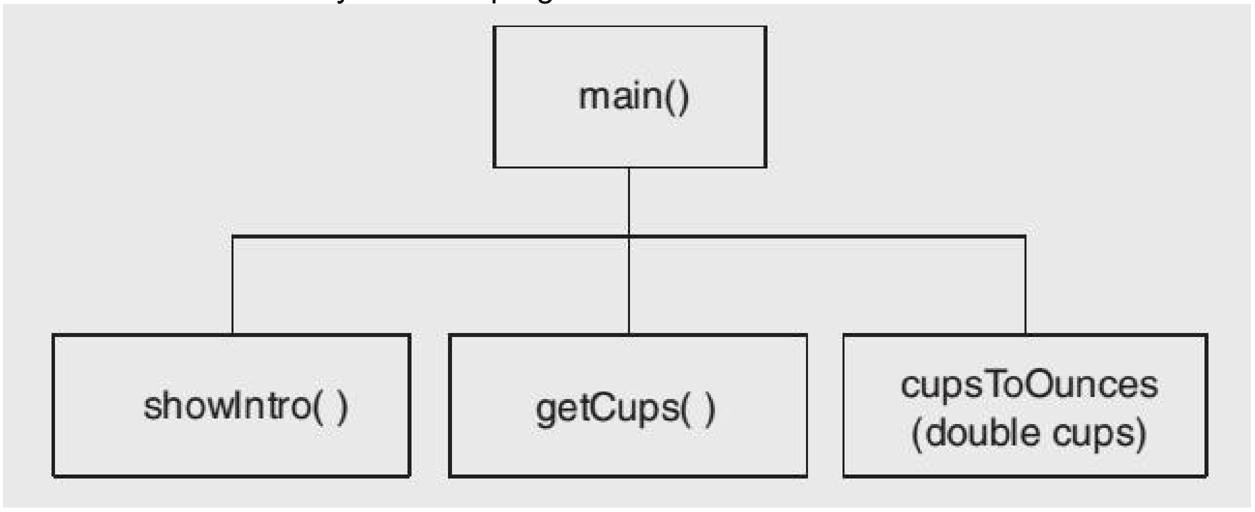


Task-01
Your friend runs a catering company. Some of the ingredients that his recipes required are measured in cups. When he goes to grocery store to buy those ingredients, however they sold only by the fluid ounces. He asked you to write a program that converts cups to fluid ounces.

You design the following algorithm

1. Display an introductory screen that explains what the program does.
2. Get the number of cups
3. Converts the number of cups to fluid ounces and display the result (One cup is equal to 8 fluid ounces)

Find below the hierarchy that how program is broken down into functions



You have to write a complete program based on the designed algorithm.

Task-02 (Monkey Business)
A local zoo wants to keep track of how many pounds of food each of its three monkeys eats each day during a typical week. Write a program that stores this information in a two dimensional 3 X 7 array, where each row represents a different monkey and each column represents a different day of the week. The program should first have the user input the data for each monkey. Then it should create a report that includes the following information:

- Average amount of food eaten per day by the whole family of monkeys.
- The least amount of food eaten during the week by any one monkey.
- The greatest amount of food eaten during the week by any one monkey.

Input Validation: Do not accept negative numbers for pounds of food eaten.

Task-03 (Rain or Shine)
A meteorologist wants to keep track of weather condition during the past years three month summer season and has designated each day as either rainy ('R'), cloudy ('C'), or sunny ('S'). Write a program that stores this information in a 3 X 30 array of characters, where the row indicates the month (0=June, 1=July, 2=August) and the column indicates the day of the month. Note that data are not being collected for the 31st of any month. The program should begin by reading the weather data through input from user. Then it should create a report that displays, for each month and for the whole three month period, how many days were rainy, how many were cloudy, and how many were sunny. It should also print which of the three months had the largest number of rainy days.

Task-04 (Tic-Tac-Toe Game)
Write a program that allows two players to play a game of Tic-Tac-Toe. Use a two dimensional char array with three rows and three columns as the game board. Each Element of the array should be initialized with an asterisk(*). The program should run a loop that

- Display the contents of the board array
- Allow player 1 to select a location on the board for an X. The program should ask the user to enter the row and column number.
- Allow player 2 to select a location on the board for an O. The program should ask the user to enter the row and column number.
- Determine whether a player has won, or a tie has occurred. If a player has won, the program should declare that player the winner and end. If a tie has occurred, the program should say so and end.

A Player wins when there are three Xs or Os in a row on the game board. The Xs and Os can appear in row, in a columns, or diagonally across the board. A tie occurs when all of the locations on the board are full but there is no winner.

Task-05 (2D array operations)
Write a program that creates a 2D array of integers of size nXn initialized with test data. The program should have the following functions:

- `getTotal`. This function should accept a 2D array as argument and return the total of all the values in the array.
- `getAverage`. This function should accept a 2D array as argument and return the average of all the values in the array.
- `getRowTotal`. This function should accept a 2D array as first argument and an integer as its second argument. The second argument should be the subscript of a row in the array. The function should return the total of the values in the specified row.
- `getColumnTotal`. This function should accept a 2D array as first argument and an integer as its second argument. The second argument should be the subscript of a column in the array. The function should return the total of the values in the specified column.
- `getHighestInRow`. This function should accept a 2D array as first argument and an integer as its second argument. The second argument should be the subscript of a row in the array. The function should return the highest value in the specified row.
- `getLowestInRow`. This function should accept a 2D array as first argument and an integer as its second argument. The second argument should be the subscript of a row in the array. The function should return the lowest value in the specified row.

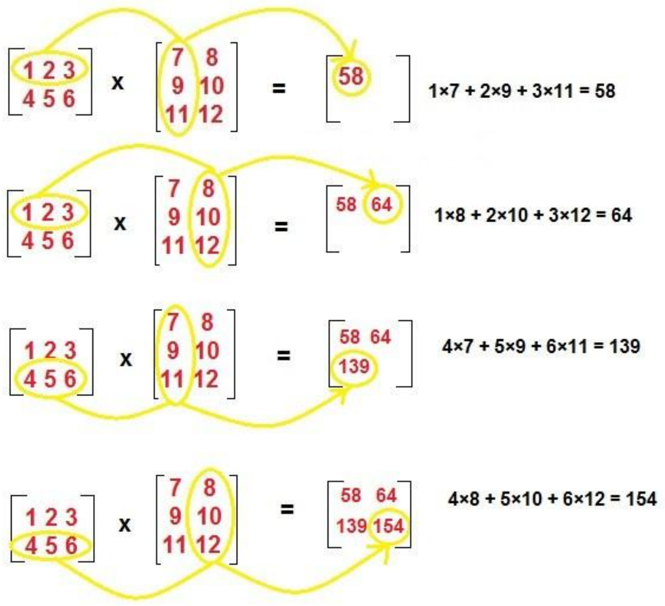
Demonstrate each of the function in your program using switch structure to design menu for function call.

Task-06 (Insert Sort)
Write a function that implements Insert Sort. Then call this function from main to sort a one dimensional array

Task-07 (Sorting 2D array)
Write a Program that sorts 2D array. Write Insert Sort function that accept a 2D array as first argument and an integer as its second argument. The second argument should be the subscript of a row or column in the array. The final array must be sorted in such way that both rows and column are sorted.

Task-08 (Matrix Multiplication)
When we do **multiplication**: The number of columns of the 1st **matrix** must equal the number of rows of the 2nd **matrix**. And the result will have the same number of rows as the 1st **matrix**, and the same number of columns as the 2nd **matrix**.

A procedure of matrix multiplication is explained through the below given image.



Write a program that accepts two matrix according the above definition. First ask the user to enter the size of two matrix and validate that the given matrix size is correct for multiplication. Then after multiplication generate a third matrix and print it.

Task-09 (Matrix Addition)

Two **matrices** may be added or subtracted only if they have the same dimension; that is, they must have the same number of rows and columns. **Addition** or subtraction is accomplished by adding or subtracting corresponding elements. For example, consider**matrix** A and **matrix** B.

$$A + B = \begin{bmatrix} 2 & 2 & 1 \\ 1 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 5 & 7 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 8 \end{bmatrix}$$

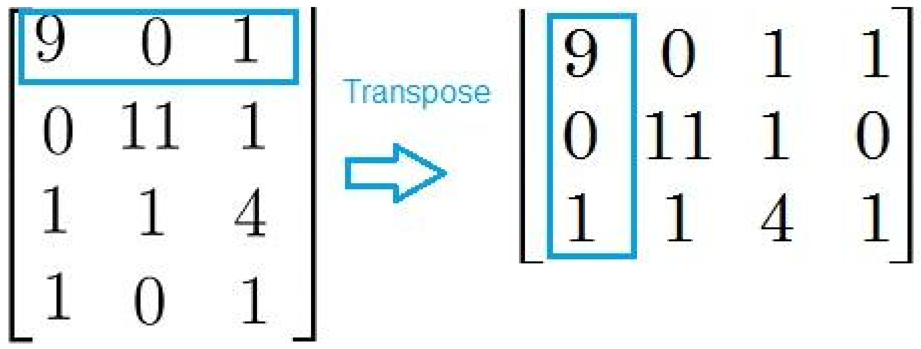
$$= \begin{bmatrix} 2+5 & 2+7 & 1+1 \\ 1+0 & 5+3 & 0+0 \\ 0+1 & 0+0 & 1+8 \end{bmatrix}$$

$$= \begin{bmatrix} 7 & 9 & 2 \\ 1 & 8 & 0 \\ 1 & 0 & 9 \end{bmatrix}$$

Write a program that implements the matrix addition.

Task-10 (Transpose of a Matrix)

The **transpose of a matrix** is a new **matrix** whose rows are the columns of the original. ... The superscript "T" means "**transpose**". Another way to look at the **transpose** is that the element at row r column c in the original is placed at row c column r of the **transpose**.



Write a program that inputs a matrix and then compute the transpose. The transpose of matrix must be stored in new matrix.

Submission Instructions

- Store File as Lab-08-Task-01.cpp (Task Number will change with each task)
- All .cpp files and .docx file must be compresses as YOUR_ROLL_NUMBER-Lab-08.zip

****************** BEST OF LUCK ***********