# Introduction to C/C++

Comp 295

Spring 2023


Department of Computer Science

Forman Christian College University

# Lab 6
# Structures

| Question # | Total Marks |
| --- | --- |
| Question 1 | - |
| Question 2 | - |
| Question 3 | - |
| Question 4 | - |

# In Lab Problem

## Question 1. Rational Numbers

Declare a struct called **RationalNumber** that has two integer member variables. Call the member variables **a** (to mean the numerator) and **b** (to mean the denominator). Then write a function named **toDouble** that takes one **RationalNumber** argument and returns the rational number as double.

Write a main program to test your function.

## Question 2. String Stat

Consider the struct declaration

```
struct StringStat
{
        string s;
        int lower, upper, digit;
};
```

Our aim is to write a program that creates an array of **StringStat** of size 5 and reads the string member variable of each element of the array from the user input. We assume the user will input strings made up lower case alphabets, upper case alphabets, and digits only. Then compute the values of the three member variables lower, upper and digit for each element of the array by counting how many lower case characters, how many upper case characters and how many digit characters the string member variable contains. Finally print each element of the array and the counts in a nicely formatted output.

## Question 3: Student Record

. Consider the **StudentRecord** struct declared as follows:

```
struct StudentRecord
{
        string name;
        float test, midterm, final; //test (20%), midterm (30%) and final (50%)
        char letterGrade;
};
```

Write a program that declares a StudentRecord variable, reads its members (name, test, midterm and final); and then calculates the letter grade. Use this assignment of letter grades [0, 50) = F, [50, 60) = D, [60, 75) = C, [75, 90) = B, and [90, 100] = A). Finally write a function named **viewRecord** that prints the StudentRecord together with the computed letter grade nicely formatted.

## Question 4: Soft Drink Machine

Write a program that simulates a soft drink machine. The program should use a structure that

stores the following data:

Drink Name

Drink Cost

Number of Drinks in Machine

The program should create a dynamic array of four structures. The elements should be initialized with the following data:

| Drink Name | Cost | Number in Machine |
|---|---|---|
| Cola | .75 | 20 |
| Root Beer | .75 | 20 |
| Grape Soda | .80 | 20 |
| Cream Soda | .80 | 20 |

Each time the program runs, it should enter a loop that performs the following steps:

- A list of drinks is displayed on the screen
- The user should be allowed to either quit the program or pick a drink.
- If the user selects a drink, he or she will next enter the amount of money according to the cost shown into the drink machine.
- The program should display the amount of change that would be returned and subtract

one from the number of that drink left in the machine.

- If the user selects a drink that has sold out, a message should be displayed. The loop then

repeats.

- When the user chooses to quit the program, it should display the total amount of money the machine earned.

**Input Validation:** When the user enters an amount of money, do not accept negative values, or

values greater than $1.00.