LAB NO 08

Floating Point Arithmetic's

Fall 2024

CSE-304L Computer Organization and Architecture Lab

Submitted by:

Name : **Hassan Zaib Jadoon**

Reg no. **: 22PWCSE2144**

Class Section **: A**

Signature: ___________

Submitted to:

Dr. Amaad Khalil

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Lab 8: FLOATING POINT Arithmetic

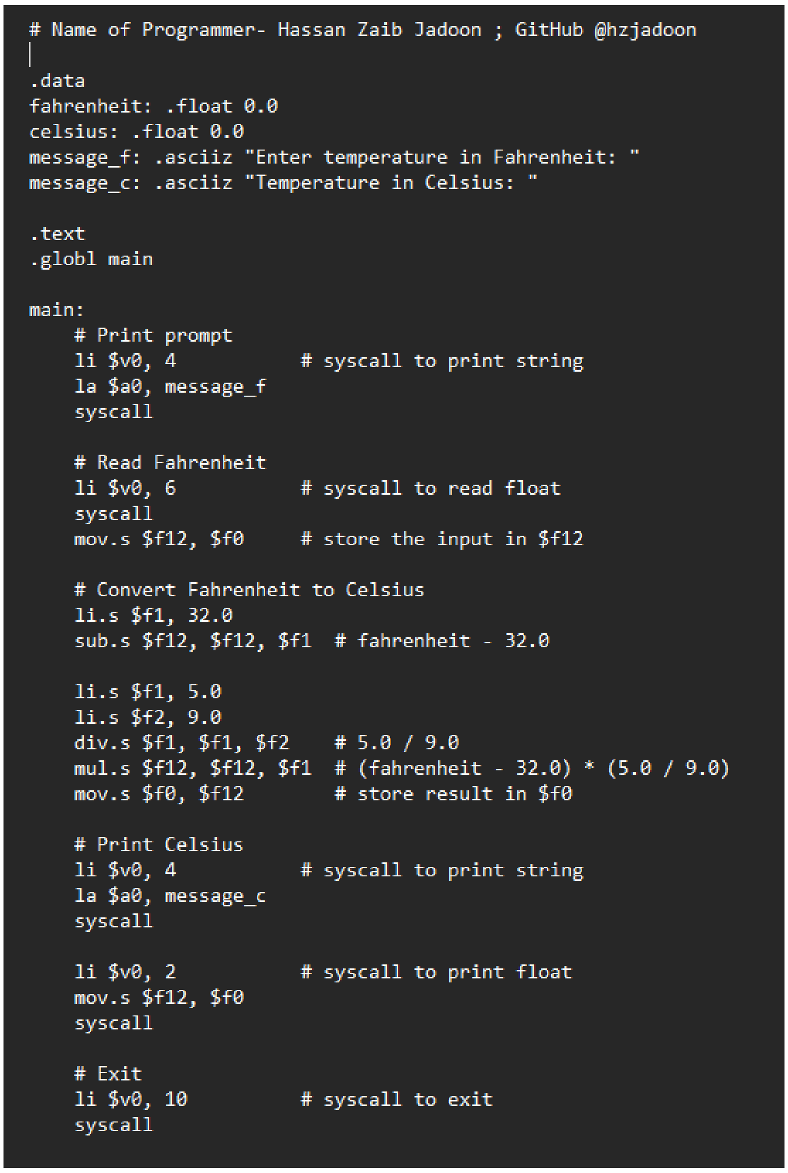
Task 1:

Write a program to convert Fahrenheit to Celsius using the formula below:

Fahrenheit = Celsius * 9.0 / 5.0 + 32.0;

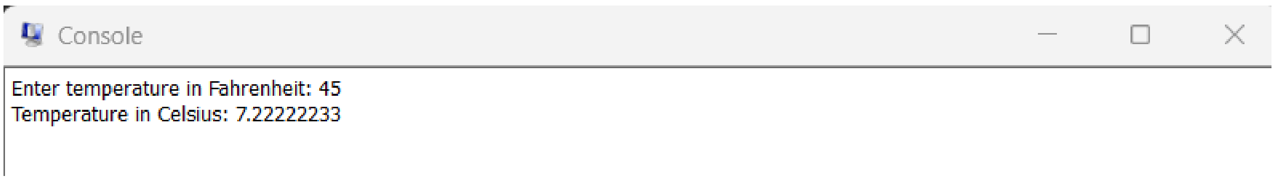
Write a function for this temperature conversion. Celsius value is be passed into the function as single-precision data and Fahrenheit value is returned also in single-precision. Remember to follow all the floating-point function conventions.

Code:



```
# Name of Programmer- Hassan Zaib Jadoon ; GitHub @hzjadoon

.data
fahrenheit: .float 0.0
celsius: .float 0.0
message_f: .asciiz "Enter temperature in Fahrenheit: "
message_c: .asciiz "Temperature in Celsius: "

.text
.globl main

main:
    # Print prompt
    li $v0, 4           # syscall to print string
    la $a0, message_f
    syscall

    # Read Fahrenheit
    li $v0, 6           # syscall to read float
    syscall
    mov.s $f12, $f0     # store the input in $f12

    # Convert Fahrenheit to Celsius
    li.s $f1, 32.0
    sub.s $f12, $f12, $f1  # fahrenheit - 32.0

    li.s $f1, 5.0
    li.s $f2, 9.0
    div.s $f1, $f1, $f2    # 5.0 / 9.0
    mul.s $f12, $f12, $f1  # (fahrenheit - 32.0) * (5.0 / 9.0)
    mov.s $f0, $f12        # store result in $f0

    # Print Celsius
    li $v0, 4           # syscall to print string
    la $a0, message_c
    syscall

    li $v0, 2           # syscall to print float
    mov.s $f12, $f0
    syscall

    # Exit
    li $v0, 10          # syscall to exit
    syscall
```

Output:

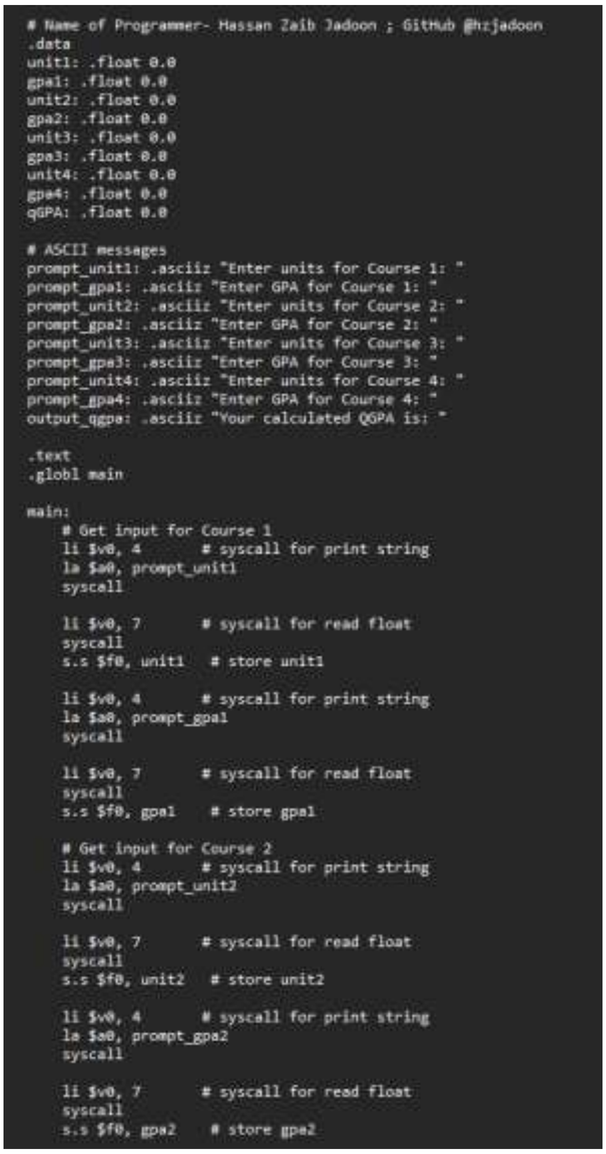


```
Enter temperature in Fahrenheit: 45
Temperature in Celsius: 7.22222233
```

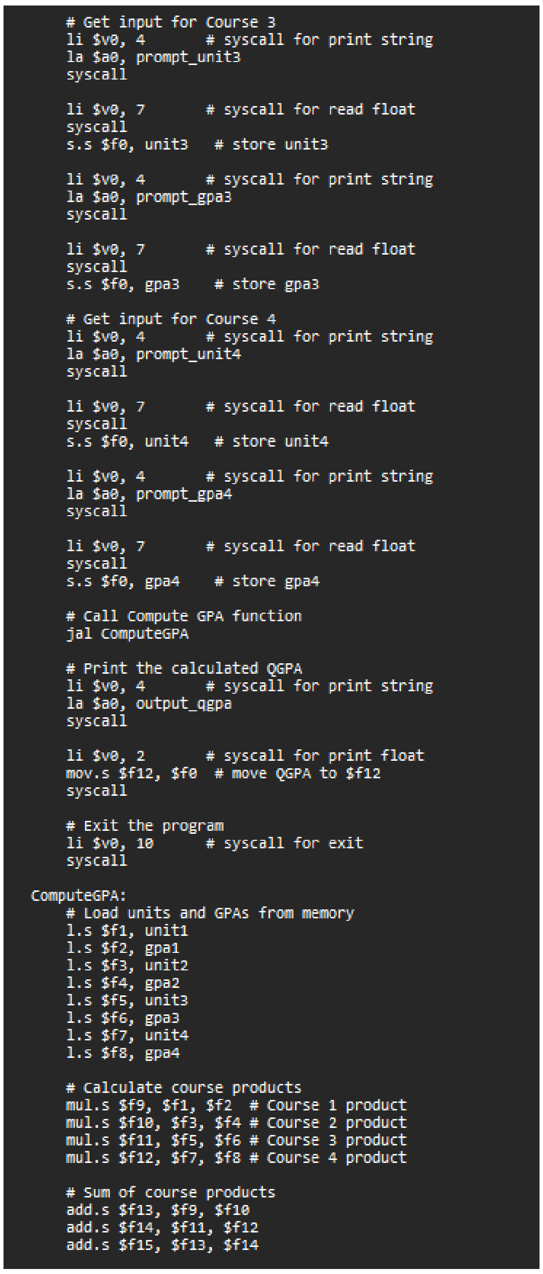
Task 2:

This exercise will familiarize you with floating point multiplication and division instructions. In this part you must write a complete 'UET Peshawar GPA calculator" program. The program should calculate GPA for a quarter only. When your program starts it should ask the unit and GPA in each of the four courses taken in the quarter. It should store all this information in memory. It should then compute the GPA for the person and display it to the user. You should have a separate **Compute GPA** function which loads all necessary info from memory and computes the GPA. Note that you will need syscalls to input and output floats to the user. For this purpose, you will have to refer to the old handout for the service code of these system calls. A more detailed instruction set is given at the end of this handout and may be useful for this exercise.

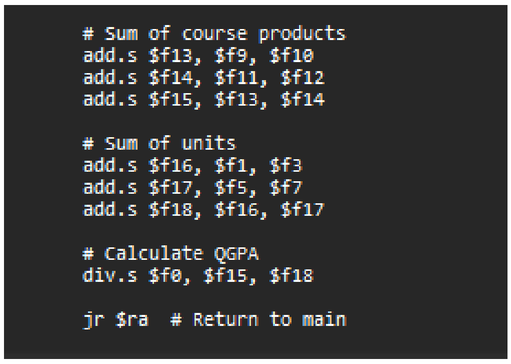
Code:



```
# Name of Programmer- Hassan Zaib Jadoon ; GitHub @hzjadoon
.data
unit1: .float 0.0
gpa1: .float 0.0
unit2: .float 0.0
gpa2: .float 0.0
unit3: .float 0.0
gpa3: .float 0.0
unit4: .float 0.0
gpa4: .float 0.0
qGPA: .float 0.0

# ASCII messages
prompt_unit1: .asciiz "Enter units for Course 1: "
prompt_gpa1: .asciiz "Enter GPA for Course 1: "
prompt_unit2: .asciiz "Enter units for Course 2: "
prompt_gpa2: .asciiz "Enter GPA for Course 2: "
prompt_unit3: .asciiz "Enter units for Course 3: "
prompt_gpa3: .asciiz "Enter GPA for Course 3: "
prompt_unit4: .asciiz "Enter units for Course 4: "
prompt_gpa4: .asciiz "Enter GPA for Course 4: "
output_qgpa: .asciiz "Your calculated QGPA is: "

.text
.globl main

main:
    # Get input for Course 1
    li $v0, 4       # syscall for print string
    la $a0, prompt_unit1
    syscall

    li $v0, 7       # syscall for read float
    syscall
    s.s $f0, unit1   # store unit1

    li $v0, 4       # syscall for print string
    la $a0, prompt_gpa1
    syscall

    li $v0, 7       # syscall for read float
    syscall
    s.s $f0, gpa1    # store gpa1

    # Get input for Course 2
    li $v0, 4       # syscall for print string
    la $a0, prompt_unit2
    syscall

    li $v0, 7       # syscall for read float
    syscall
    s.s $f0, unit2   # store unit2

    li $v0, 4       # syscall for print string
    la $a0, prompt_gpa2
    syscall

    li $v0, 7       # syscall for read float
    syscall
    s.s $f0, gpa2    # store gpa2
```

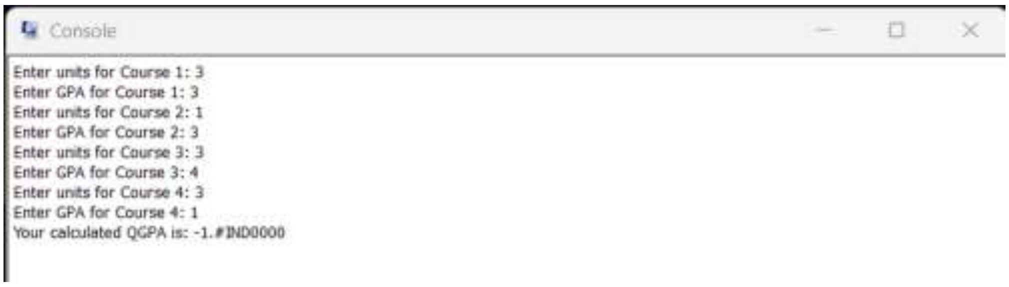


```
    # Get input for Course 3
    li $v0, 4       # syscall for print string
    la $a0, prompt_unit3
    syscall

    li $v0, 7       # syscall for read float
    syscall
    s.s $f0, unit3   # store unit3

    li $v0, 4       # syscall for print string
    la $a0, prompt_gpa3
    syscall

    li $v0, 7       # syscall for read float
    syscall
    s.s $f0, gpa3    # store gpa3

    # Get input for Course 4
    li $v0, 4       # syscall for print string
    la $a0, prompt_unit4
    syscall

    li $v0, 7       # syscall for read float
    syscall
    s.s $f0, unit4   # store unit4

    li $v0, 4       # syscall for print string
    la $a0, prompt_gpa4
    syscall

    li $v0, 7       # syscall for read float
    syscall
    s.s $f0, gpa4    # store gpa4

    # Call Compute GPA function
    jal ComputeGPA

    # Print the calculated QGPA
    li $v0, 4       # syscall for print string
    la $a0, output_qgpa
    syscall

    li $v0, 2       # syscall for print float
    mov.s $f12, $f0  # move QGPA to $f12
    syscall

    # Exit the program
    li $v0, 10      # syscall for exit
    syscall

ComputeGPA:
    # Load units and GPAs from memory
    l.s $f1, unit1
    l.s $f2, gpa1
    l.s $f3, unit2
    l.s $f4, gpa2
    l.s $f5, unit3
    l.s $f6, gpa3
    l.s $f7, unit4
    l.s $f8, gpa4

    # Calculate course products
    mul.s $f9, $f1, $f2  # Course 1 product
    mul.s $f10, $f3, $f4 # Course 2 product
    mul.s $f11, $f5, $f6 # Course 3 product
    mul.s $f12, $f7, $f8 # Course 4 product

    # Sum of course products
    add.s $f13, $f9, $f10
    add.s $f14, $f11, $f12
    add.s $f15, $f13, $f14
```



```
    # Sum of course products
    add.s $f13, $f9, $f10
    add.s $f14, $f11, $f12
    add.s $f15, $f13, $f14

    # Sum of units
    add.s $f16, $f1, $f3
    add.s $f17, $f5, $f7
    add.s $f18, $f16, $f17

    # Calculate QGPA
    div.s $f0, $f15, $f18

    jr $ra  # Return to main
```

Output:

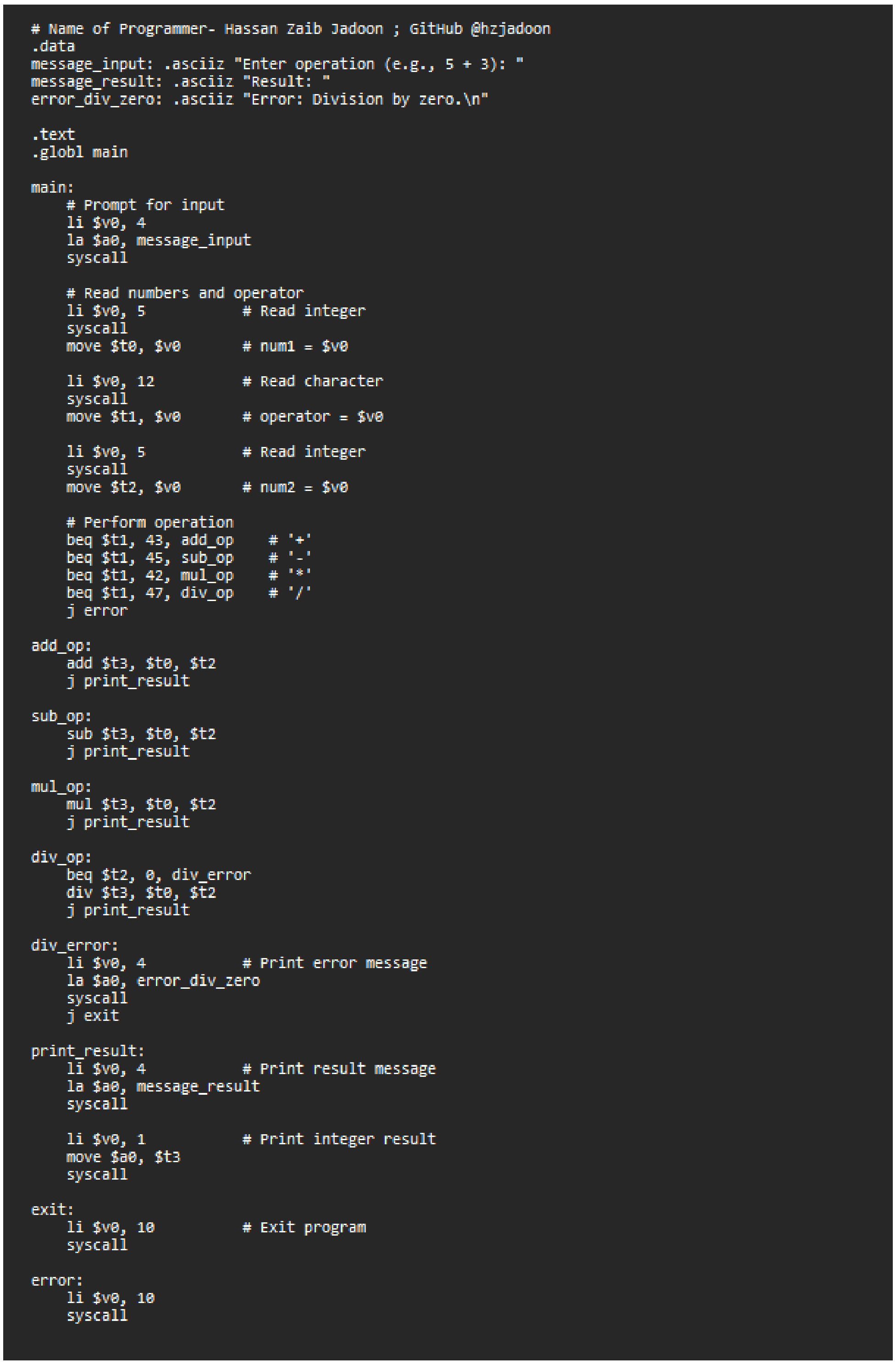


```
Enter units for Course 1: 3
Enter GPA for Course 1: 3
Enter units for Course 2: 1
Enter GPA for Course 2: 3
Enter units for Course 3: 3
Enter GPA for Course 3: 4
Enter units for Course 4: 3
Enter GPA for Course 4: 1
Your calculated QGPA is: -1.#IND0000
```

Task 3

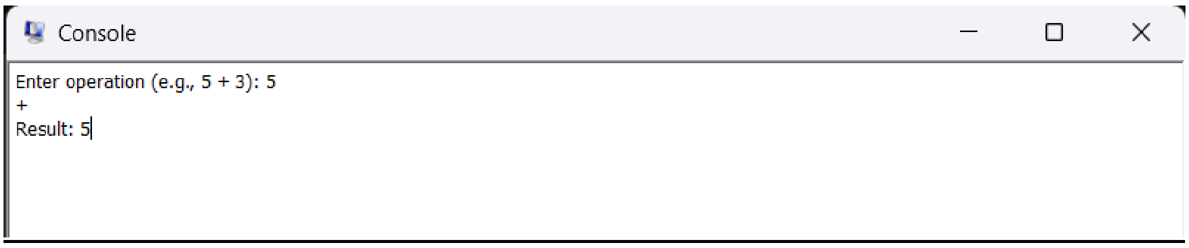
Design a calculator that can perform addition, subtraction, multiplication and division on integer as well as floating point numbers.

Code:



```
# Name of Programmer- Hassan Zaib Jadoon ; GitHub @hzjadoon
.data
message_input: .asciiz "Enter operation (e.g., 5 + 3): "
message_result: .asciiz "Result: "
error_div_zero: .asciiz "Error: Division by zero.\n"

.text
.globl main

main:
    # Prompt for input
    li $v0, 4
    la $a0, message_input
    syscall

    # Read numbers and operator
    li $v0, 5           # Read integer
    syscall
    move $t0, $v0       # num1 = $v0

    li $v0, 12          # Read character
    syscall
    move $t1, $v0       # operator = $v0

    li $v0, 5           # Read integer
    syscall
    move $t2, $v0       # num2 = $v0

    # Perform operation
    beq $t1, 43, add_op    # '+'
    beq $t1, 45, sub_op    # '-'
    beq $t1, 42, mul_op    # '*'
    beq $t1, 47, div_op    # '/'
    j error

add_op:
    add $t3, $t0, $t2
    j print_result

sub_op:
    sub $t3, $t0, $t2
    j print_result

mul_op:
    mul $t3, $t0, $t2
    j print_result

div_op:
    beq $t2, 0, div_error
    div $t3, $t0, $t2
    j print_result

div_error:
    li $v0, 4           # Print error message
    la $a0, error_div_zero
    syscall
    j exit

print_result:
    li $v0, 4           # Print result message
    la $a0, message_result
    syscall

    li $v0, 1           # Print integer result
    move $a0, $t3
    syscall

exit:
    li $v0, 10          # Exit program
    syscall

error:
    li $v0, 10
    syscall
```

Output:



```
Enter operation (e.g., 5 + 3): 5
+
Result: 5
```