Scheme Introduction

Hello World (but a little more than that)

1. Create the file SchemeIntro1.scm using Emacs. Inside that file, create Scheme code that will display ALL members of your team on separate lines.

Simple Math!

2. Create the file SchemeIntro2.scm using Emacs. Create the code to gather two float values a and b from the user and compute the fucntion f(a,b) = 3*a+4*b. Display the function and the results. For example, if a is 1 and b is 2, you should output: “f(1,2) = 11”. For this problem, you are allowed to create variables using define.

Simple Math, with no variables!

3. Create the file SchemeIntro3.scm. Create the code to gather a number from the user and compare it with 10. If the number is larger than 10, output #t, else output #f. HINT: Find “if conditions” at the end of the Introduction to Scheme notes.

Debugging and Reading - your favorite thing to do ;)

4. Create the file SchemeIntro4.scm. Find the Gambit (Scheme) Manual. Gambit has some amazing prebuilt debugging functions. Please review “un-trace” (and really trace). The example code is below. Enter the code below into SchemeIntro4.scm and run.

```
(define (fact n) (if (< n 2) 1 (* n (fact (- n 1)))))
(trace fact)
(fact 5)
```

With the given knowledge above, change the code so that it now can complete a Fibonacci number and trace the result. The new function should accept ONE parameter (n) to determine the Fibonacci value returned.

The first 21 Fibonacci numbers F_n for n = 0, 1, 2, …, 10 are:

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
0	1	1	2	3	5	8	13	21	34	55