Université d'Ottawa
Faculté de génie

School of Electrical Engineering and Computer Science

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

École de science informatique et de génie électrique

CSI2120 Programming Paradigms

FINAL EVALUATION EXERCISE

Length of Examination: 3 hrs **April 11, 2020, 14:00-17:00**

Professor: Jochen Lang

Note: This is just an exercise which is set up to look as close as possible to the final evaluation. You can submit your solution on brightspace but it will not be corrected or marked.

You must abide by and have acknowledged the declaration of integrity.

You must upload your solutions as three source code files with **comments** to Brightspace. If you experience any technical difficulties, you must notify the Professor immediately.

Question	Marks	Out of
1		13
2		13
3		12
Total		38 marks

Question 1 Prolog

Please note:

- *You must comment your code.*

a) Occurrence

Write a predicate numOccur in Prolog which is true if the number E occurs R times in the list. For example:

```
?- numOccur(4, [1, 7, 5, 0, 4, 1, 4, 6], R).
R = 2.
?- numOccur(4, [1, 7, 5, 0, 8, 1, 8, 6], R).
R = 0.
?- numOccur(4, [ ], R).
R = 0.
```

b) Flipping

Write a predicate flip. Flip is to exchange the order of pairs itself and in the list.

Example:

```
?- flip( [ (a, b), ( c, d)], L )
L = [ (d,c), (b,a) ].

?- flip( [ (a, b), ( c, d), (e, f), (g, h), (1, 2)], L ).
L = [(2, 1), (h, g), (f, e), (d, c), (b, a)].

?- flip( [], L ).
L = [].

?- flip( [(1,2)], L ).
L = [(2, 1)].
```

Question 2 Scheme

Please note:

- *You are not allowed to use set! in answering this question.*
- *You must comment your code*

a) Occurrence

Write a function numOccur in Scheme which has arguments a number E and a list L and returns how many times the number E occurs in the list L. For example:

```
(numOccur 4 '(1 7 5 0 4 1 4 6))
  ⇒ 2
(numOccur 4 '(1 7 5 0 8 1 8 6))
  ⇒ 0
(numOccur 4 '())
  ⇒ 0
```

b) Frequency

Write a Scheme function that accepts a list of numbers L and returns the frequency (number of times) of each number occurs in the list. The frequency is to be returned in a list of lists where each list contains the number as `car` and its frequency as the second element in the list as in the following examples:

```
(frequency  `())
  ⇒ ()
(frequency  `(1  5  2  7  1  6  1  6  4))
  ⇒ ((1  3)  (2  1)  (4  1)  (5  1)  (6  2)  (7  1))

(frequency  `(1  5  9  7 -1))
  ⇒ ((-1  1) (1  1)  (5  1)  (7  1)  (9  1))
```

Question 3 Go

a) Methods

Complete the following program (also available as source file) by supplying the function `printMeal` and completing the main routine to produce the output (exactly as shown):

```
Main: Schnitzel at 15.50
Desert: Pumpkin Pie at 5.60
Total: 21.10
```

Note: Do not fill the gaps below but submit your solution as golang source file.

```
package main

import "fmt"

type Desert struct {
     Name      string
     Price     float32
}

type MainCourse struct {
     Name  string
     Price float32
}

type Meal struct {
     MainCourse
     Dessert
     Total float32
}


func main() {

     m := Meal{ _____________________________________________}

     // Calculate the total price of the main course plus desert
     m.Total =  _____________________________________________

     m.printMeal()
}
```

b) Go Routines and Channels

Complete the go routine in the lambda below. The go routine is to send the numbers in the array on the channel and then close the channel.

```
package main

import "fmt"

func main() {
     numbers := []int{216, 218, 221, 260}
     ch := make(chan int)

     // Your solution
     // You will need to insert code in the source file and not
     // below.

          ____________________________________________________________


          ____________________________________________________________


     for {
          if num, ok := <-ch; !ok {
               fmt.Println("Channel closed")
               break
          } else {
               fmt.Println(num)
          }
     }
}
```