

Université d'Ottawa
Faculté de génie

School of Electrical
Engineering and
Computer Science



University of Ottawa
Faculty of Engineering

École de science
informatique et de
génie électrique

CSI2120 Programming Paradigms

MIDTERM EXAM

Length of Examination: 75 minutes

March 5, 2019, 16:00

Professor: Jochen Lang

Page 1 of 9

Family Name: _____

Other Names: _____

Student Number: _____

Signature _____

You are allowed one single-sided, letter-sized, hand-written sheet of notes.

At the end of the exam, when time is up: Stop working and close your exam booklet. Remain silent.

Question	Marks	Out of
1		10
2		4
3		6
4		6
Total		26

Question 1 [10 marks]

Given the following definitions of the structures `shape`, `triangle` and `square` in Go:

```
package main

import (
    "fmt"
    "math"
)

type Point struct {
    X float64
    Y float64
}

func Distance(p1 Point, p2 Point) (distance float64) {
    distance = math.Sqrt(math.Pow(p1.X-p2.X, 2.0)
+ math.Pow(p1.Y-p2.Y, 2.0))
    return
}

type Shape interface {
    Perimeter() float64
    Show()
}

type shape struct {
    Name      string
    Vertices []Point
}

type square struct {
    shape
}

type triangle struct {
    shape
}
```

- a) Write the methods required such that the type `triangle` will satisfy the interface `Shape`. *Note that `Show` is to simply print the Name string of `triangle`.*

```
_____ Show() {  
  
    fmt.Println(_____  
  
    _____  
  
}  
  
_____ Perimeter() _____ {  
  
    var perimeter float64 = 0.0  
    for i := 0; i < 3; i++ {  
        perimeter += Distance(t.Vertices[i], t.Vertices[(i+1)%3])  
    }  
  
    _____  
  
}
```

b) The following code creates 2 squares and 2 triangles.

```
s1 := NewSquare()  
s2 := NewSquare()  
t1 := NewTriangle()  
t2 := NewTriangle()
```

Given `s1`, `s2`, `t1` and `t2` and knowing that both types, `square` and `triangle`, satisfy the interface `Shape`,

- initialize an **array** such that it contains these 4 shapes, and
- compute the sum of all perimeters in a single loop.

Question 2 [4 marks]

The main function below creates structures of type `Triangle` and passes them to the processor function. The triangles are sent over the channel `shapeChannel`. Once all triangles have been created, the function closes the channel and waits for the processor to finish processing all triangles.

```
func main() {
    shapeChannel := make(chan Triangle, 3)
    done := make(chan bool)

    go processor(shapeChannel, done)

    for j := 1; j <= 10; j++ {
        shapeChannel <- NewTriangle(j*j)
    }

    close(shapeChannel)

    <-done
}
```

Complete the processor function below so that it is compatible with the main function above:

```
func processor(_____) {
    for {
        triangle, more := _____
        if more {
            Process(triangle)
        } else {
            fmt.Println("End of processing")
            _____
            return
        }
    }
}
```

Question 3 [6 marks]

Given the following Prolog program

```
weather(X,Y) :- snow(X),
                rain(X),
                Y=snow_and_rain.

weather(X,Y) :- cloudy(X),
                rain(X),
                Y=cloudy_and_rainy.

weather(X,Y) :- sun(X),
                cloudy(X),
                Y=mix_of_sun_and_cloud.

weather(X,Y) :- cloudy(X),
                Y=gray.

weather(X,Y) :- sun(X),
                Y=sunny.

cloudy(monday).
cloudy(tuesday).
cloudy(thursday).
cloudy(friday).
snow(thursday).
sun(wednesday).
sun(friday).
rain(tuesday).
rain(thursday).
```

- a) What is the **first** solution found by the following query?

```
?- weather(X,Y).
```

- b) Give all solutions in order that they are found by Prolog by the following query (and using ; after each answer)?

```
?- weather(X,gray) .
```

- c) Give all solutions in order that they are found by Prolog by the following query (and using ; after each answer)?

```
?- weather(thursday, Y) .
```

Question 4 [6 marks]

Given the following Prolog program:

```
unionList([],B,B) .

unionList([E|A],B,D) :-
    member(E,B) ,
    unionList(A,B,D) .

unionList([E|A],B,[E|D]) :-
    \+member(E,B) ,
    unionList(A,B,D) .
```

For example:

```
?- unionList([1,2,3,7],[3,4,5],L) .
L = [1, 2, 7, 3, 4, 5] .
```

But what will be the result if the lists contain duplicate elements?

a) Give the list obtained by the following request:

```
?- unionList([1,3,5,9,1],[3,4,1,3,1],L) .
```

L = [_____]

b) Give the list obtained by the following request:

```
?- unionList([1,3,3,6],[2,3,3,3,7],L) .
```

L = [_____]

Suppose we want to remove duplicates from a list. For example:

```
?- remove_dup([1,2,3,2,2,3],L).
```

```
L = [1, 2, 3]
```

c) Complete the predicate below correspondingly.

```
remove_dup([], []).
```

```
remove_dup([E|A],L):- member(E,A), remove_dup(A,L).
```

```
remove_dup([E|A],_____):-
```

```
\+ member(E,A),
```

```
remove_dup(_____).
```