Started on	Tuesday, 20 April 2021, 2:03 PM
State	Finished
Completed on	Tuesday, 20 April 2021, 5:13 PM
Time taken	3 hours 9 mins

Question 1

Complete

Marked out of 20.00

https://eclass.srv.ualberta.ca/pluginfile.php/6714317/question/questiontext/8504650/1/16986497/f-q1-v2.pdf

x >= y + 1, so x domain is reduced to
Dx = {2, 3}
Dy = {1, 2}

z < x so z domain is reduced to
Dz = {2}
and x is reduced to
Dx = {3}
y is still good

w = z + 1
so Dw is reduced to
Dw = {3}

Question 2

Complete

Marked out of 15.00

https://eclass.srv.ualberta.ca/pluginfile.php/6714317/question/questiontext/8504650/2/16986503/f-q2-v2.pdf

{ x->3, y-> [e1, CT0] } U CT0

context for (+ x z)

(3 \x.(+xz)) -> (3 (+3 3)) -> (3 6)
continuing from above,
eval (+ x z) in context
{x -> 3, z -> 3} U CT0

iii SECD
(LDF e' || (RTN)),
where e' is :

(LDC 3 LDF (LD (2.1) LD (2.1) * LDC 2 + RTN)

Question 3

Complete

Marked out of 15.00

https://eclass.srv.ualberta.ca/pluginfile.php/6714317/question/questiontext/8504650/3/16986509/f-q3-lisp-v1.pdf?time=1618789736697

```
    (+ (car L) (sum (cdr L)))))

ii

(defun exam1 (L)
  (remove-duplicate (reduce (lambda (x y) (append x y)) L)))

(defun remove-duplicate (X)
  (cond
   ((not X) nil)
   ((xmember (car X) (cdr X)) (remove-duplicates (cdr X)))
   (t (cons (car X) (remove-duplicates (cdr X))))))
```

Question 4

Complete

Marked out of 35.00

https://eclass.srv.ualberta.ca/pluginfile.php/6714317/question/questiontext/8504650/4/16986513/f-q4-Prolog-clp-v1.pdf

```
sum(L, PartialSum),
APlusB #= A + B,
Sum #= PartialSum + APlusB.

happens_once([], Count),
Count #= 1.
happens_once([H | T], Count):-
H #= 1,
Count #< 1,
happens_once(T, CountPlus1),
CountPlus1 #= Count + 1.

fronts_distinct(Mtr) :-
maplist(X, [X | _], Mtr, L), % map rows to just the first element in each row
all_distinct(L). % all the first elements are now distinct
```

Question 5

Complete

Marked out of 15.00

https://eclass.srv.ualberta.ca/pluginfile.php/6714317/question/questiontext/8504650/5/16986520/f-q5-ASP-v2.pdf

This is the program file mentioned in the problem.

```
% if c is in it then either d of f but not both
:- member(c, Set), member(d, Set), not member(f, Set).
:- member(c, Set), member(f, Set), not member(d, Set).

b)
%a ferry cannot be at 2 different locations at the same time
:- at(ferry, Loc, T), at(ferry, Loc1, T), Loc != Loc1.

% same car cannot be in 2 different ferries at the same time
:- in(Ferry1, Car, T), in(Ferry2, Car, T), Ferry1 != Ferry2.

% same car cannot board in different locations at the same time
:- board(Car, Loc1, T), board(Car, Loc2, T), Loc1 != Loc2.
```