COMP3007A (Fall 2018) – "Programming Paradigms"

Quiz 2 of 2 **Practice Quiz**

This is a closed book exam. No calculators, cellphones, laptops, or other aids are permitted. Answer every question in the space that has been provided. You must show all your work without skipping steps; correct answers that are presented without justification may receive a mark of zero.

Student Name	______________________
Student Number	______________________

1. [HASKELL] Recreate the following function such that it is tail-call optimized. Do not forget to include a helper function (with type declaration `[a] -> [a]`) and a new type declaration for your tail-call optimized function.

```
foo :: [a] -> [a]
foo [] = []
foo (h:t) = (bar h) : (foo t)
```

2. [HASKELL] Since following data type can be used to represent a list of integers…

```
data ListOfInts = EmptyList | Single Int | Cons Int Int ListOfInts
```

…how could you write a function that computes the arithmetic mean (i.e., sum of all the elements divided by the number of elements) of a list encoded using this type? The type declaration has been provided below, but you will likely want to implement your own `sum` and `length` helper functions. Please assume that the arithmetic mean of a list of length 0 is defined as 0.

```
avrage :: ListOfInts -> Float
```

This is a closed book exam. No calculators, cellphones, laptops, or other aids are permitted. Answer every question in the space that has been provided. You must show all your work without skipping steps; correct answers that are presented without justification may receive a mark of zero.

Student Name ____

Student Number ____

3. [HASKELL] Prove that `(calcProduct n == 0) = anyZero n` by using structural induction. Use the following implementations for `calcProduct`, and `anyZero` and refer to the individual lines in these implementations using the labels `P1`, `P2`, `Z1`, `Z2A`, and `Z2B`. You are expected to follow the process of structural induction as it was demonstrated in class and you must show all your work (including the line applied during each step of your equational reasoning).

```
        calcProduct :: [Integer] -> Integer
P1      calcProduct [] = 1
P2      calcProduct (h:t) = h * (calcProduct t)

        anyZero :: [Integer] -> Bool
Z1      anyZero [] = False
        anyZero (h:t)
Z2A       | h == 0 = True
Z2B       | otherwise = anyZero t
```

COMP3007A (Fall 2018) – "Programming Paradigms"

Quiz 2 of 2 **Practice Quiz**

This is a closed book exam. No calculators, cellphones, laptops, or other aids are permitted. Answer every question in the space that has been provided. You must show all your work without skipping steps; correct answers that are presented without justification may receive a mark of zero.

Student Name ____

Student Number ____

4. [HASKELL] In your second assignment, you needed to create an algebraic data type (such as the one below) for working with three-valued logic:

```
data TLogicVal = Duno | Troo | Falz deriving Show
```

Consider, as an alternative, how a `(Maybe Bool)` could be used as an alternative to a `TLogicVal` value (using `Nothing` to represent the unknown value), and rewrite the following function…

```
tImplication:: TLogicVal -> TLogicVal -> TLogicVal
tImplication Troo Falz = Falz
tImplication Falz _ = Troo
tImplication _ Troo = Troo
tImplication _ _ = Duno
```

…such that it has type declaration

```
tImplication :: (Maybe TLogicVal) -> (Maybe TLogicVal) -> (Maybe TLogicVal).
```

5. [HASKELL] How would you write a single Haskell function to provide the sum of every even number that appears in a list of integers? You may use the built-in `even (even :: Int -> Bool)` function to complete this question but you cannot use any other functions.

COMP3007A (Fall 2018) – "Programming Paradigms"

Quiz 2 of 2 **Practice Quiz**

This is a closed book exam. No calculators, cellphones, laptops, or other aids are permitted. Answer every question in the space that has been provided. You must show all your work without skipping steps; correct answers that are presented without justification may receive a mark of zero.

Student Name ______________________________

Student Number ______________________________

6. [HASKELL] How would you write a single expression using the higher order functions `foldr`, `map`, and/or `filter` to provide the sum of every even number that appears in a list of integers? You may use the built-in `even (even :: Int -> Bool)` function to complete this question but you cannot use any other functions.

7. [PROLOG] Assuming that the user continues pressing ";" as long as possible (thereby forcing Prolog to identify all possible solutions), what is the exact output provided by the following Prolog program in response to the query `a(X)`?

```
a(1).
a(2).
a(X) :- b(X), (c(X) ; d(X)).
a(X) :- e(X).
b(X) :- e(X), f(X).
b(1).
c(1).
d(1).
d(3).
e(2).
e(3).
f(3).
```

```
Write the Exact Output Here
```