CS 1101 A Term 2020 Exam 3 (100 points)

Pledge: ________ (2)

Question 1: ________ (18)

Question 2: ________ (20)

Question 3: ________ (20)

Question 4a: ________ (15)

Question 4b: ________ (15)

Question 5a: ________ (10)

Question 5b: ________ (5 Bonus)

TOTAL: ________ (100 or 105 with Bonus)

You have 70 minutes to complete this exam–this includes an extra bit of time for you to make your ultimate submission to Canvas. You do not need to show templates, but you may receive partial credit if you do. You also do not need to show test cases or examples of data definitions except in those cases where they are specifically requested, but you are free to develop them if they will help you write the desired programs.

Your programs may contain only the following BSL/ISL/ASL/Racket constructs:

define define-struct cond else if local begin max min

empty? cons? cons first rest list append length

+ − ∗ / = < > <= >= **and or not**

string=? string-length string-append substring add1 sub1 error format

predicates for any defined data types

filter map andmap ormap foldr foldl build-list

set! set-structure!

Plus, of course, any operators introduced by **define-struct** (including mutators)

Additionally, you may use any constants you find necessary, for example, empty, true, false, 0, etc.

Pledge (2 points):

I pledge on my honor that I am taking this Exam on my own, with help only from the provided starter file and DrRacket (if I have so chosen) and absolutely no one else, *sans* notes, book closed, search engines idled, Help Desk unconsulted. I further pledge not to discuss the Exam in any way with anyone until Sunday, October 18, 2020 because I understand that there are students with circumstances that may cause them to take the Exam after I have completed it.

"My brain is open. . . ."

a.) `#true`
b.) `#false`

1. Higher-Order Functions (18 points)

a). (5 points) Create `(list 2 4 6 8 10)` with an expression involving higher-order functions.

b). (4 points) Under what circumstances will the lengths of

```
(filter odd? ListOfNum)
```

and

```
(map sqr ListOfNum)
```

be equal, where `ListOfNum` is the same list of numbers in each expression?

c). (4 points) Suppose `(ormap positive? ListOfNum)` returns `false`.

What must `(filter positive? ListOfNum)` return, where `ListOfNum` is the same list of numbers in each expression?

d). (5 points) The higher-order function `foldr` has the following signature:

```
foldr: (X  Y -> Y)  Y  ListOfX  ->  Y
```

Briefly explain what the `(X  Y -> Y)` part represents.

2. Higher-Order Functions on Lists of Structs (20 points)

Nomanisan Island is a bicyclist's dream. The roads and trails are so popular with riders that numerous bike rental shops operate on the Island.

Here are some data definitions:

```
(define-struct bike (company style size renter))
;;a Bike is a (make-bike String String Natural String)
;;interp: represents a bicycle where
;;        company is the name of the rental shop
;;        style is the kind of bike, e.g. "road", "mountain", "tandem", etc.
;;        size is the wheel diameter (in inches)
;;        renter is the name of the person who last rented the bicycle

;;a ListOfBike is one of:
;;  empty
;;  (cons Bike ListOfBike)
```

Using filter and/or map, write a function satisfying the following signature/purpose:

```
;;recent-renters: ListOfBike String Natural -> ListOfString
;;consumes a list of bikes and a style and a wheel size
;;returns a list of the most recent renters of bikes with that style and wheel size
```

Include a signature and purpose for any helper function(s) you write.

3. Accumulator-Style Programming (20 points)

Nomanisan Island is a bicyclist's dream. The roads and trails are so popular with riders that numerous bike rental shops operate on the Island.

Here are some data definitions:

```
(define-struct bike (company style size renter))
;;a Bike is a (make-bike String String Natural String)
;;interp: represents a bicycle where
;;        company is the name of the rental company
;;        style is the kind of bike, e.g. "road", "mountain", "tandem", etc.
;;        size is the wheel diameter (in inches)
;;        renter is the name of the person who last rented the bicycle

;;a ListOfBike is one of:
;;  empty
;;  (cons Bike ListOfBike)
```

Using accumulator-style recursion, write a function satisfying the following signature/purpose:

```
;;recent-renters: ListOfBike String Natural -> ListOfString
;;consumes a list of bikes and a style and a wheel size
;;returns a list of the most recent renters of bikes with that style and wheel size
```

Include a signature and purpose for any helper function(s) you write.

4a. Mutable Variables (15 points)

Tourism on Nomanisan Island is a booming industry. With many flights arriving and departing John Donne Continental Airport, the air travel industry has a large impact on the vibrant local economy.

```
(define-struct flight (airline plane seats price))
;;a Flight is a (make-flight String String Natural Natural)
;;interp: represents an airline flight to Nomanisan Island where
;;        airline is the name of the airline
;;        plane is the kind of airplane flown
;;        seats is the number of open seats
;;        price is the cost of a ticket in dollars

;;a ListOfFlight is one of:
;;  empty
;;  (cons Flight ListOfFlight)

;;All daily flights to Nomanisan Island are contained in the list NIFLIGHTS
```

The weather around Nomanisan Island can be a bit fickle. A few times each year, when the So'westers blow, certain flights must be grounded based on the planes involved.

Using `set!` write a function that satisfies the following signature/purpose:

```
;;ground-plane:  String -> ListOfFlight
;;consumes a plane (in string form)
;;deletes all flights of the specified plane from NIFLIGHTS
;;EFFECT: Possibly removes some flights from NIFLIGHTS
```

Include a signature and purpose for any helper function(s) you write.

4b. Mutable Variables (15 points)

Tourism on Nomanisan Island is a booming industry. With many flights arriving and departing John Donne Continental Airport, the air travel industry has a large impact on the vibrant local economy.

```
(define-struct flight (airline plane seats price))
;;a Flight is a (make-flight String String Natural Natural)
;;interp: represents an airline flight to Nomanisan Island where
;;        airline is the name of the airline
;;        plane is the kind of airplane flown
;;        seats is the number of open seats
;;        price is the cost of a ticket in dollars

;;a ListOfFlight is one of:
;;  empty
;;  (cons Flight ListOfFlight)

;;All daily flights to Nomanisan Island are contained in the list NIFLIGHTS
```

Because Nomanisan Island is such a popular destination for daytrippers, the local airlines routinely double their prices on nearly-full flights the night before departure.

Write a function that satisfies the following signature/purpose:

```
;;midnight-run: -> ListOfFlight
;;doubles all ticket prices on flights with fewer than 10 open seats
;;returns error "All Flights Wide Open" if no flight has fewer than 10 open seats
;;EFFECT: The prices of some flights in NIFLIGHTS may be raised
```

Include a signature and purpose for any helper function(s) you write.

5. Data With Cycles (10 points)
Here are some data definitions:

```
(define-struct humanoid (name galaxy siblings))
;;A Humanoid is a (make-humanoid String String ListofHumanoid)
;;interp: represents a humanoid/person where
;;        name is the humanoid's name
;;        galaxy is the humanoid's galaxy of residence
;;        siblings is a list of the humanoid's siblings

;;a ListofHumanoid is one of:
;;  empty
;;  (cons Humanoid ListOfHumanoid)
```

SPOILER ALERT! A long time ago, in the galaxy Farfaraway, there lived a farmer by the name of Luke Skywalker and a princess named Leia Organa. In due course, after many swashbuckling adventures and feats of derring-do, they came to understand that they were actually twins.

Write a sequence of expressions that correctly–and *mutually*–defines `LUKE` and `LEIA` as Humanoids under the data definition above.

BONUS: Data With Cycles (5 points)
Here are some data definitions:

```
(define-struct personage (name birthday mother father))
;;A Personage is a (make-personage String String Ancestor Ancestor)
;;interp: represents a person where
;;        name is the person's name
;;        birthday is the person's date of birth
;;        mother is the person's mother, possibly "unknown"
;;        father is the person's father, possibly "unknown"

;;an Ancestor is one of:
;;  "unknown"
;;  Personage
```

b). (5 Points) SPOILER ALERT!^{n} In Robert Heinlein's celebrated 1960 short story " '—All You Zombies—' ", the narrator–Jane, born on September 20, 1945–is incrementally revealed to be both his (?!) own mother and her (!?) own father.[1]

Write a sequence of expressions that correctly defines `JANE` as a Personage in terms of herself/himself *reflecting his/her unique parentage*, or explain why such a definition is impossible.

[1] https://en.wikipedia.org/wiki/All_You_Zombies