C211: Spring 2016
Test 1

Name: _____

Username: _____

Lab Section: _____ Lecture Team: _____

1. Assume the following data definition.

```
A WeatherCondition is one of
- "Rainy"
- "Snowy"
- "Sunny"
- "Foggy"
- "Cloudy"
```

It's a good idea to carry an umbrella on rainy and cloudy days. In this problem, you will *design* a predicate `carry-umbrella?` that takes a WeatherCondition and returns `#true` if it's umbrella weather, and `#false` otherwise.

(a) Complete the following tests:

```
(check-expect (carry-umbrella? _____ ) #false)

(check-satisfied "Rainy" _____ )
```

(b) In the space below, write the contract, the purpose statement, the definition of `carry-umbrella?`.

2. The United States uses the Fahrenheit (F) system to measure temperature. In other parts of the world, Celsius (C) is used. In this problem, you will *design* a function `F->C` that takes a temperature in degrees Fahrenheit and returns the corresponding temperature in degrees Celsius. Here are two examples:

```
> (F->C 68)
20
> (F->C -459.67)
-273.15
```

   (a) Write test cases corresponding to the examples shown above.

   (b) Write the contract for the `F->C` function.

   (c) To convert a temperature `t` from Fahrenheit to Celsius, subtract `32` from `t` and then multiply the result by `5/9`. Write the definition of the `F->C` function. (Do not write a purpose statement.)

3. Assume the following data definitions.

```
A DegreeScale is one of
- 'F
- 'C

A Temperature is an Int in the range (-50..120]

A Forecast is a (make-forecast Temperature WeatherCondition)
```

(a) Complete the following contract and definition of a type predicate for DegreeScale.

```
; degree-scale?  :  ____________________ -> ____________________

(define (degree-scale?  x)

   (or ____________________________________

       ____________________________________ ))
```

(b) Complete the following contract and definition of a type predicate for Temperature.

```
; temperature?  :  ____________________ -> ____________________

(define (temperature?  x)

   (and ___________________________________

        ___________________________________

        ___________________________________ ))
```

(c) Define a structure named `forecast` for the Forecast type. Use `high` and `outlook` as the field names.

(d) Write the contract for each function that is created by the definition you made in Part (c).

4. Assume that the temperature in a Forecast is always given in degrees Fahrenheit. In this problem, you will *design* a function `forecast->string` that takes a forecast and a degree scale, and returns a string corresponding to the forecast (using the format shown in the tests below).

   (a) Complete the following tests.

```
(check-expect ____________________________________________________________
              "Snowy, with a high of 30 F.")

(check-expect ____________________________________________________________
              "Foggy, with a high of 20 C.")
```

   (b) Write the contract for `forecast->string`.

   (c) Define the `forecast->string` function. Use `error` in the case that the degree scale is neither `'F` nor `'C`. (Do not write a purpose statement.)

   (d) Write a test to check your error condition.