CS 359: Programming Paradigms
PRELIM 3: Symbolic Differentiation, or
“First Semester Calculus in One Easy PEX”
Due Date: Start of Class Lesson 27
(30 points)

Help Policy:

AUTHORIZED RESOURCES: Any, except another cadet’s program.

NOTE:

- Never copy another person’s work and submit it as your own.
- Do not jointly create a program.
- You must document all help received from sources other than your instructor.
- DFCS will recommend a course grade of F for any cadet who egregiously violates this Help Policy or contributes to a violation by others.

Documentation Policy:

- You must document all help received from any source other than your instructor.
- The documentation statement must explicitly describe WHAT assistance was provided, WHERE on the assignment the assistance was provided, and WHO provided the assistance.
- If no help was received on this assignment, the documentation statement must state “NONE.”

Vague documentation statements must be corrected before the assignment will be graded, and may result in a 5% deduction on the assignment.

1. OBJECTIVES

- Be able to create, test, and debug basic functions in F#
- Develop the first steps toward a symbolic differentiation capability
- NOTE THAT LATE PRELIMS ARE NOT ACCEPTED WITHOUT EXTENUTATING CIRCUMSTANCES. Please coordinate with your instructor EARLY if you require an extension.

2. INSTRUCTIONS

a) Read the PEX3 assignment document thoroughly. Develop a set of questions whose answers would make you better able to complete PEX3. Ask those questions in class.

b) Download the PEX3 Template file available from the course website.

c) Implement the following functions (16 points total)

   a. (2 points apiece) Make functions: make-sum, make-difference, make-product, make-quotient.

   b. (4 points) toInfixString

   c. (4 points) replace

d) Answer the following question sets to get you thinking. If you cannot complete the exercises in question set 1, you are NOT ready to begin the PEX. Question set 2 is preparation for the GR and assessment, not the PEX.

3. QUESTION SET 1 (4 POINTS)

1) Represent the following expression as a discriminated union of the type `expression` as defined in your PEX.

$$6x^2 + x + 3$$

2) Represent the following discriminated union as an algebraic expression. The discriminated union conforms to the definition of the `expression` type found in your PEX.

```
Divide(Subtract(Variable "x", Number 6.0),
       Subtract(Variable "y", Number 5.0))
```

3) Differentiate the following. You may express your answer as an algebraic expression (no need to translate to a discriminated union).

   a. $\frac{d}{dx}\left[\frac{d}{dx}(2x^4 + x^3 + 2)\right]$

   b. $\frac{d}{dx}[\tan(x^2)]$

4. QUESTION SET 2 (10 POINTS)

Write an F# functions to solve the following. You may write more than one function to help you solve each one. You have the following restrictions (which are remarkably similar to those on the PEX, the GR, and the in class assessment).

- You may use let to define a function, but NOT in an attempt to declare a variable.
- You may not use while or for
- You may not use arrays
- You may not use the mutable keyword
- You may not use list comprehension

Use MAY use the following functions, but no others from the F# List Library:

- `List.head()`
- `List.tail()`
- `List.append()`

4) Write a function `abString` that takes two arguments `numAs` and `numBs` and returns a string of `numAs "a"'s` followed by `numBs "b".`

```
> abString 4 5;;
val it : string = "aaaabbbbb"
```

5) Write a function `replicateElements` that takes a list `li` and an integer `N` and returns a list with each element in `li` replicated N times.

```
> replicateElements ([5;6;7;8]) 4;;
val it : int list = [5; 5; 5; 5; 6; 6; 6; 6; 7; 7; 7; 7; 8; 8; 8; 8]
```

6) Write a function `removeKthElement` that takes a list li and an integer `K` and returns a list with the Kth element of `li` removed.

```
> removeKthElement ([5;6;7;8]) 3;;
val it : int list = [5; 6; 8]
```

7) Write a function `createListRange` that takes three integers (`lo`, `hi`, and `step`) and returns a list containing all integers between `lo` and `hi` incremented by `step`.

```
> createListRange 1 10 2;;
val it : int list = [1; 3; 5; 7; 9]
```

8) Use partial evaluation to evaluate the following:

a. $[x \to y \to 3x^3 + y + 2](5)$

b. $[y \to y(2x)](z \to 3z)$ Leave your answer in terms of *x*

9) Curry the following function *f* into two functions that each takes a single argument. You may omit the name of the function in your answer. Use the arrow notation → for functions.

$$f(x, y) = \sqrt[x]{y}$$

CS359: PRELIM 3 Cut Sheet

Name:

Grade: /30

Implementation		
Make functions: make-sum, make-difference, make-product, make-quotient		**8**
toInfixString		**4**
replace		**4**
Implementation Subtotal:		**16**
Miscellaneous		
Question set 1		**4**
Question set 2		**10**
Miscellaneous Subtotal:		**14**
Vague/Missing Documentation:	**–**	**-2**
Total with Adjustments:		**30**

NOTE THAT LATE PRELIMS ARE NOT ACCEPTED WITHOUT EXTENUTATING CIRCUMSTANCES. Please coordinate with your instructor EARLY if you require an extension.