COMPSCI 220 Midterm 2: Fall 2019

Name: ______________________________

Spire ID: ______________________________

DO NOT OPEN EXAM BOOKLET UNTIL INSTRUCTED TO DO SO.

Time for Exam 60 minutes

Instructions Do not begin your exam until instructed. Please read all rules carefully before beginning your exam. You will have one hour to complete all exam problems to the best of your ability.

Points This exam has 6 questions worth 100 points total.

Question Difficulty The expected difficulty of each question is marked with ⋆s. Harder questions have more ⋆s. This should help you plan your time.

Rules

1. All electronics (including but not limited to cellphones, tablets, computers, smart watches, and calculators) must be turned off and placed out of sight in your backpack.
2. In order to receive credit for your midterm you *must* write your name and Spire ID on the first page, and your name on all other pages.
3. All answers must fit within the boxes allocated for that question. Any work outside of the box will not be taken into account during grading.
4. There will be no talking or leaving the exam room room during testing.
5. We have provided exam notes, which you may write on. You may not use any other material or notes.
6. All work must be your own and compliant with the Universities Academic Honesty Policy. Any exhibition of Academic Dishonesty will be reported to the Academic Honesty Board.

Version A

Question 1 What do the following programs display? *Note that none of these programs produce errors.*

Part a (5 points) *Difficulty:* ⋆

```
function F1(x) {
    if (x === 5) {
        return;
    }
    console.log(x);
    F1(x + 1);
}
F1(1);
```

Part b (5 points) *Difficulty:* ⋆

```
function F2() {
    console.log(’A’);
    function inner() {
        console.log(’B’);
    }
    console.log(’C’);
}
F2();
```

Part c (5 points) *Difficulty:* ⋆

```
function F3(x) {
    return {
        get: function() { return x; },
        set: function(y) {  x = y; }
    };
}

let A = F3(200);
let B = F3(10);
B.set(300);
console.log(A.get());
```

Part d (5 points) *Difficulty:* ⋆⋆

```
function F4(x) {
    return {
        x: x,
        get: function() { return x; },
        set: function(y) { x = y; }
    };
}
let o = F4(3);
o.set(100);
console.log(o.x);
console.log(o.get());
```

Part e (5 points) *Difficulty:* ⋆⋆

```
function F5(arr) {
    return {
        get: function(index) { return arr[index]; },
        set: function(newArr) { arr = newArr; }
    }
}
let a = [11, 22, 33];
let o = F5(a);
a[0] = 44;
console.log(o.get(0));
o.set([100, 200, 300]);
console.log(a[1]);
```

Question 2 Consider the following function.

```
function F7(f, g, h) {
  f(function(x) {
      if (x > 10) {
          h(function() { console.log(’A’); });
      }
      else {
          g(function() { console.log(’B’); })
      }
  })
}
```

Part a (5 points) *Difficulty:* ⋆⋆

Give values of `f`, `g`, and `h` such that `F7(f, g, h)` displays the following output. **Your answer must not use `console.log` or raise errors**.

```
A
B
```

Part b (5 points) *Difficulty:* ⋆⋆

Give values of `f`, `g`, and `h` such that `F7(f, g, h)` displays the following output. **Your answer must not use `console.log` or raise errors**.

```
B
```

Question 3 (5 points) *Difficulty:* ⋆

Consider the following function.

```
function F9(f, g) {
    f(function(x) { console.log(x); });
}
```

Give values for `f` and `g` such that `F9(f, g)` displays the output:

```
220
```

Your answer must not use `console.log` or raise errors.

Question 4 (5 points) *Difficulty:* ⋆⋆

Consider the following function.

```
function F8(f, r, n) {
    if (n === 0) {
        return r;
    }
    else {
        return F8(f, f(n, r), n - 1);
    }
}
```

Give a value for `f`, such that `F8(f, 1, n)` returns the value $1 \times 2 \times \cdots \times n$. (The function must not display anything. i.e., do not use `console.log`.)

Question 5 This question concerns streams, using the implementation in the exam notes. In your answers, you may assume that the stream is unbounded (i.e., you can ignore `sempty`).

Part a (5 points) *Difficulty:* ⋆

Describe the following stream, or just list the first five values that it produces.

```
nats.filter(n => n % 3 === 0).filter(n => n % 6 === 0)
```

Note that `nats` is the stream of natural numbers (definition is in the exam notes).

Part b (5 points) *Difficulty:* ⋆

Write a stream of the multiples of four, i.e. $0, 4, 8, 12, \cdots$. Your code can call the functions and definitions in the exam notes. You do not have to to reproduce them in your solution.

Part c (5 points) *Difficulty:* ⋆⋆⋆

Write a function called `alt` with the following type:

```
alt<A>(stream1: Stream<A>, stream2: Stream<A>): Stream<A>
```

The result of `alt` should be a new stream that contains all the elements of `stream1` and `stream2` interleaved, starting with the first element of `stream1`.

This is an example of `alt`'s behavior:

```
let positives = snode(1, memo0(() => positives.map(x => x + 1)));
let negatives = snode(-1, memo0(() => negatives.map(x => x - 1)));
alt(positives, negatives) // produces 1, -1, 2, -2, 3, -3, ...
```

Part d (10 points) *Difficulty:* ★★

Write a function called `index` with the following type:

```
index<A>(stream: Stream<A>): Stream<{ index: number, value: A}>
```

This function returns a stream of objects that contains the same values as the input stream. However, each value is accompanied with a number that specifies its index in the stream, starting with zero.

For example, if `stream` produces the following values:

```
100
200
300
...
```

then `index(stream)` should produce the following values:

```
{ index: 0, value: 100 }
{ index: 1, value: 200 }
{ index: 2, value: 300 }
...
```

Part e (10 points) *Difficulty:* ⋆⋆

Write a function called `split` with the following type:

```
split<A>(stream: Stream<A>): { evens:  Stream<A>, odds: Stream<B> }
```

This function returns an object with two streams, where `evens` has alternating elements of `stream`, including the first element, and `odds` has alternating elements of `stream`, excluding the first element.

Hint: Use `index` as a helper function.

Part f (5 points) *Difficulty:* ⋆⋆⋆

Describe the `mystery` stream, or just list the first five values that it produces.

```
let X = snode(0, memo0(() => X.map(n => n + 2)));
let Y = snode(1, memo0(() => Y.map(n => n + 2)));
let mystery = alt(X, Y);
```

Question 6 This question concerns error handling with results, using the implementation in the exam notes.

Part a (5 points) *Difficulty:* ⋆

What does the following program display?

```
function E0(n) {
    if (n === 0) { return new Success(10); }
    else { return new Failure(0); }
}
E0(10).then(function(v) { console.log(v); });
console.log("Done");
```

Part b (5 points) *Difficulty:* ⋆

Consider the following function:

```
function E2(r) {
    r.then(function(n) {
        console.log("A");
    });
}
```

Give a value for `r` such that `E2(r)` displays the following output:

```
A
```

Your answer must not use `console.log` or raise errors.

Part c (5 points) *Difficulty:* ⋆⋆

What does the following program display?

```
function E1(n) {
    if (n === 0) {
        return new Success(0);
    }
    else {
        return E1(n - 1).then(r => { console.log(n); return new Success(r + n) });
    }
}
E1(3);
```

End of exam.