

COMPSCI 220 Fall 2018
Midterm 1
October 11, 2018
Time for Exam: 60 minutes

Name:

Spire ID:

**DO NOT OPEN EXAM BOOKLET
UNTIL INSTRUCTED TO DO SO**

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.

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 every page.
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. You may use any printed or hand written material you wish as long as it is on paper and not a digital document.
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.

Name: _____

Helpful Information:

Reduce Function Call:

```
array.reduce(callback, initialValue)
```

Reduce Function Body:

```
let value = initialValue;
for (let i = 0; i < array.length; ++i) {
  value = callback(value, array[i]);
}
return value;
```

Map Function call:

```
array.map(callback)
```

Map Function Body:

```
let newArray = [];
for (let i = 0; i < array.length; ++i) {
  newArray.push(callback(array[i]));
}
return newArray;
```

Filter Function call:

```
array.filter(callback)
```

Filter Function Body:

```
let newArray = [];
for (let i = 0; i < array.length; ++i) {
  if (callback(array[i])) {
    newArray.push(array[i]);
  }
}
return newArray;
```

Name: _____

Question 1 (30 points): What are the outputs of the following programs?

Part a

```
function foo(x) {  
  if (x === 0) {  
    return;  
  }  
  console.log(x);  
  foo(x - 1);  
}  
foo(3);
```

Part b

```
function foo(x) {  
  if (x === 0) {  
    return;  
  }  
  foo(x - 1);  
  console.log(x);  
}  
foo(3);
```

Name: _____

Part c

```
let a = [{x: 0}];
for (let i = 0; i < 9; ++i) {
  a.push(a[0]);
}
a[0].x = 2;
let s = 0;
for (let i = 0; i < a.length; ++i) {
  s = s + a[i].x;
}
console.log(s);
```

Part d

```
let a = [{x: 0}];
let b = {x: 0};
function update(y, z) {
  z = {x: 0};
  y.push(z);
  y.push(z);
}
update(a, b);
b.x = 10;
let s = a.reduce(function(s, v) { return s + v.x; }, 0);
console.log(s);
```

Name: _____

Part e

```
let a = [];  
let o = { x: 0 };  
function f(a, o) {  
  for (let i = 0; i < 10; ++i) {  
    a.push(o);  
  }  
  o.x = 1;  
  return o;  
}  
let b = f(a, o);  
b.x = 2;  
console.log(a.reduce(function(a, b) {  
  return a + b.x;  
}, 0));
```

Part f

```
function foo() {  
  console.log('A');  
  function bar() {  
    console.log('B');  
  }  
  console.log('C');  
}  
foo();
```

Name: _____

Question 2 (15 points): Consider the following function:

```
function foo(x) {  
  if (x > 0) {  
    console.log('A');  
  }  
  function bar() {  
    if (x < 20) {  
      console.log('B');  
    }  
  }  
  if (x % 2 == 0) {  
    bar();  
  }  
}
```

Part a: Give three values of x for which calling $\text{foo}(x)$ will result in the following output:

A

B

| | | |
|-----|-----|-----|
| x = | x = | x = |
|-----|-----|-----|

Part b: Give three values of x for which calling $\text{foo}(x)$ will result in the following output:

A

| | | |
|-----|-----|-----|
| x = | x = | x = |
|-----|-----|-----|

Part c: Give three values of x for which calling $\text{foo}(x)$ will result in the following output:

B

| | | |
|-----|-----|-----|
| x = | x = | x = |
|-----|-----|-----|

Name: _____

Question 3 (10 points): The following function is supposed to insert the value x into an already-sorted array a (sorted in ascending order), such that the resulting array includes x , and is sorted:

```
function insert(a, x) {
  let result = {
    inserted: false,
    array: []
  };
  result = a.reduce(function(result, value) {
    if (!result.inserted && value > x) {
      result.array.push(x);
      result.inserted = true;
    }
    result.array.push(value);
    return result;
  }, result);
  return result.array;
}
```

Part a: Give three input pairs ($a=?$, $x=?$) for which the result is **correct**.

| | | |
|-----|-----|-----|
| a = | a = | a = |
| x = | x = | x = |

Part b: Give three input pairs ($a=?$, $x=?$) for which the result is **incorrect**.

| | | |
|-----|-----|-----|
| a = | a = | a = |
| x = | x = | x = |

Name: _____

Question 4 (15 points): The following function is supposed to take as input an image, and produce an output image where each pixel is replaced by the mean value of the original pixel, and the pixel values above and below it in the original image.

```
1 function blurVertical(img) {
2   let img2 = img.copy();
3   for (let y = 0; y < img.height; ++y) {
4     for (let x = 0; x < img.width; ++x) {
5       let c = [0, 0, 0];
6       for (let y2 = y - 1; y2 <= y + 1; ++y2) {
7         c[0] += img.getPixel(x, y2)[0] / 3;
8         c[1] += img.getPixel(x, y2)[1] / 3;
9         c[2] += img.getPixel(x, y2)[2] / 3;
10      }
11      img2.setPixel(x, y, c);
12    }
13  }
14  return img2;
15 }
```

Part a: Will this function produce any errors? (Answer yes or no.)

Part b: If it will produce errors, on which line will it produce an error **first**? What is the error?

Part c: If it will produce errors, insert lines of code to fix the error while making minimal changes to the function result. Write your code below, including line numbers from the original program to indicate where your code should be inserted to correct the error.

Insert after line: _____

Inserted Code:

Name: _____

Question 5 (10 points): Using the `reduce` function, write a function `composeFunctions()` that takes in an array of functions, and returns a function that is a composite of all the functions in the array. That is, if the input array is `[f1, f2, f3]`, the resulting function `g` should satisfy $g(x) === f3(f2(f1(x)))$

```
function composeFunctions(arrayOfFunctions) {
  let reducer = function(result, currentFunction) {

};
let initialFunction = function(x) {

};
return arrayOfFunctions.reduce(reducer, initialFunction);
}
```