

# MT1 Review

COMP110 – Spring 2018 – Lecture 21

# Midterm Preparation

- Today – Practice Problems via PollEv – Key slides will post tonight
- Key for last WS Posted (We're grading as quickly as possible!)
- Practice Worksheet on [COMP110.com](http://COMP110.com)
- Tomorrow – Review Session in SN014 at 5pm
- Exam – Thursday, in class, new random seats will post Weds evening

# Makeup

- University excused absence on Thursday?
- Make-up exam is Sunday 4/15 at 12pm in SN011
- Please fill out this form: <http://bit.ly/s18-mt1-makeup>

1. What is **c**'s value after the loop completes?

```
let a: number[] = [1, 2, 3];
let b: number[] = [4, 5, 6];
let c: number[] = [];

for (let i: number = 0; i < a.length; i++) {
  let ci: number = i * 2;
  c[ci] = b[i];
  c[ci + 1] = a[i];
}

print(c);
```

2. Given **f** and **g**, what argument could you call **f** with for it to return "B"? – What about "C"?

```
let f = (x: number): string => {  
  if (x % 2 === 0) {  
    return "A";  
  } else if (x === 4) {  
    return "C";  
  } else {  
    return g(-1 * x);  
  }  
};
```

```
let g = (x: number): string => {  
  if (x > 0) {  
    return "A";  
  } else {  
    return "B";  
  }  
};
```

### 3. What is returned when calling **f(12, 8)**?

(Assume **min** returns the smaller of two numbers.)

```
let f = (n: number, d: number): number => {  
  let g: number = min(n, d);  
  while (g > 1) {  
    if (n % g === 0 && d % g === 0) {  
      return g;  
    }  
    g--;  
  }  
  return g;  
};
```

4. What is printed when the following code runs?

```
let a: string[] = ["a", "b", "c"];
```

```
let x: string = a[2];
```

```
a[a.length] = a[1];
```

```
a[1] = a[2];
```

```
a[2] = a[3];
```

```
print(a);
```

```
print(x);
```

5. Fill in the types for the following program to type check.

```
let f = (x: 1): 2 => {  
    return x.length < 5;  
};
```

```
let a: 3 = ["1", "2", "3"];  
print(f(a));
```



6. Given the following function:

```
let foo = (n: number): boolean => {  
    return n % 3 === 0;  
};
```

And array...

```
let a: number[] = [1, 2, 3, 4];
```

What are the return *types* of the following?

1. `a.map(foo)`

2. `a.filter(foo)`

Done? What are the *values returned*?

7. Given the Iterator class, what is printed when the code right runs?

```
class Iterator {
  a: number[];
  i: number;

  constructor(a: number[]) {
    this.a = a;
    this.i = a.length;
  }

  hasNext = (): boolean => {
    return this.i > 0;
  }

  next = (): number => {
    this.i--;
    return this.a[this.i];
  }
}
```

```
let numbers = [1, 2, 3];
let itr = new Iterator(numbers);
numbers[numbers.length] = 4;
numbers[1] = 5;

while (itr.hasNext()) {
  print(itr.next());
}
```

8. What is printed when the main function below runs?

```
let main = async () => {
  print("F");
  let krackle = new Dog("Krackle");

  let hello = krackle.speak("hello");

  print(krackle.speak("world"));
  print(hello);
}
```

```
class Dog {
  name: string;

  constructor(name: string) {
    print("A");
    this.name = name;
  }

  translate = (word: string): string => {
    print("B");
    if (word !== "hello") {
      return "C";
    } else {
      return "D";
    }
  }

  speak = (word: string): string => {
    print("E");
    return this.translate(word);
  }
}
```

9. Given the following description of array's **concat** method, and the **Reducer<number, number[]>** function **baz**...

**a.concat(b)** will return a new, concatenated array starting with a's elements and ending with b's elements.

```
function baz(memo: number[], item: number): number[] {  
  let start: number[] = [item];  
  return start.concat(memo);  
}
```

What are result's elements after the following code runs:

```
let input = [1, 2, 3];  
let result = input.reduce(baz, []);
```