

This lab is group lab. Write down the names of contributing members. Total points is 16.

Please complete the following methods in **Stream.java** in the **lab4-start** branch. You could choose to copy and paste code to the pdf (and print it out) instead of hand-writing it. Only the body is needed. Uncomment the informal tests in **Main.java** to check your implementation. Note the simple tests may not be sufficient. You may want me to check them just to be sure.

1. (2 points) Complete **from**.

```
public static Stream<Integer> from(int n) {  
    return cons(() -> n, () -> from(n + 1));  
}
```

2. (2 points) Complete **iterate**.

```
public static <A> Stream<A> iterate(A init, UnaryOperator<A> f) {  
    return cons(() -> init, () -> iterate(f.apply(init), f));  
}
```

3. (2 points) Complete **dropN**.

```
public static <A> Stream<A> dropN(Stream<A> s, int n) {  
    return n == 0  
        ? s  
        : dropN(s.tail(), n - 1);  
}
```

4. (2 points) Complete **takeN**.

```
public static <A> Stream<A> takeN(Stream<A> s, int n) {  
    return n == 0  
        ? nil()  
        : cons(() -> s.head(), () -> takeN(s.tail(), n - 1));  
}
```

5. (2 points) Complete **multStream**.

```
public static Stream<Integer> multStream(Stream<Integer> s1, Stream<Integer> s2) {  
    return cons(() → s1.head() * s2.head(), () → multStream(s1.tail(), s2.tail()));  
}
```

6. (2 points) Complete **map** for **Cons** class.

```
public <B> Stream<B> map(Function<A, B> f) {  
    return cons(() → f.apply(this.head()), () → this.tail().map(f));  
}
```

7. (2 points) Complete **filter** for **Cons** class.

```
public Stream<A> filter(Predicate<A> p) {  
    return p.test(this.head())  
        ? cons(() → this.head(), () → this.tail().filter(p))  
        : cons(() → this.tail().head(), () → this.tail().tail().filter(p));  
}
```

8. (2 points) Complete **sieve**.

```
public static Stream<Integer> sieve(Stream<Integer> ns) {  
    return cons(() → ns.head(), () → sieve(ns.filter(n → n % ns.head() ≠ 0 || n.equals(ns.head())).tail()));  
}
```