CSE 143x Section Handout #6

1.

	n == 0	n % 2 == 1	x == 0
Point A	sometimes	sometimes	always
Point B	never	sometimes	sometimes
Point C	never	always	never
Point D	sometimes	sometimes	sometimes
Point E	always	never	sometimes

2.

	x > 2	x < n	n % x == 0
Point A	never	sometimes	sometimes
Point B	sometimes	always	sometimes
Point C	never	sometimes	sometimes
Point D	always	sometimes	sometimes
Point E	sometimes	never	sometimes

3.

```
public class BankAccount {
    String name;
    double balance;
}
```

4.

```
    public void deposit(double  amount) {
        balance = balance + amount;
    }
```

5.

```
    public void withdraw(double amount) {
           balance = balance – amount;
    }
```

6.

```
public class BankAccount {
    String name;
    double balance;
    double transactionFee;

    public void deposit(double  amount) {
        balance = balance + amount;
    }

    public void withdraw(double amount) {
        if(balance - amount - transactionFee >=  0) {
           balance = balance - amount;
           balance -= transactionFee;
        }
    }
}
```

CSE 143x Section Handout #6

7.
```
public boolean isVertical(Point other) {
    // Note that writing 'this.' is optional. 'x' would also refer to the field.
    // This explicit form comes in handy for clarity in situations involving
    // another instance of the same class.
    return this.x == other.x;
}
```

8.
```
public int manhattanDistance(Point other) {
    return Math.abs(this.x - other.x) + Math.abs(this.y - other.y);
}
```

9.
```
public double slope(Point other) {
    if(this.x == other.x)
        throw new IllegalArgumentException();
    return (((double)(other.y - this.y)) / (other.x - this.x));
}
```

10.
```
public static double euclideanDistance(NDPoint a, NDPoint b) {
    if(a.coordinates.length != b.coordinates.length) {
        throw new IllegalArgumentException();
    }
    double differenceSquaredSum = 0;
    for(int i = 0; i < a.coordinates.length; i++) {
        differenceSquaredSum += Math.pow(a.coordinates[i] - b.coordinates[i], 2);
    }
    return Math.sqrt(differenceSquaredSum);
}
```