CSE 142 Section Handout #4
Cheat Sheet

`Math` class (3.2)
A set of useful methods for performing mathematical operations

Method	Returns	Method	Returns
`Math.abs(value)`	absolute value	`Math.max(value1, value2)`	larger of two values
`Math.ceil(value)`	rounds up	`Math.min(value1, value2)`	smaller of two values
`Math.cos(value)`	cosine, in radians	`Math.pow(base, exponent)`	*base* to the *exponent* power
`Math.floor(value)`	rounds down	`Math.round(value)`	nearest whole number
`Math.log(value)`	logarithm, base *e*	`Math.sin(value)`	sine, in radians
`Math.log10(value)`	logarithm, base 10	`Math.sqrt(value)`	square root

```
double bigger = Math.max(Math.pow(2, 3), Math.sqrt(49));
System.out.println("The bigger value is " + bigger);
```

Constant	Value
`Math.E`	2.7182818...
`Math.PI`	3.1415926...

Return (3.2)
(A way to pass information out from a method to its caller)

```
public static type name(parameters) {
    statement(s);
    ...
    return expression;
}

public static double fToC(double degreesF) {    // Converts Fahrenheit to Celsius.
    double degreesC = 5.0 / 9.0 * (degreesF - 32);
    return degreesC;
}
```

Scanner (3.3)
(An object to read values from the keyboard)

```
import java.util.*;
Scanner console = new Scanner(System.in);
System.out.print("How old are you? ");
int age = console.nextInt();
System.out.println("You'll be 40 in " +
        (40 - age) + " years.");
```

Method	Returns
`nextInt()`	next token as `int` if possible.
`nextDouble()`	next token as `double` if possible.
`next()`	next token as `String`
`nextLine()`	next line as `String`

Conditional Logic (4.1)

```
if (test) {
    statement(s);
}


// 0, 1, or 2 branches taken
if (test) {
    statement(s);
}
if (test) {
    statement(s);
}


// 0 or 1 branches taken
if (test) {
    statement(s);
} else if (test) {
    statement(s);
}
```

```
// exactly 1 branch taken
if (test) {
    statement(s);
} else {
    statement(s);
}



// exactly 1 branch taken
if (test) {
    statement(s);
} else if (test) {
    statement(s);
} else {
    statement(s);
}
```

Cumulative Sum (4.2)

```
// add up all numbers from 1-max
int sum = 0;
for (int i = 1; i <= max; i++) {
    sum = sum + i;
}
```

Operator	Description
`<`	less than
`<=`	less than or equal
`>`	greater than
`>=`	greater or equal
`==`	equal
`!=`	not equal
`&&`	and
`||`	or
`!`	not