For Your Reference

Operator Precedence Table

Precedence	*Operator*
↓	`var++` and `var--` (postfix)
	`+, -` (Unary plus and minus), `++var` and `--var` (prefix)
	(type)(Casting)
	`!`(Not)
	`*,/,%` (Multiplication, division, and remainder)
	`+,-` (Binary addition and subtraction)
	<,<=,>,>= (Relational)
	==,`!=` (Equality)
	^ (Exclusive OR)
	`&&` (AND)
	`||` (OR)
	=,+=,-=,*=,/=,`%=` (Assignment operator)

Methods

You may find all, some, or none of these methods useful. Descriptions are taken from the textbook or the Java 8 Specification; some have been slightly abbreviated for space.

Methods from the Scanner (java.util.Scanner):

`Scanner(System.in)`	creates a `Scanner` that reads from the keyboard
`void close()`	closes the scanner
`boolean hasNext()`	returns true if this scanner has another token in its input
`boolean hasNextInt()`	returns true if the next token can be interpreted as an int
`boolean hasNextDouble()`	returns true if the next token can be interpreted as a double
`boolean hasNextLine()`	returns true if there’s another line of input
`String next()`	reads a `String` that ends with a whitespace character
`int nextInt()`	reads an integer of the `int` type
`double nextDouble()`	reads a number of the `double` type
`String nextLine()`	reads a `String` that ends with a newline character

Methods from the Math class (java.lang.Math):

`double random()`	returns a double value [0.0, 1.0)
`double pow(double a, double b)`	returns a raised to the power of `b` (a^b)
`double sqrt(double a)`	returns the square root of `a` ($\sqrt{a}$) for `a` >= 0
`double floor(double a)`	`a` is rounded down to its nearest integer, returned as double
`double ceil(double a)`	`a` is rounded up to its nearest integer, returned as double
`long round(double a)`	`a` is rounded to the closest integer, returned as long

Methods from the Character class (java.lang.Character):

`boolean isDigit(char c)`	returns true if `c` is a digit
`boolean isLetter(char c)`	returns true if `c` is a letter
`boolean isLowerCase(char c)`	returns true if `c` is a lowercase letter
`boolean isUpperCase(char c)`	returns true if `c` is an uppercase letter
`char toLowerCase(char c)`	returns the lowercase version of `c`
`char toUpperCase(char c)`	returns the uppercase version of `c`

Methods from the String class (java.lang.String):

`int length()`	returns the number of characters in this string
`char charAt(int index)`	returns the character at the specified index in this string
`String toLowerCase()`	returns a new String with all letters in lowercase
`String toUpperCase()`	returns a new String with all letters in uppercase
`boolean equals(String s1)`	returns true if this string is equal to string `s1`
`int compareTo(String s1)`	returns an integer > 0, == 0, or < 0 to indicate whether this string is greater than, equal to, or less than `s1`
`boolean contains(String s1)`	returns true if `s1` is a substring of this string
`int indexOf(String s1)`	returns the index of the first occurrence of string `s1` in this string. Returns -1 if not matched.
`int indexOf(String s1, int index)`	returns the index of the first occurrence of string `s1` in this string after index. Returns -1 if not matched.
`String substring(int begin)`	returns this string's substring that begins with the character at the specified `begin` index and extends to the end of the string
`String substring(int begin, int end)`	returns this string's substring that begins at the specified `begin` index and extends to the character at `end - 1`.