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
`Scanner(File file)` `throws FileNotFoundException`	creates a `Scanner` that reads from `file`
`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

Scanner, continued

`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, continued

`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`.
`String[] split(String delim)`	returns an array of Strings consisting of the substrings split by the delimeter `delim`

Methods from the Arrays class (java.util.Arrays):

`E[]` indicates an array of elements are of type `E`.

`static sort(E[] array)`	Sorts the specified array in memory
`static String toString(E[] array)`	Returns a String representation of the specified array

Methods from the ArrayList class (java.util.ArrayList):

`E` indicates an object of type `E`.

`int size()`	Returns the number of elements in this list.
`add(E item)`	Appends the element `item` at the end of this list
`add(int index, E item)`	Adds the element `item` at the specified index in this list
`E get(int index)`	Returns the element at the specified `index` in this list
`E set(int index, E item)`	Replaces the element at the specified `index` in this list with the element `item`.
`boolean remove(E item)`	Removes the first occurrence of the element `item` from the list, if it is present.
`E remove(int index)`	Removes the element at the specified `index` in this list
`void clear()`	Removes all the elements from this list
`boolean contains(E item)`	Returns `true` if this list contains `item`, `false` otherwise

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

`String toString()`	Returns a String representation of the object. This is the hashcode of the instance unless this method is overridden.
`boolean equals(Object obj)`	Returns true if `this` is equal to `obj`, false otherwise. This checks if the object reference `obj` is the same as `this`, unless overridden by the class.

Methods from the File class (java.io.File):

`File(String filename) throws FileNotFoundException`	Creates a `File` object for the given file name, `filename`
`boolean exists()`	Returns true if the specified file already exists, false otherwise

Methods from the PrintWriter class (java.io.PrintWriter):

`PrintWriter(String filename) throws FileNotFoundException`	Creates a new `PrintWriter` for the given file name, `filename`
`PrintWriter(File file) throws FileNotFoundException`	Creates a new PrintWriter from `file`
`void print(String str)`	Prints the String `str`
`void println(String str)`	Prints the String `str` followed by a newline character
`void close()`	Closes the PrintWriter

Methods from the Comparable<E> interface:

`int compareTo(E obj)`	Returns a negative integer, zero, or a positive integer if this object is less than, equal to, or greater than `obj`.