# Java I Cheat Sheet

## Comments

Explanation Example Single

```
// comment  text
```

Multiline (Basic)

```
/*
comment text
*/
```

Multiline JavaDoc

```
/**
 * comment text
 * @author BobSaidHi
*/
```

### Table of JavaDoc tags

Oracle Docs - JavDoc info

Complete list

| Tag | Explanation |
|---|---|
| `@author` | Author tag |
| `@version` | Version tag |
| `@param` | Explains one of the method's parameters |
| `@return` | Explains what the method returns |
| `@see` | References more information |
| `@since` | When something was first implemented |
| `@deprecated` | Deprecated tag |

## Primitives (Basic Variables)

### Example with an integer

```java
// Multiple Statements
int i;
i = 0;

// Single Statement
int i = 0;
```

Complete List of Primitives

### Table of Primitives

| Use | Symbol/ Keyword | Explanation |
|---|---|---|
| integer | `int` | Holds a 32-bit integer. There are also other primitives that can hold integers (`byte`, `short`, and `long`). |
| character | `char` | Can hold 1 unicode character. Can be treated as a special int |
| decimal | `double` | Holds a double precision floating point number, which a way to store a decimal. There are also one other primitives that can hold decimals (`float`). |
| boolean | `boolean` | Holds a boolean value (`true` OR `false`) |
| no return | `void` | In a method declaration, a return type of `void` means that nothing will be returned. |

> Note: Arrays are indicated with brackets.

## Math Operations

Order of operations

| Name | Example | Explanation |
|---|---|---|
| Parentheses | `(A)` | A is in parentheses |
| Multiplication | `A * B` | A multiped by B. |
| Division | `A / B` | A divided by B. |
| Addition | `A + B` | A plus B. |
| Subtraction | `A - B` | A minus B. |
| Modulus | `A%B` | Gives the remainder of `A / B` |

Also, `+=`, `-=`, `*=`, and `/=` allow one to perform an an operation on a variable then then assigning the result to said variable.

### Special Loop Operators

| Example | Action | Explanation |
|---|---|---|
| `A++` | post-increment | |
| `A--` | post-decrement | |
| `++A` | pre-increment | |
| `--A` | pre-decrement | |

### Math Class

| Name | Symbol | Explanation |
|---|---|---|
| Exponents | `Math.pow(base, exponent)` | Import the `Math` class by typing `import java.lang.Math`, then... |

## Strings (Fancy Variables)

A `String` is a special type of variable, because it is actually an object that is implemented as a `char` array. The first character will receive an index of `0`. When you write one, using double quotes, that is called a String literal. More info

```
// Create a String variable from a String literal
String s = "I'm a String";

// Create a String using a constructor:
char[] helloArray = { 'h', 'e', 'l', 'l', 'o', '.' };
String helloString = new String(helloArray);
System.out.println(helloString);
```

APCSA Java Quick Reference

`String` Methods:

| Method | Explanation |
|---|---|
| `int length()` | Returns the number of characters in a `String` object |
| `String substring(int from, int to)` | Returns the substring beginning at index `from` and ending at index `to - 1` |
| `String substring(int from)` | Returns `substring(from, length())` |
| `int indexOf(String str)` | Returns the index of the first occurrence of `str`; returns `-1` if not found |
| `int compareTo(String other)` | Returns a value `<0` if this is less than `other`; returns `0` if `this` is equal to `other`; returns a value `>0` if `this` is greater than `other` |

## Print Statements

JavaDoc for `System.out`

```
// Print something out
System.out.print("I'm a String");

// Print something out and add a line break
System.out.println("I'm a String");

// Concatenating a String and a integer
System.out.println("Five as a number is: " + 5);
```

> Please note that "I'm a String" (including the quotes) can be replaced by almost anything, including other data types, variables, and even some objects.

## Escape Sequences

Further Reading

| Sequence | Meaning |
|---|---|
| `\` | Open escape sequence |
| `\t` | tab |
| `\n` | line break (new line) |
| `\"` | Allows the use of double quotes in a String |
| `\'` | Allows the use of a single quot in a char |
| `\\` | Allows the use of a backslash in text |