# Strings & String Methods Reference Sheet

### Essential knowledge about Strings

* String objects can be created using string literals or by calling the String class constructor
* String objects are immutable, meaning that String methods do not change the String object
* String objects can be concatenated using the + or += operator, resulting in a new String object
* Primitive values can be concatenated with a String object. This causes implicit conversion of the values to String objects
* Escape sequences start with a \ and have a special meaning in Java. Escape sequences used in this course include \”, \\, and \n

A string is a sequence or row of characters. We can access each character using an index.



### String Methods

String(String str)

constructor – creates a string object

String name = new String(“Spongebob Squarepants”);

int length()

returns the number of characters in the string

int strLength = name.length();

The value of strLength would be 21

char charAt(int index)

returns the character found at the index of the string

char firstInitial = name.charAt(0);

The value of firstInitial would be ‘S’

int indexOf(char character)

returns the index of the first occurrence of that character, or -1 if not found

int num = name.indexOf(‘S’);

int otherNum = name.indexOf(‘x’);

The value of num is 0; the value of otherNum is -1

String substring(int from, int to)

returns the part of the string starting at index from until index to - 1

String firstName = name.substring(0,9);

The value of firstName is “Spongebob”

String substring(int from)

returns the part of the string starting at index from until end of string (length())

String lastName = name.substring(10);

The value of lastName is “Squarepants”

boolean equals(String other)

returns true if this is equal to other

boolean isFirstEqualLast = firstName.equals(lastName);

isFirstEqualLast is false (Spongebob does not equal Squarepants)

int compareTo(String other)

returns a negative number (< 0) if this is less than other;

returns 0 if this is equal to other;

returns a positive number (> 0) if this is greater than other

int compareFirstLast = firstName.compareTo(lastName);

compareFirstLast is -1. p is “less than” q in the alphabet