

```
1
2 public class Main {
3
4     public static void main(String[] args)
5     {
6
7         // String is a Class !!
8         // Standard one in Java
9         String str = "Hello John Smith";
10        System.out.println(str);
11
12        // There are a few methods which we will use.
13        // The ones given on the AP card at the back:
14        /*
15        class java.lang.String
16        int length()
17        String substring(int from, int to) - returns the substring beginning at from and ending at to-1
18        String substring(int from) - returns substring(from, length())
19        int indexOf(String str) - returns the index of the first occurrence of str; returns -1 if not found
20        int compareTo(String other) - returns a value < 0 if this is less than other
21                                   returns a value = 0 if this is equal to other
22                                   returns a value > 0 if this is greater than other
23
24        Don't have to use, not given in the back of the card, but is REALLY helpful:
25        char charAt(int index)
26        */
27
28
29        // length - the number of characters in the string
30        // int length()
31        System.out.println("String length = " + str.length());
32
33        // Addition of strings : Concatentation
34        System.out.println(str + " -- " + str );
35
36        // Whe using concatentation, one of them should always be a string
37        // the easiest way to convert a value to a trsing is to add to an empty string.
38        // "" + 90
39
40        // substring
41        // Sring Substring (from, pastEnd);
42        // *not* including last index, in this case 7.
43        System.out.println("substring :" + str.substring(0,7));
44
45        // substring
46        // String substring(int from) - returns substring(from, length())
47        System.out.println("substring :" + str.substring(6));
48
49
50        // String is composed of characters
51        // char charAt(int index)
52        // String chars start from 0!!
53        System.out.println("3rd character is:" + str.charAt(3));
54
55
56        char c1 = 'A' ;
```

```

57
58 // Escape sequences
59 String str2 = " quoting \" He said\" ";
60 String str3 = " a \" string" ;
61 String str4 = " a new line\n " ;
62
63 char c2 = '\n';
64
65
66 // indexOf (char), or String
67 // -1 if doesn't find
68 System.out.println("index of H is " + str.indexOf('H'));
69
70
71
72 // int compareTo(String other) - returns a value < 0 if this is less than other
73 // returns a value = 0 if this is equal to other
74 // returns a value > 0 if this is greater than other
75 String str5 = "a string";
76 String str6 = "b string";
77 System.out.println(str5 + ".compareTo("+str6 + ")=" + str5.compareTo(str6));
78 String str7 = "a string longer";
79 System.out.println(str5 + ".compareTo("+str7 + ")=" + str5.compareTo(str7));
80 String str8 = "A string";
81 System.out.println(str5 + ".compareTo("+str8 + ")=" + str5.compareTo(str8));
82
83
84 // CodingBat:
85 // go there, open an account, and then go to
86 // http://codingbat.com/pref
87 // At the bottom, and zbaharav@kehillah.org And then i can see your work!
88 // Solve at-least 5 problems from there.
89 // char: single character + single quotes
90 // One of the primitive types!
91
92
93 // String compareTo - Comparable
94 // Object identity (==) .vs. object equality (equals)
95 // Strings are immutable!
96 String str9 = "a ";
97 str9 = str9 + "string";
98 System.out.println(str5 + "==" + str9 + " is " + (str5==str9));
99 System.out.println(str5 + ".compareTo(" + str9 + ") is " + (str5.compareTo(str9)));
100 System.out.println(str5 + ".equals(" + str9 + ") is " + (str5.equals(str9)));
101
102
103
104
105
106
107 // OUR own class
108 // Example for: Going over all elements of a String
109 // String KHyphen()
110
111 KString kstr;
112 kstr = new KString("Hello");
113 System.out.println(kstr + " hyphenated is " + kstr.KHyphen());
114

```

```

115 kstr = new KString("AB");
116 System.out.println(kstr + " hyphenated is " + kstr.KHyphen());
117
118 kstr = new KString("A");
119 System.out.println(kstr + " hyphenated is " + kstr.KHyphen());
120
121 kstr = new KString("");
122 System.out.println(kstr + " hyphenated is " + kstr.KHyphen2());
123
124
125
126
127 // For the homework!!
128
129 // KHyphen2()
130 kstr = new KString("Hello");
131 System.out.println(kstr + " hyphenated2 is " + kstr.KHyphen2());
132
133 kstr = new KString("ABC");
134 System.out.println(kstr + " hyphenated2 is " + kstr.KHyphen2());
135
136 kstr = new KString("AB");
137 System.out.println(kstr + " hyphenated2 is " + kstr.KHyphen2());
138
139 kstr = new KString("A");
140 System.out.println(kstr + " hyphenated2 is " + kstr.KHyphen2());
141
142 kstr = new KString("");
143 System.out.println(kstr + " hyphenated2 is " + kstr.KHyphen2());
144
145 // Palindrome
146
147 kstr = new KString("AbA");
148 System.out.println(kstr + " is a palindrome: " + kstr.isPalindrome());
149 kstr = new KString("AbbA");
150 System.out.println(kstr + " is a palindrome: " + kstr.isPalindrome());
151 kstr = new KString("A");
152 System.out.println(kstr + " is a palindrome: " + kstr.isPalindrome());
153 kstr = new KString("Ab");
154 System.out.println(kstr + " is a palindrome: " + kstr.isPalindrome());
155 kstr = new KString("Abb");
156 System.out.println(kstr + " is a palindrome: " + kstr.isPalindrome());
157 kstr = new KString("");
158 System.out.println(kstr + " is a palindrome: " + kstr.isPalindrome());
159
160
161
162
163
164
165
166 }
167
168 }
169

```