

## AP COMPUTER SCIENCE A – TOKENIZATION OF STRINGS

Consider the program StringTokenizer.

```

public class StringTokenizer {
    /**
    public static void main ( String [] arg ) {
        /**
        String s;
        String [] tok1, tok2;
        /**/
        s = "John:Quincy:Adams"; // tokens for s.split(":")
        /**/
        tok1=tokenizer1(s);
        tok2=tokenizer2(s);
        /**/
        printTokens(tok1);
        printTokens(tok2);
        /**/
        return;
    }
    /**/
    private static String [] tokenizer1 ( String s ) {
        /**/
        return s.split(":"); // ":" is called the tokenization character
    }
    /**/
    private static String [] tokenizer2 ( String s ) {
        /**/
        int numTokens, numNonEmptyTokens, k;
        String [] sa, rv;
        /**/
        sa=s.split(":");
        numTokens=sa.length;
        /**/
        numNonEmptyTokens=0;
        for ( int i=0; i<numTokens; ++i ) {
            if ( !sa[i].equals("") ) ++numNonEmptyTokens;
        }
        /**/
        rv=new String [numNonEmptyTokens];
        k=0;
        for ( int i=0; i<numTokens; ++i ) {
            if ( !sa[i].equals("") ) rv[k++]=sa[i];
        }
        /**/
        return rv;
    }
    /**/
    private static void printTokens ( String [] tokens ) {
        /**/
        int numTokens;
        /**/
        numTokens=tokens.length;
        System.out.println();
        System.out.println( " numTokens = " + numTokens );
        for ( int i=0; i<numTokens; ++i ) {
            System.out.println(
                "tokens[" + String.format("%2d",i) + "] = " + tokens[i] );
        }
        /**/
        return;
    }
}

```

*Instead of ":", a space would be the most usual case (for space delimited files)*

*Count number of non-empty tokens*

*put only non-empty tokens into the String [] rv.*

*"i" is right-justified in a field-width of 2. d = decimal integer*

*backslash is an escape character, i.e., "\\" is the string "*

## AP COMPUTER SCIENCE A – TOKENIZATION OF STRINGS

Compile and run the program. Its output should be

```
numTokens = 11
tokens[ 0] = ""
tokens[ 1] = ""
tokens[ 2] = ""
tokens[ 3] = "John"
tokens[ 4] = ""
tokens[ 5] = ""
tokens[ 6] = ""
tokens[ 7] = ""
tokens[ 8] = "Qunicy"
tokens[ 9] = ""
tokens[10] = "Adams"

numTokens = 3
tokens[ 0] = "John"
tokens[ 1] = "Qunicy"
tokens[ 2] = "Adams"
```