```
Lecture 11 Summary
==================

6 parts: String, Character, Conversion, StringBuffer, Splitting string, Command line arg.


======================================================================

[Class String]

- What is a string?
  - A series of characters in double quotes
  - For Java: Java.lang.String
    - A class representing string
    - Create string objects (reference type)
    - Syntax:
      String aStringObject = new String( "Hello String" );  // OR
      String aStringObject = "Hello String" ;
    - Note:
      aStringObject contains **reference address**

- Constructors provided by class String
  - There are several, see P.7
    OR see http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html

- IO
  - Input from Keyboard:
    - System.in.read()
      - Reads one character (in integer format) from the keyboard
    - Scanner class
      - Using methods next(), nextLine(), etc.
  - Output to Screen:
    - System.out.println( ... )
      - Writes a string to the screen with carriage return at the end
    - System.out.print( ... )
      - Writes a string to the screen

- Instance methods provided by Class String
  - length()
    - For arrays, length is an instance variable
      For String, length() is a method
  - charAt( int pos )
    - return the character at a position (starts from zero)
  - Comparisons: equals(), equalsIgnoreCase(), compareTo() , and compareToIgnoreCase()
    - lexicographic order:
      space < digits (0-9) < letters (A-Z, a-z)
    - equals()    -> return true/false
    - compareTo() -> return 0, >0, or <0
  - indexOf()     -> locates the first occurrence
    lastIndexOf() -> locates the last  occurrence
  - substring()
    - extract substring from a string
  - concat()
    - concatenate two strings to form a new string
    - Or you can use: +
  - Replacement of strings
    - toLowerCase()
    - toUpperCase()
    - trim()
      -> removing whitespace
    - replace( char oldChar , char newChar )
      -> replacing all occurrences of oldChar by newChar
  - others:
    - getChars(), hasCode(), startsWith(), and endsWith()
    - see http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html


======================================================================
```

```
[Class Character]

- java.lang.Character
  - A Java class that represents char as objects
  - Class Methods:
    - For testing char and returns boolean:
      - isDigit(), isLetter(), isSpaceChar(), isWhiteSpace(),
        isLowerCase(), and isUpperCase()
    - For converting characters:
      - toLowerCase() and toUpperCase()
  - http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Character.html


=========================================================================

[Conversion: Numbers <-> String]

- Conversion Methods:
  1. Numbers to Strings: Three ways:
     - String str = "" + num ;
     - Use Wrapper classes
       - class method of Integer, Long, Float and Double
         String str = Integer.toString( i );  // for int i
         String str = Double.toString(  d );  // for double d
     - Use String class method:
       String valueOf(Type Value)
  2. Strings to Numbers Conversion
     - Use Wrapper classes' class method:
       - int    i = Integer.parseInt(str1);
         long   l = Long.parseLong(str2);
         float  f = Float.parseFloat(str3);
         double d = Double.parseDouble(str4);


=========================================================================

[StringBuffer]

- Compare:
  - String Class
    - Provides many methods for processing strings
    - But String objects cannot be changed once created
  - StringBuffer Class
    - Provides methods for creating dynamic string information
    - StringBuffer objects can be modified and extended

- StringBuffer Class
  - Instance methods
    - Modify the string contents:
      append() -> Adds a data item to the end
      insert() -> Inserts a data item at a specified index position
      delete() -> removes character(s)
    - About string length:
      length()   -> Returns # characters currently inside
      capacity() -> Returns # characters that can be stored
      ensureCapacity(), setLength()
    - Others (similar to String):
      charAt(), setCharAt(), getChars(), reverse(), replace(),
      substring(), toString()
  - Constructors:
    StringBuffer()
    StringBuffer( int    length  )
    StringBuffer( String aString )


=========================================================================

[Splitting a String: StringTokenizer and Scanner]
```

- StringTokenizer Class
  - From package java.util
  - Main purpose:
    - Breaks up a string into components or tokens
      as separated by delimiters
  - constructors:
    - StringTokenizer(String str) -> white-spaces as delimiter
    - StringTokenizer(String str , String delim)
  - After creating a StringTokenizer object, instances methods:
    - hasMoreTokens() -> return true/false
    - nextTokens()   -> return next substring
    - countTokens()   -> return # tokens remaining

- Scanner Class
  - In Scanner class, a word can be used as a delimiter:
    Scanner scObj = new Scanner("ABC DEF GHI");
    scObj.useDelimiter("DEF");
    while (scObj.hasNext()) { String next = scObj.next() ; }

==========

[Command Line Arguments]

- The command line is the string of characters we type when starting a program

- Arguments can be given to commands as options.
  $java CommandLineApp argument1, argument2, ...

- Arguments of main() methods receive these input arguments:
  ... main(String[ ] args) ...
  -> args stores the array of argument strings
     -> args[0] stores argument1, etc.