Keywords and Strings

- **Uses of Strings**
  - Display messages
  - Input text from users from TextField, TextArea, keyboard
  - Manipulate files
  - Manipulate URLs

- **String objects are immutable:** they can’t be changed. This characteristic provides for efficient memory usage.

- **Note** – when the string is one character in length, *char* can be used; processing time is less. See following example.
  - char initial = ‘M’;
  - char marker = ‘\n’;
  - char letter;
  - letter = initial;
  - compare:
    - If (initial == ‘s’)... 

- **Successive digits values:**
  - If ((initial >= ‘0’)&(initial <= ‘9’))...
  - If (initial >= ‘A’)&(initial <= ‘Z’))...
Strings

• Three string classes: String, StringBuffer, and StringBuilder
  – String x, y;
  – String myName = “Tom”;
  – x = “France”;
  – y = x;
  – x = “”;
• Concatenation:
  – int number = 123;
  – G.drawString(“value is “ + number, 100, 100);
• Appending:
  – x = x + “someString”;
• Comparing strings:
  – if (string1.equals(string2))…
  – If (lastName.equalsIgnoreCase(“sMiTh”))…
  – n = “ant”.compareTo(“bee”);             // n = negative value
  – n = “bee”.compareTo(“ant”);             // n = positive value
  – n = “bee”.compareTo(“bee”);             // n = 0
  – n = “INSECT”.compareTo(“ant”);          // n = negative value
Strings

- Array of strings (note – a regular object is a *char* array):
  - `String cities[] = new String[10];`
  - `cities[1] = "Atlanta";`
- Converting strings to int or float using wrappers
  - `int intValue = Integer.parseInt(dataString);`
- Characters within strings
  - `g.drawString("A \"tricky\" problem!", 100, 100);`
- StringDemo example
- Amending strings
  - `string1 = "Florida".replace(‘i’,’o’);`  // yields “Floroda”
  - `string1 = "Version 1.1";`
  - `result = string1.toLowerCase();`  // yields “version 1.1”
  - `result = string1.toUpperCase();`  // yields “VERSION 1.1”
  - `string1 = "     center     ";`
  - `result = string1.trim();`  // yields “center”
StringBuffer and StringBuilder

- Since string objects are immutable the content of a string can be manipulated to produce a new string using either the StringBuffer or StringBuilder classes.
- Content manipulation includes such operations as reversing the character order, replacing characters, appending multiple strings, etc.
  
  Example:
  ```java
  StringBuffer sb = new StringBuffer("airplane");
  sb.setChar(4,'h');
  sb.setChar(5,'o');  // yields “airphone”
  ```
- Note: the string ‘sb’ was actually modified at its memory location.
- The StringBuilder class is virtually identical to the StringBuffer class, except that it provides better performance; it was introduced in J2SDK v1.5.
- Example2:
  ```java
  int i = 17;
  double d = 13.22;
  char c = ‘q’;
  StringBuffer str = new StringBuffer("");
  str.append(i);
  str.append(d);
  str.append(c);  //str holds 1713.22q
  ```