

Unit 10

- The String class



The String Class

Def. String – a collection of characters stored as one object.

Ex.

A	P		C	o	m	p	u	t	e	r
0	1	2	3	4	5	6	7	8	9	10

Strings are **immutable**. That is once a String object is created, none of its methods can change it.



The String Class (continued)

`java.lang.String` implements `java.lang.Comparable`

What does this mean??

`Comparable` is an interface, meaning that there are methods defined, but not written for the class.

```
interface java.lang.Comparable
```

```
int compareTo(Object other)
```

```
// return value < 0 if this is less than other
```

```
// return value = 0 if this is equal to other
```

```
// return value > 0 if this is greater than other
```



The String Class (continued)

How are Comparable objects used??

Ex. Write a method that returns the greatest of two String objects (in lexicographical order)

```
public String greatest (String s1, String s2)
{
    if(s1.compareTo(s2) > 0)
        return s1;
    else
        return s2;
}
```



The String Class (continued)

Other String methods.

```
int length() // returns the # of characters in String

String substring(int from, int to)
// returns the substring beginning at from
// and ending at to-1

String substring(int from)
// returns substring(from, length())

int indexOf(String s)
// returns the index of the first occurrence of s;
// returns -1 if not found
```

See page 265 in your book for a list of other methods



String Constructors

- String's no-args and copy constructors are not used much.

```
String s1 = new String ();
```

```
String s2 = new String (s1);
```

```
String s1 = "";
```

```
String s2 = s1;
```

- Not to be confused with an uninitialized string:

```
private String errorMsg;
```

**errorMsg
is null**



The String Class (continued)

Ex. Analyze the following code segment:

```
String s = "Help me, I am lost!!";  
  
int a = s.length();  
String b = s.substring(5, 14);  
String c = s.substring(13);  
int d = s.indexOf("me");  
int e = s.indexOf("z");
```



Escape Sequences

<u>Sequence</u>	<u>Meaning</u>
\n	newline
\t	tab
\'	single quote
\"	double quote
\\	backslash

```
System.out.print("\nDon't let me down\nDon't let me down");
```



Input/Output (I/O) from the console screen in Java (the little black screen)

One input class: `EasyReader.java`

Examples

```
EasyReader console = new EasyReader();  
  
String lastName = console.readWord();  
String fullName = console.readLine();  
  
int age = console.readInt();  
double GPA = console.readDouble();
```



File Input

The last slide used `EasyReader` to read from the keyboard. Now we will use it to read from a file.
How??

```
EasyReader inFile = new EasyReader(filename);

while(!inFile.eof())           // for char, int, double
{
    char c = inFile.readChar();
    ... do stuff
}
```

```
EasyReader inFile = new EasyReader(filename);

String s;
while((s = inFile.readWord()) != null) // for Strings
{
    ... do stuff
}
```



File Input (continued)

Ex. Open a file that contains text and count how many words contain the character 'e'.

```
EasyReader inFile = new EasyReader(filename);

String s;
int count = 0;

while((s = inFile.readWord()) != null)
{
    if(s.indexOf("e") != -1)
        count++;
}

System.out.println("There are " + count +
    " words with \'e\' in the file");
```

