CSPP 511-01:
Introduction to Object-Oriented Programming

Harri Hakula

Ryerson 256, tel. 773-702-8584
hhakula@cs.uchicago.edu

June 28, 2000
Outline

- User-Defined Types or Classes
- Implementation vs Interface
- String Basics
- String API
User-Defined Types or Classes

As opposed to the primitive types, the user-defined types consist of data and methods that act on that data.

In modern programming environments there may be hundreds or thousands of classes available. Go to any reasonable bookstore and have a look at their JAVA-book selection . . .

Note, that sometimes it is useful to write a class with only static methods and no data.
Implementation vs Interface

It is important to distinguish between the actual implementation and the interface of the class. In order to use a class what we need is the interface or the API (application programmer’s interface) of that class. It may even be that the implementation is not available for us to study.

Note, that JAVA has a construct called interface, which has a more specific meaning than what is used here.
String Basics

String class can be used to store and process strings of characters. Every string is immutable, an instance can only be altered by assignment.

Every character in a string has its own position. A position is usually referred to as an index. Indices are counted from 0, so that in "Harri", the letter 'i' has an index 4.

Strings can be constructed from other strings by concatenation, which is performed by the concatenation operator ‘+’. 
More String Basics

Consider:

```java
String hi = "Hi!";
String who = "Harri";
String greeting = hi + " " + who;
```

`+` understands other types also:

```java
String one = "" + '1'; // char '1'
String two = '2'; // int value of char '2'
```
String API

In the following table for every command we list the return value.

**length ()**  The length of the string object.

**toLowerCase ()**  All characters in lower case.

**toUpperCase ()**  All characters in upper case.

**trim ()**  Leading and trailing white space removed.
String API Cndt.

charAt (Position)  Character in the string at the Position in the range 0, 1, \ldots \text{length()} − 1.

substring (Start)  Substring of the string object starting from the position Start.

substring (Start, End)  As above, but only up to position End.
String API Cntd.

**indexOf (AString)**  Position of the first occurrence of AString in the string object.

**indexOf (AString, Start)**  As above, but only from position Start onwards.

**lastIndexOf (AString)**  Position of the last occurrence of AString in the string object.

**compareTo (AString)**  Lexicographic comparison: equality is zero, argument first is positive, calling string first is negative.


**Special Characters**

A string constant is defined with inverted commas " ". If we want to include them as characters in a string, we must escape them:

```java
String quote = "Let me quote:\" ... \"";
```

- `\'` Single quote.
- `\ ` Backslash.
- `\n` Newline. Go to the beginning of the next line.
- `\r` Carriage return. Go to the beginning of the current line.
- `\t` Tab.