1 Administrivia

- Pset2 Final due Sunday, March ?? 2012
- Quiz 1 next week, Wednesday March 14
- Project Milestone 1 due Monday, March 19

2 Regular Expressions

See slides of Lecture 05 on Stellar.
Java Reference: docs.oracle.com/javase/tutorial/essential/regex/intro.html

Regular expressions can describe a set of strings based on common characteristics among each string in the set. They can be used to search, edit, or manipulate text and data.

Java supports regular expression functionality through the java.util.regex package.

There are two main classes from that package: Pattern and Matcher. (There is an exception class PatternSyntaxException as well).

Listing 1: Example Usage of Regular Expressions in Java

```java
Pattern p = Pattern.compile(regEx);
Matcher m = p.matcher(inputString);

while (m.find()) {
    System.out.format("Found " +
        " %s starting at " +
        "index %d and ending at index %d.%n",
        m.group(), m.start(), m.end());
}
```

We will practice the following examples in this recitation session.

- Email address validator
- Removing overenthusiastic punctuation from a string!!!
- Breaking a sentence into a set of words

3 Metacharacters

These are characters with special meaning.

. < > ( ) [ ] { } \ ^ - = $ ! | ? * +

Use \ to force a metacharacter to be treated normally.

4 Character Classes

- Example: [abc] → a, b or c
- Example: [^abc] → any character except a, b or c (negation)
- Example: [a-zA-Z] → range a,...,z or A,...,Z
5  Predefined Classes

- .  → any character
- \d  → [0-9] (digits)
- \D  → [^0-9] (negation)
- \s  → white space (tab, newline,...)
- \S  → [^\s] (negation)
- \w  → word character ([a-zA-Z0-9])
- \W  → [^w] (negation)

Remember to use "\d" in a Java program to represent the regular expression for a digit.

6  Boundary Matchers

- ^  → beginning of line
- $  → end of line (e.g., ^dog$)
- \b  → word boundary (e.g., \bdog\b will match with the "dog" in the string "the dog plays ")
- \B  → non-word boundary (e.g., \bdog\B will match with the "dog" in "the doggie plays ")

7  More Useful Stuff

- Quantifiers - you can match a pattern some n number of times (repetitions)
  - X?  → X once or none
  - X*  → X zero or more times
  - X+  → X once or more
  - X{n}  → X exactly n times
  - X{n,} → X at least n times
  - X{n,m}  → X at least n times, at most m times

- Some useful Java pointers
  - Patterns with flags in Java. E.g., Pattern.compile(regEx, Pattern.CASE_INSENSITIVE);
  - To quickly check for a pattern in a given input, use Pattern.matches(String, CharSequence);
    Example: Pattern.matches("\d", "1") returns true
  - Use the split( ) instance method in Pattern class to gather the text lying on either side of a
    pattern. Example:
    Pattern p = Pattern.compile(regEx);
    String [] items = p.split(input);