CS106A Handout #17
Winter 2013-2014 February 3, 2014

Section Handout 4

Based on a handout by Eric Roberts, Patrick Young, and Jeremy Keeshin

Problem One: The Wizard of Java

Below is a program that explores what happens when you combine parameter passing, return values, and objects. Trace through the execution of this program. What does it print out?

```
/*
 * File: TheWizardOfJava.java
 * A program that explores parameters, return values, primitives, and objects.
 */
import acm.program.*; // For ConsoleProgram
import acm.graphics.*; // For GPoint

public class TheWizardOfJava extends ConsoleProgram {
    public void run() {
        String dorothy = "Somewhere over the rainbow...";
        GPoint toto = new GPoint(2.718, 3.141);
        int scarecrow = 137;

        cowardlyLion(dorothy, toto);
        println("dorothy = " + dorothy);
        println("toto.getX() = " + toto.getX());

        tinMan(dorothy, scarecrow);
        println("dorothy = " + dorothy);
        println("scarecrow = " + scarecrow);

        toto = overTheRainbow(dorothy, toto);
        println("dorothy = " + dorothy);
        println("scarecrow = " + scarecrow);
        println("toto.getX() = " + toto.getX());
    }

    private void cowardlyLion(String dorothy, GPoint toto) {
        dorothy += " way up high";
        toto = new GPoint(1.61, 98.6);
    }

    private String tinMan(String dorothy, int scarecrow) {
        dorothy = "The scarecrow said " + scarecrow;
        scarecrow = 14;
        return dorothy;
    }

    private GPoint overTheRainbow(String dorothy, GPoint toto) {
        toto = new GPoint(99.9, 44.4);
        dorothy.toUpperCase();
        return toto;
    }
}
```

Problem Two: Adding Commas to Numeric Strings (Chapter 8, Exercise 13, page 290)

When large numbers are written out on paper, it is traditional—at least in the United States—to use commas to separate the digits into groups of three. For example, the number one million is usually written in the following form:

```
1,000,000
```

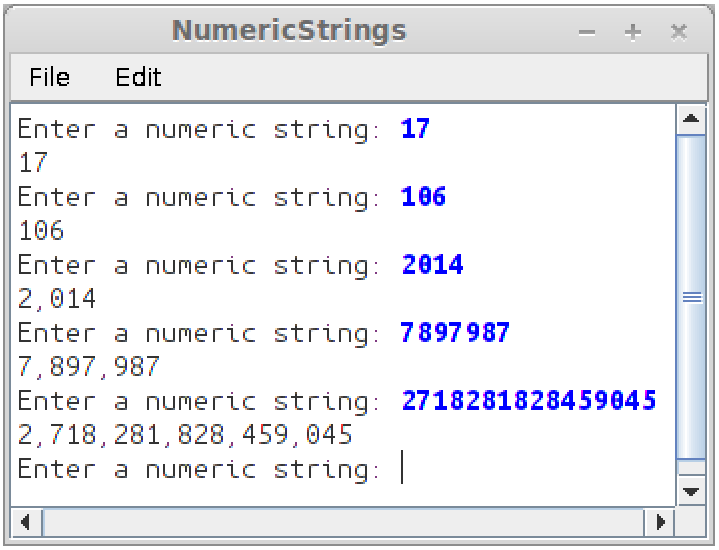
To make it easier for programmers to display numbers in this fashion, implement a method

```
private String addCommasToNumericString(String digits)
```

that takes a string of decimal digits representing a number and returns the string formed by inserting commas at every third position, starting on the right. For example, if you were to execute the main program

```
public void run() {
   while (true) {
      String digits = readLine("Enter a numeric string: ");
      if (digits.length() == 0) break;
      println(addCommasToNumericString(digits));
   }
}
```

your implementation of the **addCommasToNumericString** method should be able to produce the following sample run:



```
Enter a numeric string: 17
17
Enter a numeric string: 106
106
Enter a numeric string: 2014
2,014
Enter a numeric string: 7897987
7,897,987
Enter a numeric string: 2718281828459045
2,718,281,828,459,045
Enter a numeric string: 
```

Problem Three: Deleting Characters from a String

Write a method

```
public String removeAllOccurrences(String str, char ch)
```

that removes all occurrences of the character **ch** from the string **str**. For example, your method should return the values shown:

```
removeAllOccurrences("This is a test", 't')       returns  "This is a es"
removeAllOccurrences("Summer is here!", 'e')      returns  "Summr is hr"
removeAllOccurrences("---0---", '-')              returns  "0"
```

Problem Four: cOnVeRtInG a StRiNg To AlT cApS

Write a method

```
private String convertToAltCaps(String str)
```

which converts a string to alt caps, meaning you alternate between uppercase and lowercase. This style of typing was prevalent on the Internet in the late 90's (and, fortunately, has mostly disappeared). For example:

```
convertToAltCaps("hello")                  returns  "hElLo"
convertToAltCaps("section is awesome")     returns  "sEcTiOn Is AwEsOmE"
```