Chris Piech
CS 106A

Handout #24A
February 8, 2017

Solution to Section #4

Portions of this handout by Eric Roberts, Patrick Young, Jeremy Keeshin and Nick Troccoli

1. Adding commas to numeric strings

```
private String addCommasToNumericString(String digits) {
   String result = "";
   int len = digits.length();
   int nDigits = 0;
   for (int i = len - 1; i >= 0; i--) {
      result = digits.charAt(i) + result;
      nDigits++;
      if (((nDigits % 3) == 0) && (i > 0)) {
         result = "," + result;
      }
   }
   return result;
}
```

2. Deleting characters from a string

```
private String removeAllOccurrences(String str, char ch) {
   String result = "";
   for (int i = 0; i < str.length(); i++) {
      if (str.charAt(i) != ch) {
         result += str.charAt(i);
      }
   }
   return result;
}
```

A slightly different approach that involves a **`while`** loop instead of a **`for`** loop:

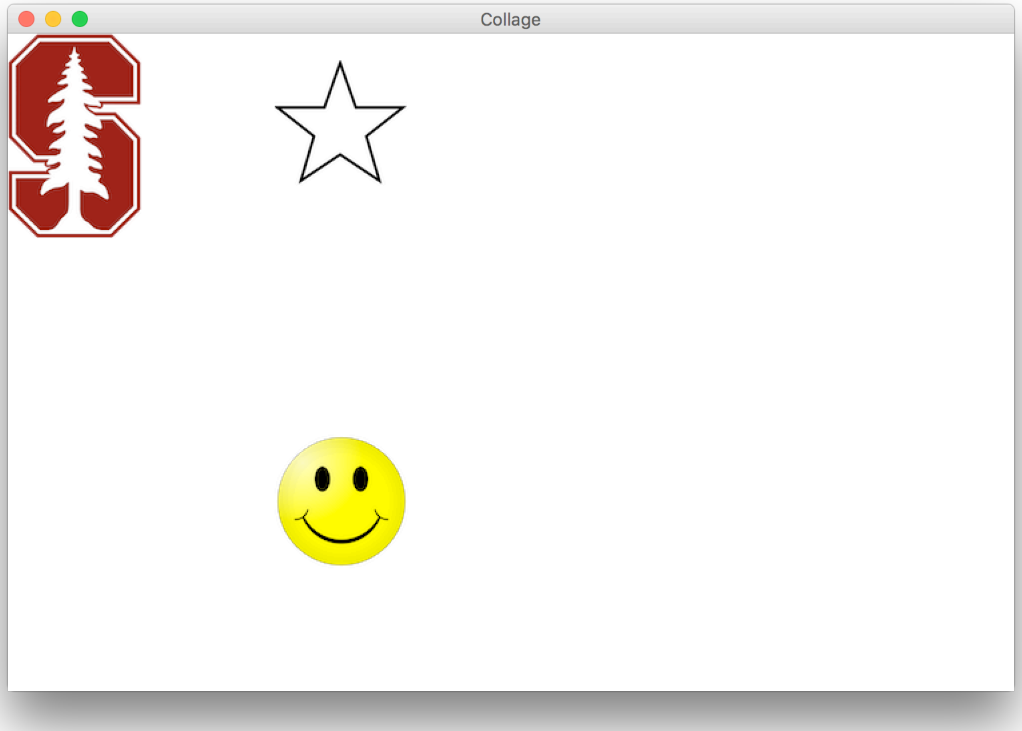
```
private String removeAllOccurrences(String str, char ch) {
   while (true) {
      int pos = str.indexOf(ch);
      if (pos >= 0) {
         str = str.substring(0, pos) + str.substring(pos + 1);
      } else break;
   }
   return str;
}
```

3. Converting a string to alternating capital letters

```
private String altCaps(String str) {
    String result = "";
    int counter = 0;
    for(int i = 0; i < str.length(); i++) {
       if (Character.isLetter(str.charAt(i))) counter++;

       if ((counter % 2) == 0) {
          result += Character.toUpperCase(str.charAt(i));
       }else{
          result += Character.toLowerCase(str.charAt(i));
       }
    }
    return result;
 }
```

4. Tracing method execution



Style Focus for Section 4

Common Programming Idioms: A programming *idiom* is a commonly used expression or pattern, like using ++ to increment a variable, or the loop-and-a-half. In this section we went over a common pattern of iterating through a string and building up a new result string. It is good to familiarize yourself with common programming idioms because you will see them appear in others' code, and it will make your own code better.