Name: _____

USC loginid (e.g., ttrojan):_____

CS 455 Midterm Exam 1
Spring 2013 [Bono]

Feb. 21, 2013

There are 5 problems on the exam, with 54 points total available. There are 7 pages to the exam, including this one; make sure you have all of them. There is also a one-page code handout that accompanies the exam. If you need additional space to write any answers, you may use the backs of exam pages (just direct us to look there).

Note: if you give multiple answers for a problem, we will only grade the first one. Avoid this issue by labeling and circling your final answers and crossing out any other answers you changed your mind about (though it's fine if you show your work).

Put your name and USC loginid at the top of the exam. Please read over the whole test before beginning. Good luck!

	value	score
Problem 1	10 pts.	
Problem 2	12 pts.	
Problem 3	8 pts.	
Problem 4	9 pts.	
Problem 5	15 pts.	
TOTAL	54 pts.	

Problem 1 [10 pts.]

Consider the following program:
(Note: more about `Point` class on code handout.)

```
public class Prob1 {
  public static void foo(Point a, Point b) {
    b = a;
    b.translate(5, 10);
    System.out.println(a + " " + b);
  }
  public static void main(String[] args) {
    Point p = new Point(3, 6);
    Point r = new Point(25, 40);
    foo(p, r);
    System.out.println(p + " " + r);
  }
}
```

Part A [6]. In the space below, draw a box-and-pointer diagram (a.k.a., memory diagram) showing all object variables, objects, and their state as they have changed during the code sequence. This includes showing `foo`'s parameters.

Part B [4]. What is printed by the code? For the purpose of this problem assume a `Point` is printed as follows: `[x, y]`

Problem 2 [12 pts]

Complete the implementation of the class `FibonacciGenerator`, which generates the Fibonacci sequence.

The Fibonacci sequence starts as follows: 1, 1, 2, 3, 5, 8, 13, . . .

The rules for generating the sequence are, where f_k, below, means the k'th Fibonacci number:

$f_1 = 1$

$f_2 = 1$

for any $k > 2$, $f_k = f_{k-1} + f_{k-2}$

So, in the sequence shown above, the third value is 1 + 1 = 2; the fourth value is 1 + 2 = 3, and the fifth value is 2 + 3 = 5, etc.

Here is some example code that uses this class. Given a value, *n*, it prints out the first *n* Fibonacci numbers, one per line:

```
public static void printFibs(int n) {
   FibonacciGenerator fib = new FibonacciGenerator();
   for (int i = 1; i <= n; i++) {
      System.out.prinln(fib.next());
   }
}
```

Hint: you will only need the last two values in the sequence to compute the next one.

Turn the page for space for your answer (includes the class interface for you) →

Problem 2 (cont.)

```
/**
   Generates the Fibonacci sequence.
*/
public class FibonacciGenerator{









   /**
      Create a Fibonacci Generator
   */
   public FibonacciGenerator() {









   }


   /**
     Generate the next number in the Fibonacci sequence.
   */
   public int next() {









   }

}
```

Problem 3 [8 pts.]

Implement the static method `readScores`, which reads in an indeterminate number of non-negative student scores from a Scanner, `in`, and stores them into an `ArrayList` which gets returned. `-1` is the sentinel for the input sequence. Here are some examples of legal input for this method:

Example 1: `94 99 86 -1`

(returns the arraylist containing [94, 99, 86])

Example 2: `-1`

(returns the empty arraylist)

Example 3: `68 73 55 76 -1`

(returns the arraylist containing [68, 73, 55, 76])

You may assume the input is in the correct format (i.e., no error-checking required).

```
/**
   Returns an arraylist of non-negative scores read from "in".
   Uses -1 as a sentinel.
   @param in  the Scanner to read from (like most params, you should assume it's
       already initialized by the caller)
   @returns  the scores in the order they were read in.
*/
public static ArrayList<Integer> readScores(Scanner in) {
```

Problem 4 [9 pts. total]

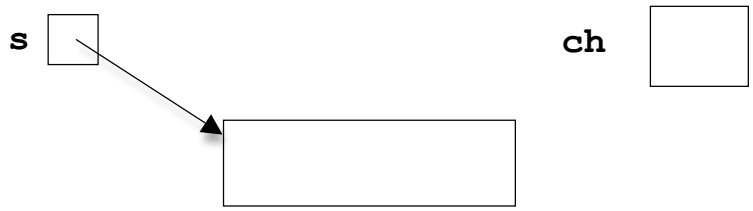
Part A (4). Complete the following static boolean method, `contains`, by completing the return statement below with the correct boolean expression.

```
/**
   Returns true if and only if string s contains the char ch.
*/
public static boolean contains(String s, char ch) {
   int i = 0;
   while (i < s.length() && ch != s.charAt(i)) {
      i++;
   }

   return _____________________________________________________ ;
}
```

For parts B & C, consider the method below, `contains2`, that is the same as the code above, but with the order of the conditions in the `&&` reversed (changed part shown in bold). This version of the code is not correct.

```
public static boolean contains2(String s, char ch) {
   int i = 0;
   while (ch != s.charAt(i) && (i < s.length()) {
      i++;
   }

   return      <assume the return expression you gave in part a>          ;
}
```

Part B (2). Give sample input for `contains2`, shown above, such that it doesn't work (i.e., but for which `contains` would work):



Part C (3). What `contains2` do on the input you gave in part B? Why? (be specific)

Problem 5 [15 pts. total]

Implement the static method `rotateLeft`, which takes an array and a value *k*, and returns an array like the one passed in, but with the values rotated left by *k* positions. A rotation moves values that "fall out" of the front of the array around to the end of the array. ***The original array is unchanged by the method.*** Here are some examples:

	`arr:`	**`k:`**	return value of **`rotateLeft(arr):`**
Ex1:	`1 2 3 4 5 6 7`	`3`	`4 5 6 7 1 2 3`
Ex2:	`1 2 3 4 5 6 7`	`10`	`4 5 6 7 1 2 3`
Ex3:	`1 2 3 4 5 6 7`	`1`	`2 3 4 5 6 7 1`
Ex4:	`12 2 9`	`0`	`12 2 9`
Ex5:	`12 2 9`	`3`	`12 2 9`

For full credit, your answer must move each data value from the array at most once.

```
/**
   Returns an array with the same values as arr, but with the values rotated
   left by k positions.  arr is unchanged by the method.
   PRE: arr.length > 0
        and k >= 0
*/
public static int[] rotateLeft(int[] arr, int k) {
```