ASSIGNMENT #2 SOLUTION

```
import java.util.*;                                          // Need this for Scanner.

class ThreePointsOnACircle
  {
      /* DETERMINE IF k IS ON THE ARC WHEN MOVING CLOCKWISE FROM j TO i ON A CIRCLE OF SIZE n. */

    static Scanner keyboard = new Scanner (System.in);       // Global, to avoid repeated re-instantiation each time getAndVerifyInput is called.

    static final int REUSE_PREVIOUS_INPUT = -1;              // Value input by user used to signal that previously entered value is to be re-used.

    public static void main (String[] args)
      {
        int n, i, j, k;                                      // n is size of circle, and i, j, and k are the circular coordinates of 3 points on it.

        n = i = j = k = REUSE_PREVIOUS_INPUT;                // This initialization enables us to determine if a previous value was ever entered for these values.

        while (true)                                         // Repeat indefinitely until user signals termination.
          {
            // for each of n, i, j, and k, prompt user, get value, and verify that the value is within bounds

            n= getAndVerifyInput ("Enter size of circle, n", n, 0, Integer. MAX_VALUE);

            if (n == 0) return;                              // User has signaled termination, so return back to JVM to terminate program execution.

            i= getAndVerifyInput ("Enter first point, i", i, 0, n);

            j= getAndVerifyInput ("Enter second point, j", j, 0, n);
```

```
        k= getAndVerifyInput ("Enter third point, k", k, 0, n);

        // determine if n, i, j, and k satisfy all 4 necessary conditions and display appropriate message

        if ( n > 0  &&  i ≥ 0  &&  i ≤ n  &&  j ≥ 0  &&  j ≤ n  &&  k ≥ 0  &&  k ≤ n  &&  i ≠ j  &&          // Conditions 1–3.
            (k ≤ i < j  ||  i < j < k  ||  j < k ≤ i) )                                                      // Condition 4.

            System.out.println ("\n\n" + k + " lies on the arc from " + j + " to " + i + " when moving clockwise on a circle
                                 of size " + n);
        else

            System.out.println ("\n\n" + k + " does not lie on the arc from " + j + " to " + i + " when moving clockwise on a
                                 circle of size " + n);
    }                                                                                                        // End of while (true).
}                                                                                                            // End of main method.

/* PROMPT USER, RECEIVE INPUT, AND VERIFY THAT IT'S WITHIN RANGE; REPEAT IF NECESSARY */

static int getAndVerifyInput (String userPrompt, int previousValue, int lowerBound, int upperBound)

{
  int input;                                                                                                 // Value input by user.

  do
    {
      System.out.print (userPrompt  + " ('" + REUSE_PREVIOUS_INPUT + "'" + " to use previously entered value): ");

      input = keyboard.nextInt();

      if (input == REUSE_PREVIOUS_INPUT)  input = previousValue;          // Adjust input if user wants previously entered value is to be re-used.
```

```
        }
        while (input == REUSE_PREVIOUS_INPUT || input < lowerBound || input > upperBound);  // Re-prompt & re-input until we get valid value.

        return input;                                                                       // Validated input is returned to caller.
    }                                                                                       // End of getAndVerifyInput method.
}                                                                                           // End of class ThreePointsOnACircle.
```