```
// Blair Hagen
// Lab 1

// Honor Code: As a student at Union College, I am part of a community
that values intellectual effort, curiosity and discovery. I understand
that in order to truly claim my educational and academic achievements,
I am obligated to act with academic integrity. Therefore, I affirm
that I will carry out my academic endeavors with full academic
honesty, and I rely on my fellow students to do the same.

/* Runs each of the methods in the Analyzer class in turn.
 *
 */
public class Client {

        public static void main(String[] args) {
                Analyzer tester = new Analyzer();
                double min, max;
                double[] buyingData = new double[10];
                for(int i=0; i<buyingData.length; i++) {
                        buyingData[i]=((long)
(100*(Math.random()*100)))/100.0;
                }
                System.out.println("***PART 1 - numberCruncher");
                tester.numberCruncher();
        System.out.println("\n");

                min=9; max=50;  // change these to test
                System.out.println("###PART 2 - purchaseAnalyzer:
min=" + min + "    max=" + max);
                tester.purchaseAnalyzer(buyingData, min, max);
        System.out.println("\n");

                min=8; max=85;  // change these to test
                System.out.println("^^^PART 3 - inDepthAnalyzer:
min=" + min + "    max=" + max);
                tester.inDepthAnalyzer(buyingData, min, max);

                tester.printer(buyingData);
        }

}


/* Class to analyze simulated purchase trends from a department store.
 * Blair Hagen
 * Lab 1
 */
public class Analyzer {
```

```
        // Declares four local variables and initializes them, then
prints them with context.
        public void numberCruncher()
        {
                int polyPerim = 14;
                double savingsBalance = 3400.59;
                char subwayLine = 'E';
                boolean isMarried = false;

                System.out.println();
                System.out.println("The perimeter of the polygon is:
" + polyPerim);
                System.out.println("The balance of the savings
account is: $" + savingsBalance);
                System.out.println("The current subway line is line "
+ subwayLine);
                System.out.print("Current marriage status is: " +
isMarried);
        }

        // Analyzes the first three purchases and returns if the day
has been good or bad based on sale values.
        public void purchaseAnalyzer(double[] purchase, double min,
double max)
        {

                if (purchase[0] > max && purchase[1] > max &&
purchase[2] > max)
                {
                        System.out.println();
                        System.out.println("Great Morning!");
                        System.out.println("Purchase 1: $" +
purchase[0]);
                        System.out.println("Purchase 2: $" +
purchase[1]);
                        System.out.print("Purchase 3: $" +
purchase[2]);

                }
                else if (purchase[0] <= min || purchase[1] <= min ||
purchase[2] <= min)
                {
                        System.out.println();
                        System.out.print("Bad Morning");
                }
        }

        // Prints the first good purchase of the day, then prints the
number of good purchases made before the first bad one.
        public void inDepthAnalyzer(double[] purchase, double min,
```

```
double max)
        {
                //Job 1
                boolean isGoodPurchFound = false;
                int index = 0;

                while (index < purchase.length && isGoodPurchFound ==
false)
                {
                        if (purchase[index] > max)
                        {
                                isGoodPurchFound = true;
                                System.out.println();
                                System.out.println("First best
purchase of the day: $" + purchase[index]);
                                System.out.println("The first best
purchase was purchase index " + index);
                        }
                        index++;
                }

                if (isGoodPurchFound == false)
                {
                        System.out.println();
                        System.out.println("No good purchases were
found");
                }

                //Job 2
                index = 0;
                int numGoodPurch = 0;
                boolean badPurchFound = false;

                while (index < purchase.length && badPurchFound ==
false)
                {
                        if (purchase[index] > max)
                        {
                                numGoodPurch++;
                        }
                        else if (purchase[index] <= min)
                        {
                                badPurchFound = true;
                        }
                        index++;
                }

                if (badPurchFound == true)
                {
                        System.out.println();
```

```
                        System.out.println("There were " +
numGoodPurch + " good purchases before the first bad one");
                }
                else
                {
                        System.out.println();
                        System.out.println("There were " +
numGoodPurch + " good purchases and no bad ones");
                }
        }

        /** a really dumb way of printing the entire array
         *
         *  purchase is a chronological collection of purchases
         */
        public void printer(double[] purchase)
        {
                System.out.println();
                System.out.println("purchase[0]: " + purchase[0]);
                System.out.println("purchase[1]: " + purchase[1]);
                System.out.println("purchase[2]: " + purchase[2]);
                System.out.println("purchase[3]: " + purchase[3]);
                System.out.println("purchase[4]: " + purchase[4]);
                System.out.println("purchase[5]: " + purchase[5]);
                System.out.println("purchase[6]: " + purchase[6]);
                System.out.println("purchase[7]: " + purchase[7]);
                System.out.println("purchase[8]: " + purchase[8]);
                System.out.println("purchase[9]: " + purchase[9]);
        }

}
```