RAMAIAH
Institute of Technology

Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of Computer Science & Engineering

Object Oriented Programming Laboratory (CSL37)

Semester:III **Week #: 02** **Section:A,B,C**

.

Arrays and Control Structures

Create a File *Sales.java that should contain a* Java program that prompts for and reads in the sales for each of 5 salespeople in a company. It should then prints out the id and amount of sales for each salesperson and the total sales. Now modify the program as follows:

1. Compute and print the average sale. (You can compute this directly from the total; no loop is necessary.)
2. Find and print the maximum sale. Print both the id of the salesperson with the max sale and the amount of the sale, e.g., "Salesperson 3 had the highest sale with $4500." Note that you don't need another loop for this; you can do it in the same loop where the values are read and the sum is computed.
3. Do the same for the minimum sale.
4. After the list, sum, average, max and min have been printed, ask the user to enter a value. Then print the id of each salesperson who exceeded that amount, and the amount of their sales. Also print the total number of salespeople whose sales exceeded the value entered.
5. The salespeople are objecting to having an id of 0—no one wants that designation. Modify your program so that the ids run from 1–5 instead of 0–4. Do not modify the array—just make the information for salesperson 1 resides in array location 0, and so on.
6. Instead of always reading in 5 sales amounts, at the beginning ask the user for the number of sales people and then create an array that is just the right size. The program can then proceed as before.

```
// ********************************************************************
// Sales.java
// Reads in and stores sales for each of 5 salespeople. Displays
// sales entered by salesperson id and total sales for all salespeople.
// ********************************************************************
import java.util.Scanner;
public class Sales
{
      public static void main(String[] args)
      {
            final int SALESPEOPLE = 5;
            int[] sales = new int[SALESPEOPLE];
```

```
        int sum;
        Scanner scan = new Scanner(System.in);
        for (int i=0; i<sales.length; i++)
        {
              System.out.print("Enter sales for salesperson"+i+":");
              sales[i] = scan.nextInt();
        }
        System.out.println("\nSalesperson Sales");
        System.out.println("                      ");
        sum = 0;
        for (int i=0; i<sales.length; i++)
        {
              System.out.println(" " + i + " " + sales[i]);
              sum += sales[i];
        }
        System.out.println("\nTotal sales: " + sum);
}
```

Grading Quizzes

Write a program that grades arithmetic quizzes as follows:

1. Ask the user how many questions are in the quiz.
2. Ask the user to enter the key (that is, the correct answers). There should be one answer for each question in the quiz, and each answer should be an integer. They can be entered on a single line, e.g., 34 7 13 100 81 3 9 10 321 12 might be the key for a 10-question quiz. You will need to store the key in an array.
3. Ask the user to enter the answers for the quiz to be graded. As for the key, these can be entered on a single line. Again there needs to be one for each question. Note that these answers do not need to be stored; each answer can simply be compared to the key as it is entered.
4. When the user has entered all of the answers to be graded, print the number correct and the percent correct. When this works, add a loop so that the user can grade any number of quizzes with a single key. After the results have been printed for each quiz, ask "Grade another quiz? (y/n)."

RAMAIAH
Institute of Technology

Reversing an Array

Write a program that prompts the user for an integer, then asks the user to enter that many values. Store these values in an array and print the array. Then reverse the array elements so that the first element becomes the last element, the second element becomes the second to last element, and so on, with the old last element now first. Do not just reverse the order in which they are printed; actually change the way they are stored in the array. Do not create a second array; just rearrange the elements within the array you have.

(Hint: Swap elements that need to change places.) When the elements have been reversed, print the array again.

1. Write a program in java to illustrate the use of for each loop by searching an element in the unsorted array.
2. Write a program in java to illustrate the use of continue, labeled continue by printing the following output.

    0

    0 1

    0 2 4

    0 3 6

1.
2.