King Abdulaziz University
Faculty of Engineering
Electrical and Computer Engineering
EE-202-Object-Oriented Computer Programming

Lab-9

Q-1 Write a program that grades arithmetic quizzes as follows:

- Ask the user how many questions are in the quiz.
- 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.
- 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.
- 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)."

Sample Output

```
Quiz Grading

Enter the number of questions on the quiz: 4
Enter the answer key: 1 2 3 4
Enter the student answers: 1 2 3 4
4 correct for a grade of 100.00 %

Grade another quiz (y/n)? y

Enter the student answers: 1 2 2 1
2 correct for a grade of 50.00 %

Grade another quiz (y/n)? y

Enter the student answers: 1 1 1 1
1 correct for a grade of 25.00 %

Grade another quiz (y/n)? y

Enter the student answers: 5 5 5 5
0 correct for a grade of 0.00 %

Grade another quiz (y/n)? n
```

Answer:

quiz.java

```java
import java.util.Scanner;

public class quiz {

    public static void main(String[] arg) {
        Scanner inp = new Scanner(System.in);

        System.out.print("Enter number of questions in the quiz: ");
        final int qNum = inp.nextInt();

        System.out.print("Enter quiz answers key: ");
        int[] ansKey = new int[qNum];
        for(int j = 0; j < qNum; j++) {
            ansKey[j] = inp.nextInt();
        }

        boolean isTrue = true;

        do {
            int correct = 0;

            System.out.print("Enter quiz answers: ");
            int[] ansStudent = new int[qNum];
            for (int j = 0; j < qNum; j++) {
                ansStudent[j] = inp.nextInt();
            }

            for (int j = 0; j < qNum; j++) {
                if (ansKey[j] == ansStudent[j]) {
                    correct++;
                }
            }

            double ratio = ((double) correct) / qNum;
            System.out.printf("You got %.2f %s of answers correct.%n", (ratio * 100), "%");

            System.out.print("\nGrade another quiz (y/n)? ");
            String another = inp.next();
            switch (another) {
                case "y":
                    break;
                case "n":
                    isTrue = false;
                    break;
                default:
                    System.out.print("invalid input\n");
                    isTrue = false;
            }
        } while(isTrue);
    }
}
```

```
Enter number of questions in the quiz: 4
Enter quiz answers key: 1 2 2 2
Enter quiz answers: 1 2 2 1
You got 75.00 % of answers correct.

Grade another quiz (y/n)?y
Enter quiz answers: 1 2 2 2
You got 100.00 % of answers correct.

Grade another quiz (y/n)?n
```

Q-2 File Sales.java contains a Java program that prompts for and reads in the sales for each of 5 salespeople in a company. It then prints out the id and amount of sales for each salesperson and the total sales. Study the code, then compile and run the program to see how it works. Now modify the program as follows:

- Compute and print the average sale. (You can compute this directly from the total; no loop is necessary.)
- 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.
- Do the same for the minimum sale.
- 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.
- 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 reside in array location 0, and so on.
- 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.

Sample Output

```
Enter the number of sales people: 4
Enter sales for salesperson 1: 1000
Enter sales for salesperson 2: 2000
Enter sales for salesperson 3: 2500
Enter sales for salesperson 4: 3000

Salesperson   Sales
--------------------
    1             1000
    2             2000
    3             2500
    4             3000

Total sales: 8500
Average sale: 2125
Maximum sales 3000 by Salesperson 4
Minimum sales 1000 by Salesperson 1

Enter a sales amount: 1500

List of sales over 1500

Salesperson        Sales
   2                    2000
   3                    2500
   4                    3000

3 salespeople had sales over 1500
```

Sales.java

```java
package lab9;
import java.util.Scanner;

public class Sales {
    public static void main(String[] args) {
        Scanner inp = new Scanner(System.in);
        System.out.print("Enter number of sales persons: ");

        final int size = inp.nextInt();

        int[] salesPersons = new int[size];

        for(int i = 0; i < size; i++) {
            System.out.printf("Enter sales for sales person %d: ", i+1);
            salesPersons[i] = inp.nextInt();
        }

        System.out.print("\nSalesperson\t\tSales\n" +
                "--------------------\n");

        for(int j = 0; j < size; j++) {

            System.out.printf("%2d\t\t\t%-5d\n",j+1,salesPersons[j]);
        }

        System.out.print("\nEnter a sales amount: ");
        int min = inp.nextInt();

        System.out.printf("\nList of sales over %d%n", min);

        System.out.print("\nSalesperson\t\tSales\n" +
                "--------------------\n");

        int z = 0;
        for(int k = 0; k < size; k++) {
            if (salesPersons[k] >= min) {
                int index = 0;
                for(int q = 0; q < size; q++) {
                    if (salesPersons[q] == salesPersons[k]) {
                        index = q;
                        break;
                    }
                }
                System.out.printf("%2d\t\t\t%-5d\n", index + 1, salesPersons[k]);
                z++;
            }
        }
        System.out.printf("\n%d salesperson's had sales over %d",z,min);
    }
}
```

```
Enter number of sales persons: 4

Enter sales for sales person 1: 80000

Enter sales for sales person 2: 3500

Enter sales for sales person 3: 22222

Enter sales for sales person 4: 10000


Salesperson     Sales

--------------------

 1           80000

 2           3500

 3           22222

 4           10000


Enter a sales amount: 5000


List of sales over 5000


Salesperson     Sales

--------------------

 1           80000

 3           22222

 4           10000


3 salesperson's had sales over 5000
```