

Hands-on Experiment # 7 : Worksheet

Section _____ Date _____

No more than 3 students per one submission of this worksheet.

Student ID _____ Name _____

Student ID _____ Name _____

Student ID _____ Name _____

Part A: Score Look Up Program

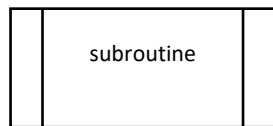
This time, you will write a score look up Java application that reads *score.csv* (from the previous experiment) which list exam scores of 1,000 students. The user of the program can enter a student ID and the program shows the scores from the 5 questions as well as the total score associated with that student ID.

Objectives:

- Practice creating Java methods using correct syntax.
- Practice factorizing (dividing) the program into methods with distinct functionalities.
- Try making the program as “readable” as possible.

Instructions:

- Obtain understanding of the program by studying *L07Design.pdf*.
 - The file contains flow charts detailing some parts of the program.
 - Pay attention to the “subroutine (or subprogram)” (as shown below) blocks. These should be method calls.



- Complete the program by adding codes to *ScoreLookup.java*.

Key challenge → Try to divide tasks into methods so that the resulting code is as “readable” as you can.

Have you been able to complete the program? If not, what were the problems?

Yes, I can.

Does it work correctly in all cases? If not, what are the cases those your program does not work correctly?

Yes, it does.

Include the screenshots below.

```
Enter Student ID: 5600574721
          Q1  Q2  Q3  Q4  Q5  TOTAL
5600574721  3   3   4   8   4   22
Main Menu
L) Look up score
Q) Quit
>>> █
```

How many methods have you created in the program?

3

List all the methods you created in the table below.

#	Return Type	Method Name	Method Signature	Description
1	String	constructLookUpStringFromFile	constructLookUpStringFromFile(String)	Read the csv file whose name is specified in the argument list of the method and construct a String containing all lines of the file. The String is then returned.
2	Char	showMainMenu	showMainMenu()	Construct main menu and receive input from keyboard with the type of char. Then, the char is return.
3	No return(void)	commenceLookUpProcedure	commenceLookUpProcedure(String)	Create subroutine that find the student's id that match with the input from keyboard and show all the score. and the sum of the scores.

Do you think each of your methods is short and explicit enough so that the method can be understood easily? If not, what do you think should be improved?

Yes, it is short and simple for people to read and understand the code easily.

List all your source code here.

```
import java.util.Scanner;
import java.io.*;

public class ScoreLookup {
    public static void main(String[] args) throws IOException {
        String lookupString = constructLookupStringFromFile("score.csv");
        boolean toQuit = false;
        do {
            char choice = showMainMenu();
            switch (choice) {
                case 'L':
                    commenceLookupProcedure(lookupString);
                    break;
                case 'Q':
                    toQuit = true;
                    break;
                default:
                    System.out.println("Invalid choice. Quitting.");
                    toQuit = true;
            }
        } while (!toQuit);
    }

    // Add static methods here.
    // The methods include, but not limited to
    // - constructLookupStringFromFile(String)
    // - showMainMenu()
    // - commenceLookupProcedure(String)
    public static String constructLookupStringFromFile(String a) throws IOException {
```

```
Scanner sc = new Scanner(new File(a));
sc.useDelimiter("\\r\\n|,");
sc.nextLine();
String score = "";
while (sc.hasNext()) {
    String id = sc.next();
    int q1 = sc.nextInt();
    int q2 = sc.nextInt();
    int q3 = sc.nextInt();
    int q4 = sc.nextInt();
    int q5 = sc.nextInt();
    score += id + " " + q1 + " " + q2 + " " + q3 + " " + q4 + " " + q5 + ",";
}
return score;
}

public static char showMainMenu() {
    Scanner kb = new Scanner(System.in);
    System.out.println("Main Menu");
    System.out.println("L) Look up score");
    System.out.println("Q) Quit");
    System.out.print(">>> ");
    char input = kb.next().charAt(0);
    return input;
}

public static void commenceLookUpProcedure(String x) {
    Scanner kb = new Scanner(System.in);
    System.out.print("Enter Student ID: ");
    String lookup = kb.next();
    int startIndex = x.indexOf(lookup);
    int endIndex = x.indexOf(";", startIndex);
    String z = x.substring(startIndex, endIndex);
    Scanner as = new Scanner(z);
    String id = as.next();
    int q1 = as.nextInt();
```

```
int q2 = as.nextInt();
int q3 = as.nextInt();
int q4 = as.nextInt();
int q5 = as.nextInt();
System.out.println("      Q1 Q2 Q3 Q4 Q5 TOTAL");
System.out.println(
    id + " " + q1 + " " + q2 + " " + q3 + " " + q4 + " " + q5 + " " + (q1 + q2 + q3 + q4 +
q5));
}
}
```

Submit this worksheet (by only one member of the group) via <http://www.myCourseVille.com> (Assignments > Hands-on Experiment # 7) **within the day after your lecture.**