

Hands-on Experiment # 8 : Worksheet

Section 2 Date 22/03/2018

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

Student ID 6031851521 Name Sarun Nuntaviriyakul

Student ID 6031848721 Name Watcharin Kriengwatana

Student ID 6031847021 Name Wasuthon Klyhirun

Part A: Score Look Up Program

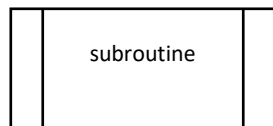
This time, you will write a score look up Java application that reads *score.csv* (from the previous experiment) which list exam scores of 1,000 students. The user of the program can enter a student ID and the program shows the scores from the 5 questions as well as the total score associated with that 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 *L08Design.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

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

Include the screenshots below.

```
C:\Users\Him\Java (Him)\week8>java ScoreLookup
-----
Main Menu
-----
L)      Look up score
Q)      Quit
>>L
Enter student ID: 5600334721
          Q1      Q2      Q3      Q4      Q5      Total
5600334721  5      1      2      7      7      22
-----
Main Menu
-----
L)      Look up score
Q)      Quit
>>La
Invalid choice. Quitting.

C:\Users\Him\Java (Him)\week8>java ScoreLookup
-----
Main Menu
-----
L)      Look up score
Q)      Quit
>>Q

C:\Users\Him\Java (Him)\week8>
```

```

-----
Main Menu
-----
L)      Look up score
Q)      Quit
>>L
Enter student ID: 123456789
ID not found

```

How many methods have you created in the program?

4

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()	Show main menu and allow user to choose what to do
3	Void	commenceLookUpProcedure	commenceLookUpProcedure(String score)	Let user enter id and check if it is in database
4	Void	printScore	printScore(String id, String q1, String q2, String q3, String q4, String q5)	Print score

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

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);

}

public static String constructLookupStringFromFile(String file) throws IOException{

    String str = "";

    Scanner sc = new Scanner(new File(file));

    sc.nextLine();

    while(sc.hasNext()){

        str += sc.next() + ",";

    }

    return str;

}
```

```
}

public static char showMainMenu(){

    Scanner sc = new Scanner(System.in);

    System.out.println("-----");

    System.out.println("Main Menu");

    System.out.println("-----");

    System.out.println("L) \t Look up score");

    System.out.println("Q) \t Quit");

    System.out.print(">>");

    String choice = sc.next();

    return choice.length() == 1 ? choice.charAt(0) : 'a';

}

public static void commenceLookUpProcedure(String score){

    Scanner sc = new Scanner(score).useDelimiter(",");

    System.out.print("Enter student ID: ");

    Scanner kb = new Scanner(System.in);

    String input = kb.nextLine();

    boolean found = false;

    int total = 0;

    while(sc.hasNext()){

        String id = sc.next();

        if(id.equals(input)){

            printScore(id, sc.next(), sc.next(), sc.next(), sc.next(), sc.next());

            found = true;

            break;

        }

    }

}
```

```
        }

    }

    if(!found)

        System.out.println("ID not found");

    }

    public static void printScore(String id, String q1, String q2, String q3, String q4, String q5){

        System.out.println("\t\tQ1\tQ2\tQ3\tQ4\tQ5\tTotal");

        int total = Integer.parseInt(q1) + Integer.parseInt(q2) + Integer.parseInt(q3) +
Integer.parseInt(q4) + Integer.parseInt(q5);

        System.out.printf("%s\t%s\t%s\t%s\t%s\t%s\t%d\n",id,q1,q2,q3,q4,q5,total);

    }

}
```

Submit this worksheet (by only one member of the group) via <http://www.myCourseVille.com> (Assignments > Hands-on Experiment # 8) before noon of the day after your lecture.