King Abdulaziz University
Faculty of Computing and Information Technology
Summer (2020)

Course Code: CPCS 203 Course Name: Programming II

Assignment # 1 (Exam Grading App.)

Assigned Date	**Tuesday, 09/06/2020**
Delivery Date and Time	**Saturday, 20/06/2020 at 11:59 PM**

WARNING:

- This program must ONLY be submitted on the Blackboard!
- This assignment is worth 10% of the overall module marks (100%).
- NO assignment will be accepted after **Saturday, 20/06/2020 at 11:59 pm** for any reasons.
- For discussion schedule, check the teacher name, date and time on the blackboard. **Further information is provided in the course syllabus.**

Objectives

- **Learn to use and implement multidimensional array.**
- **Performing procedure on array elements.**
- **Learn to use and implement String, File I/O (Reading/Writing from/to files).**

Delivery:

- Submit your assignment on the Blackboard ONLY.
- **Make sure to add your name / IDs / Section / course name / Assignment number, as comment at the beginning of your program**.

Description:

This is a **Grade app program** as shown in **figure 1**. KAU wants you to develop a program that automatically calculate grades of different assessments for a course. This program helps the instructor to find the sum of marks all Exams and the generate reports related to specific or all courses.

The Data Creation of the Program

Grades For Exam Grading System for the Course CPCS202

Separate 1-D String Array

	Exam Type						
Student Name	Exam1(10)	Exam2(15)	Final(25)	Lab Exams(15)	Assignments(30)	Quiz(5)	∑ (Total)
Fatima	9	15	23	14	25	5	91
Laila	8	13	18	13	28	4	84
Maryam	7	12	16	12	23	5	75
Khalida	8	12	15	10	22	3	70
Zayneb	5	11	15	11	20	2	64

Separate 2-D String Array

Max. marks 91 Fatima Topper for 202

Separate 3-D int Array to store marks obtained

Figure 1

As you can see from the figure 1, you will need to make different arrays to store all the data to be used in the program, for example you need one-dimension array to store assessment types, such as [Exam1, Exam2, Final Exam, Lab Exams, Assignments and Quizzes], more details are available in next page.

Exam Grading program must store the following data:

- ✓ Courses offered in the Faculty.
- ✓ Students registered in different courses.
- ✓ Assessment type.
- ✓ Marks obtained in each assessment for each student in different courses.
- ✓ Topper students for each course.
- ✓ Topper students from all courses.

Basic Requirements:

You will use text I/O to read input from a given input file[input.txt]. Make sure the file exists or display a message that the file does not exist. The file consists of:

Five (5) integers to determine array size for course and total students registered in the course [see the input file]:

A. The first number (4) in the file refers to the number of courses in the System [means system will accept ONLY Four courses]
CPCS202, CPCS203, CPCS204 and CPCS353.

B. The second number (5) refers to the number of Students registered in the course CPCS202 [means system will accept ONLY 5 students' details for CPCS202]

C. The third number (3) refers to the number of Students registered in the course CPCS203 [means system will accept ONLY 3 students' details for CPCS203]

D. The fourth number (4) refers to the number of Students registered in the course CPCS204 [means system will accept ONLY 4 students' details for CPCS204]

E. The fifth number (3) refers to the number of Students registered in the course CPCS353 [means system will accept ONLY 3 students' details for CPCS353]

Note: The program shall generate a ***text file as the output*** called "print.txt" that contains the results of the commands written in the input file – see print file for more details. The program shall load data dynamically from the given text file "input.txt" and ***user is not involved*** in data entry.

The commands you will have to implement are as follows:

a) Add_Course – Your program must read Courses code and stores in 1-D array to be used in the system. [see input.txt] for example:

Add_Course CPCS202 CPCS203 CPCS204 CPCS353

In above line **Add_Course** is a command and CPCS202 CPCS203 CPCS204 CPCS353 are course codes.

b) Add_ExamType – Your program must read Exam types and stores in an array to be used in the system. [see input.txt]

Add_ExamType Exam1 Exam2 FinalExam LabExams Assignments Quiz

In above line **Add_ExamType** is a command and **Exam1 Exam2 FinalExam LabExams Assignments Quiz** are six assessments (exam types) for a course.

c) Add_Student_For_Course – Your program must read Students name and store in an array to be used in the system. [see input.txt]

Add_Student_For_Course 0 Fatima Laila Maryam Khalida Zayneb

In above line **Add_Student_For_Course** is a command and **0** is the index, which indicate CPCS202 and **Fatima, Laila** etc are the name of students registered in CPCS202, similarly;

.

.

Add_Student_For_Course 3 Hina Reem Mona

In above line **Add_Student_For_Course** is a command and **3** is the index, which indicate course CPCS353 and **Hina Reem Mona** are the name of students registered in CPCS353.

d) **Add_Marks** - Your program must read marks of the students obtained from different assessment(exams) of a course and stores them in an array to be used in the program from the given input file. [see input.txt]

```
Add_Marks
9 15 23 14 25 5
8 13 18 13 28 4
7 12 16 12 23 5
8 12 15 10 22 3
5 11 15 11 20 2
8 13 24 13 24 5
7 15 23 14 30 5
3 10 20 12 22 5
9 15 25 14 28 4
8 14 24 15 26 5
9 15 23 13 27 5
6 12 22 15 29 4
3 10 22 12 25 3
7 15 23 13 28 4
8 13 24 14 23 5
```

In above lines **Add_Marks** is a command and in Yellow color indicate marks for five students in the course **CPCS202** and **Fatima Laila Maryam Khalida Zayneb** are the name of students registered in CPCS202, similarly

.

.
in green color three students' marks, indicate course CPCS203 and Waleed Rashid Hassan are the name of students registered in CPCS203, and so on.

Note: There will be total 15 students' marks (5+3+4+3)

e) Print_Result – Your program must read Students marks for a course and automatically calculates the sum of marks of all students out of 100 and finds the maximum marks gainer (Topper) of the course and also record marks result of each students in an output file. [see print.txt]

Print_Result CPCS202

In above line **Print_Result** is a command and **CPCS202** is the Course, your program must read all cpcs202 Students marks and automatically calculates out of 100 marks for all students, further, the program finds the maximum marks (topper) of the course CPCS202 and also record marks and result of each students in an output file. [see print.txt], similarly

f) Print_Result_For_All - Your program must read Students marks for all courses and automatically finds the maximum marks (Topper) of all the courses and store in an output file. [see print.txt]

[YOU MUST GENERATE EXACTLY SAME OUTPUT FILE AS GIVEN TO YOU (PRINT.TXT)]

g) Quit- Your program will end!

The Data Creation of the Program

1) **Create a Single Dimension Array** to store course codes and **create a method** that read course codes from input.txt file and store in array.

// String array for the course code

2) **Create a Single Dimension Array** to store Exam Type and **create a method** that read types of Exam from input.txt file and store in array.

// String array for the Exam types

3) **Create a Two Dimension Array** to store Students names registered in a Course and **create a method** that read Student name from input.txt file and store in array.
// String Two-Dimensional array for the Students

4) **Create a Three-Dimensional Array** to store Marks of each student of different courses for all the Exam Types, and create a method, that reads 3D array of all students Marks. As given **Figure 2**.
// Three-Dimensional Array

MARKS OBTAINED

		Exam1(10)	Exam2(15)	Final(25)	Lab Exams(15)	Assignments (30)	Quiz(5)
CPCS202	Student-1	9	15	23	14	25	5
	Student-2	8	13	18	13	28	4
	Student-3	7	12	16	12	23	5
	Student-4	8	12	15	10	22	3
	Student-5	5	11	15	11	20	2

		Exam1	Exam2	Final	Lab Exams	Assignments	Quiz
CPCS203	Student-1	8	13	24	13	24	5
	Student-2	7	15	23	14	30	5
	Student-3	3	10	20	12	22	5

		Exam1	Exam2	Final	Lab Exams	Assignments	Quiz
CPCS204	Student-1	9	15	25	14	28	4
	Student-2	8	14	24	15	26	5
	Student-3	9	15	23	13	27	5
	Student-4	6	12	22	15	29	4

		Exam1	Exam2	Final	Lab Exams	Assignments	Quiz
CPCS353	Student-1	3	10	22	12	25	3
	Student-2	7	15	23	13	28	4
	Student-3	8	13	24	14	23	5

Figure 2. Students marks in 3 Dimension Array

The 3- Dimensional array virtually link Course array and student array. Each course has number of registered students; you will use the 3 D array to store student marks, so, this 3-Dimensional array have different rows and columns based on the total

students registered in the course. Figure 2 and Figure 3 illustrates the structure of the 3D array.

courses[0] --> CPCS202	EXAM1	EXAM2	FINAL	LABS	ASSIGN..	QUIZZES
student-1	9	15	23	14	25	5
student-2	8	13	18	13	28	4
student-3	7	12	16	12	23	5
student-4	8	12	15	10	22	3
student-5	5	11	15	11	20	2

Figure 3 The structure of the 3D array

Important Notes:

- Your Code, output, results etc must be in a readable form.
- Organize your code in separate methods.
- Repeat the program until Quit command is read by your program.
- Use comments in your code.
- Use meaningful variables.
- Use dash lines separator between each method.

Deliverables:

You have to submit **only one java** file. The file name must be like **P1_123456.java** where, **123456 your ID.**

NOTE: your name, ID, and section number must be included as comments in the file!

Output Format

Your program must generate output in a similar format to the given sample output file [print.txt].

Good Luck and Start Early!