420-AP1-AS - ALGORITHMS AND PROGRAMMING – COMPUTER SCIENCE TECHNOLOGY – PROGRAMMING
Section 07218

ASSIGNMENT 8 – Arrays (two dimensions)

XIMENA CARRILLO

LaSalle College
November 2nd, 2023

Assignment 8 – Arrays (Two dimensions)

For the following exercises:

- Create a new project called “Assignment8” into your solution “AP1_2023”
- Write all the code in a unique C# file.
- You must validate entries from the console (input data validation).
- Send the whole project, compressed in a “zip” or “rar” file by Lea.
- You must use TWO DIMENSIONAL arrays

Exercises:

Context: A teacher requires a program to storage information for all of the students of a course (10 students) and calculate their grades.

You are required to write a C# application that allows the teacher (user) to perform the following operations:

1. **Display the menu**

   At the beginning execution of the program, the following main menu must be displayed:

   MY STUDENTS
   1. Enter **one** student data
   2. Enter **one** student grades
   3. Show all the students information
   4. Show a specific student by ID
   5. Get the average of group
   6. Get best student
   7. Exit the application
   Enter your choice:

   Note: Each option allows to execute a different segment of code. Once, each segment is executed, you should display the menu again in order to allows the teacher (user) chooses another option of the menu.

   You must validate the teacher’s choice.

2. **Enter one student data (Option 1)**.
   2.1. The teacher inputs one student’s information:
   - Student ID (integer)
   - First Name (string)
   - Last Name (string)

   2.2. Validate Entries

Input	Valid	Invalid
Student ID	• 123 • 1234567	• 12345678943 • asdf

		• 123asd • 12365.25
First Name	Anna	“” (empty string)
Last Name	Brown	“” (empty string)

2.3. Save the information. Note: You should save the information in order.

3. **Enter one student grades (Option 2)**.
   3.1. The following is the evaluation chart for each component to evaluate across the course.

Evaluation Chart

Components	**Weight**
Midterm Exam	30%
Final Project	30%
Final Exam	40%
TOTAL	**100%**

   3.2. The program asks for a student row (index) to insert data in.
   3.3. The teacher inputs the students’ grades for each component:
   - Midterm Exam (real number)
   - Final Project (real number)
   - Final Exam (real number)

   3.4. Validate entries: All the grades must be greater than zero and less than or equal to the weight expressed in the evaluation chart.
   3.5. Calculate final grade.
   3.6. The program saves the student information and the final grade in the index read by the console (3.2).

4. **Show all the students' information (Option 3):**
   4.1. The program will list all the students, displaying:
   - Row (index)
   - Student ID
   - First Name
   - Last Name
   - Grades for each component
     - Midterm Exam
     - Final Project
     - Final Exam
     - Final Grade

5. **Show a specific student by ID (Option 4)**
   5.1. The program asks for a student ID to search.
   5.2. The program searches that student ID and:
   - If it doesn’t exist, display the message “The Student doesn’t exist, please try again” and return to the menu.

- If the student exists, you should display the same information as 4.1, but only for the student with the student ID read by console (5.1).

6. **Get the average of group (Option 5)**
   6.1. Get the average of the group and displayed it.
7. **Get best student (Option 6)**
   7.1. Get the student with best final grade and display:
   - Student ID
   - Student Name and First Name
   - Final Grade
8. **Exit the application (Option 7)**
   If this option is selected, a confirmation message must be displayed to allow the teacher to exit the application.