CSE110: Principles of Programming with Java

LAB 11

This lab is for practicing Array of objects.

Use the following Coding Guidelines:

- When declaring a variable, you usually want to initialize it.
- Use white space to make your program more readable.
- Use comments after the ending brace of classes, methods, and blocks to identify to which block it belongs.

Assignments Documentation:

At the beginning of each programming assignment you must have a comment block with the following information:

```
/*---------------------------------------------------------------------------
// AUTHOR:          (Put your name here)
// FILENAME:        Lab11.java
// SPECIFICATION:   Go through the instructions and write your own
//                  description.
// LAB LETTER:      (Put your Lab Letter here)
// DATE:            mm/dd/yyyy
//-------------------------------------------------------------------------*/
```

Making Student Class (Student.java)

```
// Declare class Student
// Do not declare main method
// Declare two private variables [name] and [marks] of type String and integer
respectively

// Declare Student constructor that takes in two arguments [someName] of type
String and [someMarks] of type integer in the order as they are mentioned
// Within Student constructor: assign [someName] to [name], [someMarks] to
[marks]

// Declare a public method setMarks() of void type, with one argument
[newMarks] of type integer
// In this method, assign [newMarks] to [marks]

// Declare a public method getMarks() of int return type with no argument
// In this method, return [marks]

// Declare a public method getName() of String return type with no argument
// In this method, return [name]
```

Getting Started

You are done with making the **Student class**.
Create a new class called **Lab11**.

Part 1: Methods:

1. **public static int sum(Student[] _student)**

// Here, in the declaration of the method, we have given a reference of array of Student objects as an argument

// Within this method, declare an int variable called [sum] and initialize it to zero.

// Make a for loop to iterate through all the elements of the reference passed. (start the loop from i=0 and continue till i<_student.length)

// In the for loop, use getMarks() method of Student class to take the marks of each student and add that marks to sum variable.

// At the end of the for loop, return [sum]

2. **public static void update(Student[] _student, String _name, int _value)**

// Make a for loop to iterate through all the elements of the reference passed.

// In the for loop, use getName() method of Student class to check whether the argument _name matches with any of the student's name.

// If a student name matches, set marks of that student to the _value given as an argument. (use setMarks(_value) to do that)

// At the end of the for loop, call print() method to print the array of objects _student. (use print(_student))

// We are going to create print() method later, don't worry !!

3. **public static int max(Student[] _student)**

// Declare a variable of int type called [max] and initialize it to an appropriate value.

// Make a for loop to iterate through all the elements of the reference passed.

// In the for loop, check for each _student's marks and see if the marks are greater than the max value. (use _student[i].getMarks()) If so, assign _student[i].getMarks() to [max]

// After the end of the for loop, return [max]

4. **public static void swap(Student[] _student, String name1, String name2)**

// Declare two int variables [index1] and [index2] and initialize them to 0.

// Make a for loop to iterate through all the elements of the reference passed.

// In the for loop, if a _student's name matches with name1 then assign i to index1

// Also, have a second if condition to see if a _student's name matches with name2 then assign i to index2

// After the for loop, swap the marks of _student at [index1] with the marks of _student at [index2]. (use the logic of swapping elements for that use a variable called [temp] of type int)

```
// At the end of the swapping, call print() method to print the elements of the array object. (use print(_student))
```

5. public static void print(Student[] _student)

```
// Print (“The list is :”)
// Make a for loop to iterate through all the elements of the reference passed.
// In the for loop, Print (“Name : ” + _student[i].getName() + “, Marks : ” + _student[i].getMarks())
```

```
********** Done with all the methods **********
```

In the main method

Part 1: Declaration

```
// Initialize the Scanner
// Declare variable [name] of type String
// Declare variable [marks] of type int
// Declare variable [num] of type int
// Declare variable [choice] of type int
// Prompt the user to “Enter the total number of Students”
// Take the input value into [num] variable

// Declare a Student class array of objects called students. The size of the array object will be [num]
// Create a for loop that runs from i=0 to i<num
// In the for loop, Prompt the user to “Enter name and marks of student”
// Take the first input from the user into [name] variable
// Take the second input from the user into [marks] variable
// Now, call the Student constructor for each Student object and give in arguments name and marks. (use students[i] = new Student(name, marks))

// Outside the for loop, declare constant integers: sum = 1, update = 2, swap = 3, max_marks = 4 and quit = 5.
```

Part 2: Preparation

```
// Create a do-while loop which will terminate only when the choice is quit
// Print the following in the loop:
// “This program will do the following”
// “1. Total marks”
// “2. Update a student’s marks”
// “3. Swap marks”
// “4. Find maximum marks”
// “5. Exit”
// “Give your choice for the above options”
// Take that choice that user enters, into the [choice] variable
```

Part 3: Switch Case

```
// case ???:
// Print the sum returned by sum(students) method
// break the case

// case ???:
// Prompt the user to “Enter student name”
// Declare a String variable called [nName]
// Take the input into [nName] variable
// Prompt the user to “Enter student marks”
// Declare an integer variable called [nMarks]
// Take the input into [nMarks] variable
// Call the update() method  ( use update(students, nName, nMarks))
// break the case

// case ???:
// Prompt the user to “Enter first student’s name”
// Declare a String variable called [name1]
// Take the input into [name1] variable
// Prompt the user to “Enter second student’s name”
// Declare a String variable called [name2]
// Take the input into [name2] variable
// Call the swap() method (use swap(students, name1, name2)
// break the case

// case ???:
// Print the maximum returned by max(students) method
// break the case

// case ???:
// Print “Exiting the program…”
// break the case

// default:
// Print “Invalid choice, try again!”
// break the case
```

Sample Output:

```
Enter the total number of Students
5
Enter name and marks of student(1)
Jacob
90
Enter name and marks of student(2)
William
100
Enter name and marks of student(3)
Anna
95
Enter name and marks of student(4)
Ben
85
Enter name and marks of student(5)
Daniel
90
This program will do the following
1. Total marks
2. Update a student's marks
3. Swap marks
4. Find maximum marks
5. Exit
```

```
Give your choice for the above options
1
The sum of marks is 460
This program will do the following
1. Total marks
2. Update a student's marks
3. Swap marks
4. Find maximum marks
5. Exit
Give your choice for the above options
2
Enter student name
William
Enter new marks
95
The list is :
Name : Jacob, Marks : 90
Name : William, Marks : 95
Name : Anna, Marks : 95
Name : Ben, Marks : 85
Name : Daniel, Marks : 90
This program will do the following
1. Total marks
2. Update a student's marks
3. Swap marks
4. Find maximum marks
5. Exit
Give your choice for the above options
3
Enter first student's name :
Ben
Enter second student's name :
Jacob
The list is :
Name : Jacob, Marks : 85
Name : William, Marks : 95
Name : Anna, Marks : 95
Name : Ben, Marks : 90
Name : Daniel, Marks : 90
This program will do the following
1. Total marks
2. Update a student's marks
3. Swap marks
4. Find maximum marks
5. Exit
Give your choice for the above options
4
Maximum marks obtained are: 95
This program will do the following
1. Total marks
2. Update a student's marks
3. Swap marks
4. Find maximum marks
5. Exit
Give your choice for the above options
5
Exiting the program......
```

Submission: Submit your Lab11.java file and Student.java file to the Submission Server.
You should then check to make sure that the actual files are submitted properly and are readable to the grader.

Note:
1. You may resubmit as many times as you like until the deadline, but we will only mark your last submission.
2. If you see a lot of errors on ASCII issue in the online submission, change the file property as UTF-8 (Especially Eclipse User) and re-type the quotations (' or ") and apostrophe (') again. Ask your TA for more details.
3. Scanner MUST be created only once.
4. The scanner methods, next() and nextLine() are different. Use next() for this exercise.

ENJOY CODING!!